《用户指南》

AWS CodeStar

AWS CodeStar: 《用户指南》

Table of Contents

2024 年 7 月 31 日，Amazon Web Services（AWS）将停止支持创建和查看 AWS CodeStar 项目。2024 年 7 月 31 日后，您将无法再访问 AWS CodeStar 控制台或创建新项目。但是，由 AWS CodeStar 创建的 AWS 资源（包括您的源存储库、管道和构建）将不受此更改的影响，并将继续运行。AWS CodeStar连接和 AWS CodeStar 通知不受此次停止支持的影响。

如果您想跟踪工作，开发代码并构建、测试和部署应用程序，Amazon CodeCatalyst 可提供简化的入门流程和其他功能来管理您的软件项目。详细了解 Amazon CodeCatalyst 的功能和定价。

什么是 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。AWS CodeStar 项目会为您的项目开发工具链创建和集成 AWS 服务。根据您选择的 AWS CodeStar 项目模板，该工具链可能包含源代码控制、构建、部署、虚拟服务器或无服务器资源等。AWS CodeStar 还管理项目用户（称为“团队成员”）所需的权限。通过将用户作为团队成员添加到 AWS CodeStar 项目，项目所有者可以快速轻松地向每个团队成员角色授予对项目及其资源的相应访问权限。

主题

AWS CodeStar 可以用来做什么？

您可以使用 AWS CodeStar 帮助您在云中设置您的应用程序开发，并从一个集中式的控制面板管理您的开发。具体来说，您可以：

- 通过使用适用于 Web 应用程序、Web 服务等的模板，在几分钟内在 AWS 上启动新的软件项目：AWS CodeStar 包括适用于各种项目类型和编程语言的项目模板。由于 AWS CodeStar 负责设置，因此您的所有项目资源均配置为配合使用。
- 管理团队的项目访问权限：AWS CodeStar 提供了一个中央控制台，您可在该控制台中向项目团队成员分配其访问工具和资源所需的角色。这些权限将自动应用于项目中使用的所有 AWS 服务，因此，您无需创建或管理复杂的 IAM 策略。
- 在一个位置可视化、操作和协作处理您的项目：AWS CodeStar 包括一个项目控制面板，该面板概述了项目、工具链和重要事件。您可以监控最新的项目活动（如最近的代码提交）并跟踪代码更改的状态、生成结果和部署，所有这一切操作都在同一网页中进行。您可以从一个控制面板中监控项目的进展情况，然后深入了解问题以进行调查。
- 使用所需的所有工具快速迭代：AWS CodeStar 包含一个适用于您的项目的集成式开发工具链。团队成员推送代码，并且将自动部署更改。通过与问题跟踪集成，团队成员可以跟踪接下来需要执行的操作。您和您的团队可在代码交付的所有阶段更快速高效地协作。

如何开始使用 AWS CodeStar？

开始使用 AWS CodeStar：

1. 按照中的步骤操作，准备AWS CodeStar使用 设置 AWS CodeStar。
2. 按照 开始使用 AWS CodeStar 教程中的步骤，使用 AWS CodeStar 来进行试验。
3. 按照 中的步骤操作，与其他开发人员共享向 AWS CodeStar 项目添加团队成员 您的项目。
4. 按照 将 AWS CodeStar 与 IDE 配合使用 中的步骤，集成您最喜欢的 IDE。

设置 AWS CodeStar

必须先完成以下步骤才能开始使用 AWS CodeStar。

主题

步骤 1：创建账户

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。

步骤 2：创建 AWS CodeStar 服务角色

创建用于授予 AWS CodeStar 权限（用于代表您管理 AWS 资源和 IAM 权限）的服务角色。您只需创建服务角色一次。

> **Important**
>
> 您必须以管理员用户（或根账户）身份登录才能创建服务角色。有关更多信息，请参阅创建您的第一个 IAM 用户和组。

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 选择 Start project。

   如果您没有看到启动项目而是被定向到项目列表页面，则表示服务角色之前已创建。
3. 在创建服务角色中，选择是，创建角色。
4. 退出向导。您稍后将会回到这一步。

步骤 3：配置用户的 IAM 权限

除了管理用户，您还能以 IAM 用户、联合用户、根用户或代入角色的身份使用 AWS CodeStar。有关 AWS CodeStar 可为 IAM 用户与联合身份用户提供哪些服务的信息，请参阅AWS CodeStar IAM 角色。

如果您未设置任何 IAM 用户，请参阅 IAM 用户。

要提供访问权限，请为您的用户、群组或角色添加权限：

- AWS IAM Identity Center 中的用户和群组：

  创建权限集。按照《AWS IAM Identity Center 用户指南》中创建权限集的说明进行操作。
- 通过身份提供商在 IAM 中托管的用户：

  创建适用于身份联合验证的角色。按照《IAM 用户指南》中为第三方身份提供商创建角色（联合身份验证）的说明进行操作。
- IAM 用户：
  - 创建您的用户可以代入的角色。按照《IAM 用户指南》中为 IAM 用户创建角色的说明进行操作。
  - （不推荐使用）将策略直接附加到用户或将用户添加到用户群组。按照《IAM 用户指南》中向用户添加权限（控制台）中的说明进行操作。

步骤 4：为 AWS CodeStar 项目创建 Amazon EC2 密钥对

许多 AWS CodeStar 项目使用 AWS CodeDeploy 或 AWS Elastic Beanstalk 将代码部署到 Amazon EC2 实例。要访问与您的项目关联的 Amazon EC2 实例，请为您的 IAM 用户创建一个 Amazon EC2 密钥对。您的 IAM 用户必须拥有创建和管理 Amazon EC2 密钥的权限（例如，执行 `ec2:CreateKeyPair` 和 `ec2:ImportKeyPair` 操作的权限）。有关更多信息，请参阅 Amazon EC2 密钥对。

步骤 5：打开 AWS CodeStar 控制台

登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。

后续步骤

恭喜您，您已完成设置！要开始使用 AWS CodeStar，请参阅开始使用 AWS CodeStar。

开始使用 AWS CodeStar

在本教程中，您使用 AWS CodeStar 创建 Web 应用程序。此项目包括源存储库中的示例代码、一个持续部署工具链以及一个您可在其中查看和监控项目的项目控制面板。

通过执行这些步骤，您将：

- 在 AWS CodeStar 中创建项目。
- 浏览项目。
- 提交代码更改。
- 查看自动部署的代码更改。
- 添加处理项目的其他人员。
- 在不再需要项目资源时将其清除。

> **Note**
>
> 如果您尚未这样做，请首先完成设置 AWS CodeStar 中的步骤，包括步骤 2：创建 AWS CodeStar 服务角色。您必须使用 IAM 中管理用户的账户登录。要创建项目，您必须使用具有 **`AWSCodeStarFullAccess`** 策略的 IAM 用户登录 AWS Management Console。

主题

- 教程：使用 GitHub 源代码库创建项目

步骤 1：创建 AWS CodeStar 项目

在此步骤中，您将为 Web 应用程序创建一个 JavaScript (Node.js) 软件开发项目。您将使用 AWS CodeStar 项目模板创建项目。

> **Note**
>
> 本教程中使用的 AWS CodeStar 项目模板使用以下选项：
>
> - Application category：Web 应用程序
> - Programming language：Node.js
> - AWS 服务：Amazon EC2
>
> 如果您选择其他选项，则您的体验可能与本教程中记录的体验不匹配。

在 AWS CodeStar 中创建项目

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。

   确保您已登录要在其中创建项目及其资源的 AWS 区域。例如，要在美国东部（俄亥俄州）中创建项目，请确保您已选择 AWS 区域。有关 AWS CodeStar 在其中可用的 AWS 区域的信息，请参阅 AWS 一般参考中的区域和端点。
2. 在 AWS CodeStar 页面上，选择创建项目。
3. 在创建项目模板页面上，从 AWS CodeStar 项目模板的列表中选择项目类型。您可使用筛选栏缩小所选内容的范围。例如，对于要部署到 Amazon EC2 实例、用 Node.js 编写的 Web 应用程序项目，请选中 Web 应用程序、Node.js 和 Amazon EC2 复选框。随后，从可用于此选项集的模板中进行选择。

   有关更多信息，请参阅AWS CodeStar 项目模板。
4. 选择下一步。
5. 在项目名称文本输入字段中，输入项目的名称，例如#######。在项目 ID 中，项目的 ID 派生自此项目名称，但限制为 15 个字符。

例如，名为*#######*的项目的默认 ID 是 *my-first-projec*。此项目 ID 是与项目关联的所有资源的名称的基础。AWS CodeStar 使用此项目 ID 作为代码存储库 URL 的一部分，以及 IAM 中相关安全访问角色和策略的名称的一部分。创建项目后，项目 ID 便无法更改。要在创建项目之前编辑项目 ID，请在项目 ID 中输入要使用的 ID。

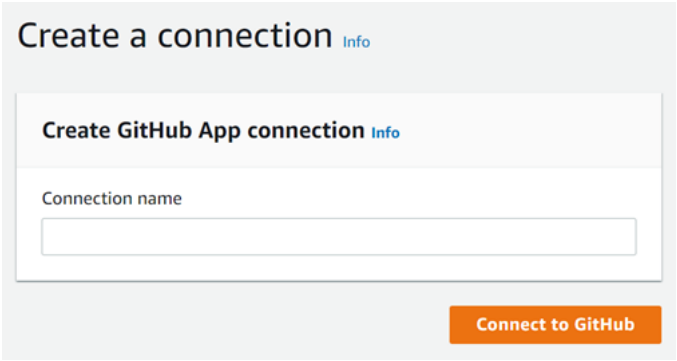
有关项目名称和项目 ID 的限制的信息，请参阅 AWS CodeStar 中的限制。

> **Note**
>
> 对于您的 AWS 账户来说，项目 ID 在某个 AWS 区域必须唯一。

6. 选择存储库提供商 AWS CodeCommit 或 GitHub。
7. 如果您选择了 AWS CodeCommit，则对于存储库名称，请接受默认的 AWS CodeCommit 存储库名称，或输入其他名称。然后跳至步骤 9。
8. 如果您选择 GitHub，则需要选择或创建连接资源。如果您已有连接，请在搜索栏中选择该连接。否则，立即创建新连接。选择连接到 GitHub。

   创建连接页面随即显示。

   > **Note**
   >
   > 要创建连接，您必须拥有 GitHub 账户。您必须是组织所有者才能为组织创建连接。



   a. 在创建 GitHub 应用程序连接下连接名称的输入文本字段中，输入连接的名称。选择连接到 GitHub。

      连接到 GitHub 页面将出现，并显示 GitHub 应用程序字段。

b. 在 GitHub 应用程序下，选择一个应用程序安装或选择安装新应用程序来创建一个应用程序安装。

> **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 AWS Connector for GitHub 应用程序，选择它并跳过此步骤。

c. 在安装 AWS Connector for GitHub 页面中，选择要在其中安装应用程序的账户。

> **Note**
>
> 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。

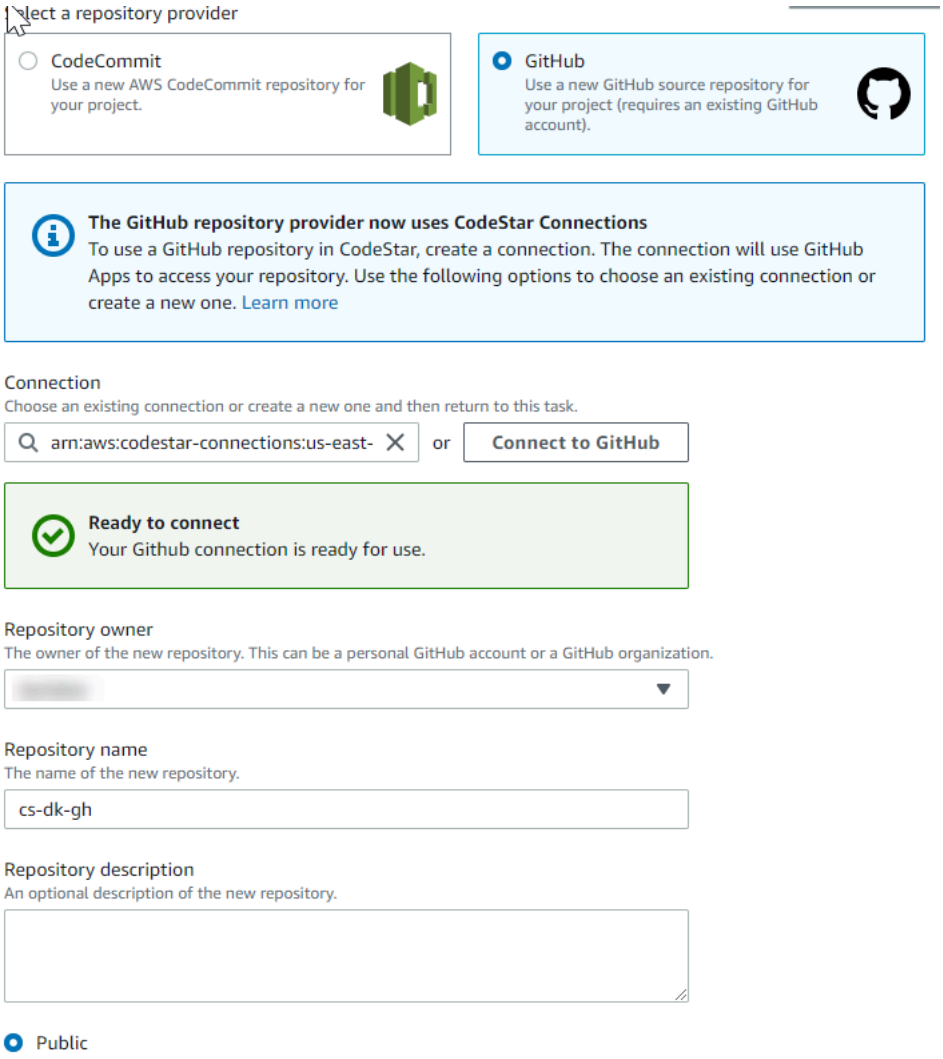
d. 如果显示确认密码以继续页面，请输入您的 GitHub 密码，然后选择登录。

e. 在安装 AWS Connector for GitHub 页面中，保留所有默认设置，然后选择安装。

f. 在连接到 GitHub 页面上，新安装的连接 ID 将显示在 GitHub 应用程序文本输入字段中。

创建连接后，在 CodeStar“创建项目”页面中，将显示准备连接消息。

> **Note**
>
> 您可以在开发人员工具控制台的设置下查看您的连接。有关更多信息，请参阅开始使用连接。



g. 对于存储库所有者，请选择 GitHub 组织或您的个人 GitHub 账户。

h. 对于存储库名称，请接受默认的 GitHub 存储库名称，或输入其他名称。

i. 选择 公共或私有。

> **Note**
>
> 此外，如果您想使用 AWS Cloud9 作为开发环境，则必须选择公共。

j. （可选）对于存储库描述，请为 GitHub 存储库输入描述。

> **Note**
>
> 如果您选择 Alexa Skill 项目模板，则需要关联 Amazon 开发人员账户。有关如何使用 Alexa Skill 项目的更多信息，请参阅 教程：在 AWS CodeStar 中创建 Alexa 技能项目。

9. 如果您的项目已部署到 Amazon EC2 实例，并且您想进行更改，请在 Amazon EC2 配置中配置您的 Amazon EC2 实例。例如，您可以从项目的可用实例类型中进行选择。

   > **Note**
   >
   > 不同的 Amazon EC2 实例类型将提供不同级别的计算能力，并且可能具有不同的关联成本。有关更多信息，请参阅 Amazon EC2 实例类型和 Amazon EC2 定价。
   > 如果您在 Amazon Virtual Private Cloud 中创建了多个虚拟私有云 (VPC) 或多个子网，则还可选择要使用的 VPC 和子网。但是，如果您选择专用实例上不支持的 Amazon EC2 实例类型，则无法选择其实例租赁设置为专用的 VPC。
   > 有关更多信息，请参阅什么是 Amazon VPC？和专用实例基本信息。

   在密钥对中，选择在 步骤 4：为 AWS CodeStar 项目创建 Amazon EC2 密钥对 创建的 Amazon EC2 密钥对。选择我确认我有权访问私钥文件。
10. 选择下一步。
11. 查看资源和配置详细信息。
12. 选择 Next 或 Create project。（显示的选择取决于您的项目模板。）

    创建项目（包括存储库）可能需要几分钟时间。
13. 在项目拥有存储库后，您可以使用存储库页面来配置对它的访问权限。使用后续步骤中的链接来配置 IDE，设置问题跟踪或向项目中添加团队成员。

步骤 2：为您的 AWS CodeStar 用户配置文件添加显示信息

在创建项目时，您将作为所有者添加到项目团队。如果这是您首次使用 AWS CodeStar，系统会要求您提供：

- 要显示给其他用户的显示名称。
- 要显示给其他用户的电子邮件地址。

此信息将在您的 AWS CodeStar 用户配置文件中使用。用户配置文件不是特定于项目的，而是受限于 AWS 区域。您必须在您属于其中项目的每个 AWS 区域中创建一个用户配置文件。如果您愿意，每个配置文件可包含不同的信息。

输入用户名和电子邮件地址，然后选择下一步。

> **Note**
>
> 您的 AWS CodeStar 用户配置文件将使用此用户名和电子邮件地址。如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），这些资源提供者可能具有自己的用户配置文件，这些配置文件具有不同的用户名和电子邮件地址。有关更多信息，请参阅资源提供者的文档。

步骤 3：查看您的项目

您的 AWS CodeStar 项目页面是可供您和您的团队查看项目资源的状态的位置，包括项目的最新提交、持续交付管道的状态以及实例的性能。要查看有关这些资源的更多信息，请从导航栏中选择相应的页面。

在新项目中，导航栏包含以下页面：

- 概述页面包含有关您的项目活动、项目资源和项目 README 内容的信息。
- IDE 页面用于将项目连接到集成式开发环境 (IDE) 以修改、测试和推送源代码更改。它包含为 GitHub 和 AWS CodeCommit 存储库配置 IDE 的说明以及有关您的 AWS Cloud9 环境的信息。
- 存储库页面显示您的存储库详细信息，包括名称、提供商、上次修改时间和克隆 URL。您还可以查看有关最新提交的信息，并查看和创建拉取请求。
- 管道页面显示有关您管道的 CI/CD 信息。您可以查看管道详细信息，例如名称、最近的操作和状态。您可以查看管道历史记录并发布更改。您还可以查看管道中各个步骤的状态。
- 监控页面根据您的项目配置显示 Amazon EC2 或 AWS Lambda 指标。例如，它显示由您管道中的 AWS Elastic Beanstalk 或 CodeDeploy 资源部署到的任何 Amazon EC2 实例的 CPU 利用率。在使用 AWS Lambda 的项目中，它会显示 Lambda 函数的调用和错误指标。此信息按小时显示。如果您使用了本教程中建议的 AWS CodeStar 项目模板，则您在将应用程序首次部署到这些实例时应看到活动中出现明显的峰值。您可以刷新监控以查看您的实例运行状况的更改，这可帮助您确定问题或对更多资源的需求。
- 问题页面用于将您的 AWS CodeStar 项目与 Atlassian JIRA 项目集成。配置此磁贴将使您和您的项目团队能够从项目控制面板跟踪 JIRA 问题。

在控制台左侧的导航窗格中，您可以在项目、团队和设置页面之间导航。

步骤 4：提交更改

首先，查看项目中包含的示例代码。在项目导航中的任意位置选择查看应用程序，即可查看应用程序的外观。您的示例 Web 应用程序将显示在新窗口或浏览器标签页中。这是 AWS CodeStar 构建并部署的项目示例。

如果您想查看代码，请在导航栏中选择存储库。选择存储库名称下的链接，您的项目存储库将在新的选项卡或窗口中打开。读取存储库的自述文件 (`README.md`) 的内容，然后浏览这些文件的内容。

在这个步骤中，您将更改代码，然后将更改推送到存储库。可以通过以下方式之一执行此操作：

- 如果项目的代码存储在 CodeCommit 或 GitHub 存储库中，则您可以使用 AWS Cloud9 直接从 Web 浏览器处理代码，而无需安装任何工具。有关更多信息，请参阅为项目创建 AWS Cloud9 环境。
- 如果项目的代码存储在 CodeCommit 存储库中，并且安装了 Visual Studio 或 Eclipse，则使用 AWS Toolkit for Visual Studio 或 AWS Toolkit for Eclipse 可更方便地连接代码。有关更多信息，请参阅将 AWS CodeStar 与 IDE 配合使用。如果没有 Visual Studio 或 Eclipse，请安装 Git 客户端，然后按照本步骤后面的说明进行操作。
- 如果项目代码存储在 GitHub 存储库中，可以使用 IDE 中的工具连接 GitHub。
  - 对于 Visual Studio，可以使用 GitHub Extension for Visual Studio 等工具。有关更多信息，请参阅 GitHub Extension for Visual Studio 网站上的 Overview 页面及 GitHub 网站上的 Getting Started with GitHub for Visual Studio。
  - 对于 Eclipse，可以使用 EGit for Eclipse 等工具。有关更多信息，请参阅 EGit 网站上的 EGit 文档。
  - 对于其他 IDE，请参阅相应 IDE 文档。
- 对于其他类型的代码存储库，请参阅存储库提供商的文档。

以下说明介绍如何对示例进行次要更改。

设置您的计算机以提交更改（IAM 用户）

> **Note**
>
> 在此过程中，我们假定项目的代码存储在 CodeCommit 存储库中。对于其他类型的代码存储库，请参阅存储库提供商的文档，然后向前跳到下一个过程克隆项目存储库并进行更改。

如果代码存储在 CodeCommit 中并且您已在使用 CodeCommit，或者您已使用 AWS CodeStar 控制台为项目创建了 AWS Cloud9 开发环境，则无需其他配置。请向前跳到下一个过程克隆项目存储库并进行更改。

1. 在您的本地计算机上安装 Git。
2. 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/。

   以将使用 Git 凭证来连接您在 CodeCommit 中的 AWS CodeStar 项目存储库的 IAM 用户身份登录。
3. 在 IAM 控制台的导航窗格中，选择 用户，然后从用户列表中选择您的 IAM 用户。
4. 在用户详细信息页上，选择安全凭证选项卡，然后在 CodeCommit 的 HTTPS Git 凭证中，选择生成。

   > **Note**
   >
   > 您不能为 Git 凭证选择自己的登录凭证。有关更多信息，请参阅将 Git 凭证和 HTTPS 与 CodeCommit 配合使用。
5. 复制 IAM 为您生成的登录凭证。您可以选择 Show，然后将此信息复制并粘贴到本地计算机上的安全文件中，也可以选择 Download credentials 来以 .CSV 文件格式下载此信息。您需要这些信息才能连接到 CodeCommit。

   保存您的凭证后，选择 Close。

   > **Important**
   >
   > 这是您保存登录凭证的唯一机会。如果您未保存，可以从 IAM 控制台复制用户名，但无法查找密码。此时，您必须重置密码，然后保存它。

设置您的计算机以提交更改（联合身份用户）

您可以使用控制台将文件上传到存储库，也可以使用 Git 从您的本地计算机进行连接。如果您使用的是联合访问，请按照以下步骤使用 Git 从您的本地计算机连接到存储库并进行克隆。

> **Note**
>
> 在此过程中，我们假定项目的代码存储在 CodeCommit 存储库中。对于其他类型的代码存储库，请参阅存储库提供商的文档，然后向前跳到下一个过程克隆项目存储库并进行更改。

1. 在您的本地计算机上安装 Git。
2. 安装 AWS CLI。
3. 为联合身份用户配置临时安全凭证。有关信息，请参阅临时访问 CodeCommit 存储库。临时凭证包括：

   - AWS 访问密钥
   - AWS 私有密钥
   - 会话令牌

   有关临时凭证的更多信息，请参阅 GetFederationToken 的权限。
4. 使用 AWS CLI 凭证辅助程序连接到您的存储库。有关信息，请参阅在 Linux、macOS 或 Unix 上使用 AWS CLI 凭证助手设置到 CodeCommit 存储库的 HTTPS 连接的步骤或在 Windows 上使用 AWS CLI 凭证助手设置到 CodeCommit 存储库的 HTTPS 连接的步骤
5. 以下示例演示如何连接到 CodeCommit 存储库并向其推送提交。

示例：克隆项目存储库并进行更改

> **Note**
>
> 此过程介绍如何将项目的代码存储库克隆到您的计算机，更改项目的 `index.html` 文件，然后将更改推送到远程存储库。在此过程中，我们假定项目的代码存储在 CodeCommit 存储库中，并且您从命令行使用 Git 客户端。对于其他类型的代码存储库或工具，请参阅提供商的文档，了解如何克隆存储库、更改文件，然后推送代码。

1. 如果您已使用 AWS CodeStar 控制台为项目创建了 AWS Cloud9 开发环境，请打开该开发环境，然后跳到此过程中的步骤 3。要打开该开发环境，请参阅为项目打开 AWS Cloud9 环境。

在项目在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择存储库。在克隆 URL 中，选择您为 CodeCommit 设置的连接类型的协议，然后复制该链接。例如，如果您执行前一个过程中的步骤为 CodeCommit 设置了 Git 凭证，请选择 HTTPS。

2. 在本地计算机上，打开终端或命令行窗口，然后将目录更改为临时目录。运行 git clone 命令以将存储库克隆到您的计算机。粘贴所复制的链接。例如，对于使用 HTTPS 的 CodeCommit：

```
git clone https://git-codecommit.us-east-2.amazonaws.com/v1/repos/my-first-projec
```

在第一次连接时，系统会提示您提供该存储库的登录凭证。对于 CodeCommit，请输入您在上一步下载的 Git 凭证登录凭证。

3. 导航到计算机上的克隆目录，然后浏览相应内容。

4. 打开 index.html 文件（在公共文件夹中），然后对该文件进行更改。例如，在 <H2> 标签后添加一个段落，如：

```
<P>Hello, world!</P>
```

保存该文件。

5. 在终端或命令提示符下，添加更改后的文件，然后提交并推送您的更改：

```
git add index.html
git commit -m "Making my first change to the web app"
git push
```

6. 在存储库页面上，查看正在进行的更改。您应看到，已使用您的提交更新了存储库的提交历史记录，包括提交消息。在管道页面上，您可以看到管道接受您对存储库进行的更改，并开始构建和部署它。部署 Web 应用程序后，您可以选择查看应用程序以查看您的更改。

> **Note**
>
> 如果任何管道阶段显示失败，请参阅以下内容获取问题排查帮助：
>
> - 对于源阶段，请参阅 AWS CodeCommit 用户指南中的 AWS CodeCommit 问题排查。
> - 对于构建阶段，请参阅 AWS CodeBuild 用户指南中的 AWS CodeBuild 问题排查。

- 对于部署阶段，请参阅 AWS CloudFormation 用户指南中的 AWS CloudFormation 问题排查。
- 有关其他问题，请参阅故障排除 AWS CodeStar。

步骤 5：添加更多团队成员

每个 AWS CodeStar 项目已配置了三个 AWS CodeStar 角色。每个角色提供对项目及其资源的自身级别的访问权限：

- 所有者：可添加和删除团队成员、更改项目控制面板以及删除项目。
- 贡献者：可更改项目控制面板和提供代码（如果代码存储在 CodeCommit 中），但不能添加或删除团队成员或者删除项目。这是您应在 AWS CodeStar 项目中为大多数团队成员选择的角色。
- 查看者：可查看项目控制面板、项目代码（如果代码存储在 CodeCommit 中）以及项目的状态，但不能在项目控制面板中移动、添加或删除磁贴。

> **Important**
>
> 如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），对这些资源的访问由资源提供者而非 AWS CodeStar 控制。有关更多信息，请参阅资源提供者的文档。
>
> 有权访问 AWS CodeStar 项目的任何人都能使用 AWS CodeStar 控制台访问位于 AWS 以外但与此项目有关的资源。
>
> AWS CodeStar 不会允许项目团队成员参与项目的任何相关 AWS Cloud9 开发环境。要允许团队成员参与共享环境，请参阅与项目团队成员共享 AWS Cloud9 环境。

有关团队和项目角色的更多信息，请参阅与 AWS CodeStar 团队协作。

将团队成员添加到 AWS CodeStar 项目（控制台）

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您的项目。
3. 在项目的侧导航栏中，选择团队。

4. 在 Team members 页面上，选择 Add team member。
5. 在 Choose user 中，执行下列操作之一：

   - 如果要添加的人员已有 IAM 用户，请从列表中选择该 IAM 用户名。

     > **Note**
     >
     > 已添加到另一个 AWS CodeStar 项目的用户将显示在现有 AWS CodeStar 用户列表中。

     在项目角色中，选择要此用户的 AWS CodeStar 角色（所有者、贡献者或查看者）。这是只能由项目所有者更改的 AWS CodeStar 项目级角色。此角色在应用于 IAM 用户时将提供访问 AWS CodeStar 项目资源所需的所有权限。它会应用执行以下操作所需的策略：在 IAM 中为存储在 CodeCommit 中的代码创建和管理 Git 凭证，或在 IAM 中为用户上传 Amazon EC2 SSH 密钥。

     > **Important**
     >
     > 您无法提供或更改 IAM 用户的显示名称或电子邮件信息，除非您已经以该用户身份登录到控制台。有关更多信息，请参阅管理 AWS CodeStar 用户配置文件的显示信息 。

     选择添加团队成员。

   - 如果要添加到项目的人员没有 IAM 用户，请选择创建新 IAM 用户。您将被重定向到 IAM 控制台，可以在其中创建新的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的创建 IAM 用户。创建 IAM 用户后，返回 AWS CodeStar 控制台，刷新用户列表，然后从下拉列表中选择您创建的 IAM 用户。输入要应用于此新用户的 AWS CodeStar 显示名称、电子邮件地址和项目角色，然后选择添加团队成员。

   > **Note**
   >
   > 为了便于管理，应向至少一个用户分配了该项目的“所有者”角色。

6. 向新团队成员发送以下信息：

   - 您的 AWS CodeStar 项目的连接信息。

- 为了从本地计算机访问 CodeCommit 存储库而使用 Git 凭证设置访问权限的说明（如果源代码存储在 CodeCommit 中）。
- 有关用户如何管理其显示名称、电子邮件地址和公有 Amazon EC2 SSH 密钥的信息，如 使用您的 AWS CodeStar 用户配置文件 中所述。
- 一次性密码和连接信息，前提是用户是首次使用 AWS，并且您已为该人员创建 IAM 用户。此密码将在用户首次登录后过期。用户必须选择一个新密码。

步骤 6：清除

恭喜您！您已完成本教程。如果您不想继续使用此项目及其资源，则应将其删除，以避免对您的 AWS 账户持续产生可能的费用。

在 AWS CodeStar 中删除项目

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 在导航窗格中，选择项目。
3. 选择要删除的项目，然后选择删除。

   或者打开项目，并在控制台左侧的导航窗格中选择设置。在项目详细信息页上，选择 Delete project。
4. 在删除确认页面中，选择删除。如果您要删除项目资源，请选中删除资源。选择删除。

   可能需要花费几分钟的时间才能删除项目。删除项目后，项目将不再显示在 AWS CodeStar 控制台中的项目列表中。

> **Important**
>
> 如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），即使选中此复选框，也不会删除这些资源。
> 如果任何 AWS CodeStar 托管策略已手动附加到非 IAM 用户的角色，则无法删除您的项目。如果您已将项目的托管策略附加到联合身份用户的角色，则必须先分离策略才能删除项目。有关更多信息，请参阅???。

步骤 7：为生产环境准备好您的项目

创建项目后，您随时可以创建、测试和部署代码。查看以下有关在生产环境中维护您的项目的注意事项：

- 定期对应用程序所使用的依赖项应用补丁并审查安全最佳实践。有关更多信息，请参阅AWS CodeStar 资源的安全最佳实践。
- 定期监控项目的编程语言所推荐的环境设置。

后续步骤

下面是可帮助您了解 AWS CodeStar 的一些其他资源：

- 教程：在 AWS CodeStar 中创建和管理无服务器项目 使用一个创建和部署 Web 服务的项目，此 Web 服务使用 AWS Lambda 中的逻辑，可通过 Amazon API Gateway 中的 API 调用。
- AWS CodeStar 项目模板 介绍可以创建的其他类型的项目。
- 与 AWS CodeStar 团队协作提供有关邀请他人参加项目的信息。

教程：在 AWS CodeStar 中创建和管理无服务器项目

在本教程中，将使用 AWS CodeStar 创建一个使用 AWS 无服务器应用程序模型 (AWS SAM) 的项目，以便为托管在 AWS Lambda 中的 Web 服务创建和管理 AWS 资源。

AWS CodeStar 使用 AWS SAM（依赖于 AWS CloudFormation）提供一种简化方式，以创建和管理支持的 AWS 资源，包括 Amazon API Gateway API、AWS Lambda 函数和 Amazon DynamoDB 表。（此项目不使用任何 Amazon DynamoDB 表。）

有关更多信息，请参阅 GitHub 上的 AWS 无服务器应用程序模型（AWS SAM）。

先决条件：完成设置 AWS CodeStar中的步骤。

> **Note**
>
> 您的 AWS 账户可能产生与本教程相关的费用，包括 AWS CodeStar 使用 AWS 服务的费用。有关更多信息，请参阅 AWS CodeStar 定价。

主题

概述

在本教程中，您将：

1. 使用 AWS CodeStar 创建一个项目，此项目使用 AWS SAM 构建和部署基于 Python 的 Web 服务。此 Web 服务托管在 AWS Lambda 中，可通过 Amazon API Gateway 访问。
2. 浏览项目的主要资源，其中包括：
   - 存储项目源代码的 AWS CodeCommit 存储库。源代码包含 Web 服务逻辑并定义了相关的 AWS 资源。
   - 自动构建源代码的 AWS CodePipeline 管道。此管道使用 AWS SAM 创建函数并将其部署到 AWS Lambda、在 Amazon API Gateway 中创建相关的 API，然后将该 API 连接至函数。
   - 部署到 AWS Lambda 的函数。
   - 在 Amazon API Gateway 中创建的 API。
3. 测试 Web 服务，确认 AWS CodeStar 按预期构建和部署了 Web 服务。
4. 设置本地工作站来处理项目的源代码。
5. 使用本地工作站更改项目的源代码。当您向项目中添加一个函数、然后将更改推送到源代码时，AWS CodeStar 将重新构建和重新部署 Web 服务。
6. 再次测试 Web 服务，确认 AWS CodeStar 按预期重新构建和重新部署了 Web 服务。

7. 使用本地工作站编写单元测试，用自动测试替换部分手动测试。当您推送单元测试时，AWS CodeStar 将重新构建和重新部署 Web 服务并运行单元测试。
8. 查看单元测试的结果。
9. 清理项目。这一步可避免向您的 AWS 账户收取与本教程相关的费用。

步骤 1：创建项目

在本步骤中，使用 AWS CodeStar 控制台创建项目。

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。

   > **Note**
   >
   > 您必须使用与在 设置 AWS CodeStar 中创建或标识的 IAM 用户关联的凭证登录 AWS Management Console。此用户必须附加了 **AWSCodeStarFullAccess** 托管策略。

2. 选择要在其中创建项目及其资源的 AWS 区域。

   有关 AWS CodeStar 在其中可用的 AWS 区域的信息，请参阅 AWS 一般参考中的区域和端点。
3. 请选择创建项目。
4. 在选择项目模板页面上：

   - 对于应用程序类型，选择 Web 服务。
   - 对于编程语言，选择 Python。
   - 对于AWS 服务，选择 AWS Lambda。
5. 选择您所需项对应的框。选择下一步。
6. 对于项目名称，输入项目的名称（例如，**My SAM Project**）。如果使用不同于示例所用的名称，请确保在本教程中通篇使用它。

   对于项目 ID，AWS CodeStar 将为此项目选择一个相关的标识符（例如，my-sam-project）。如果看到不同于示例所用的项目 ID，请务必在本教程中通篇使用它。

   将 AWS CodeCommit 保留为选中状态，不要更改存储库名称值。
7. 选择下一步。
8. 检查您的设置，然后选择创建项目。

如果是第一次在此 AWS 区域使用 AWS CodeStar，则对于显示名称和电子邮件，输入希望 AWS CodeStar 用于您的 IAM 用户的显示名称和电子邮件地址。选择下一步。

9. 等待 AWS CodeStar 创建项目。这可能需要花几分钟的时间。请在刷新时看到项目已配置横幅后再继续。

步骤 2：浏览项目资源

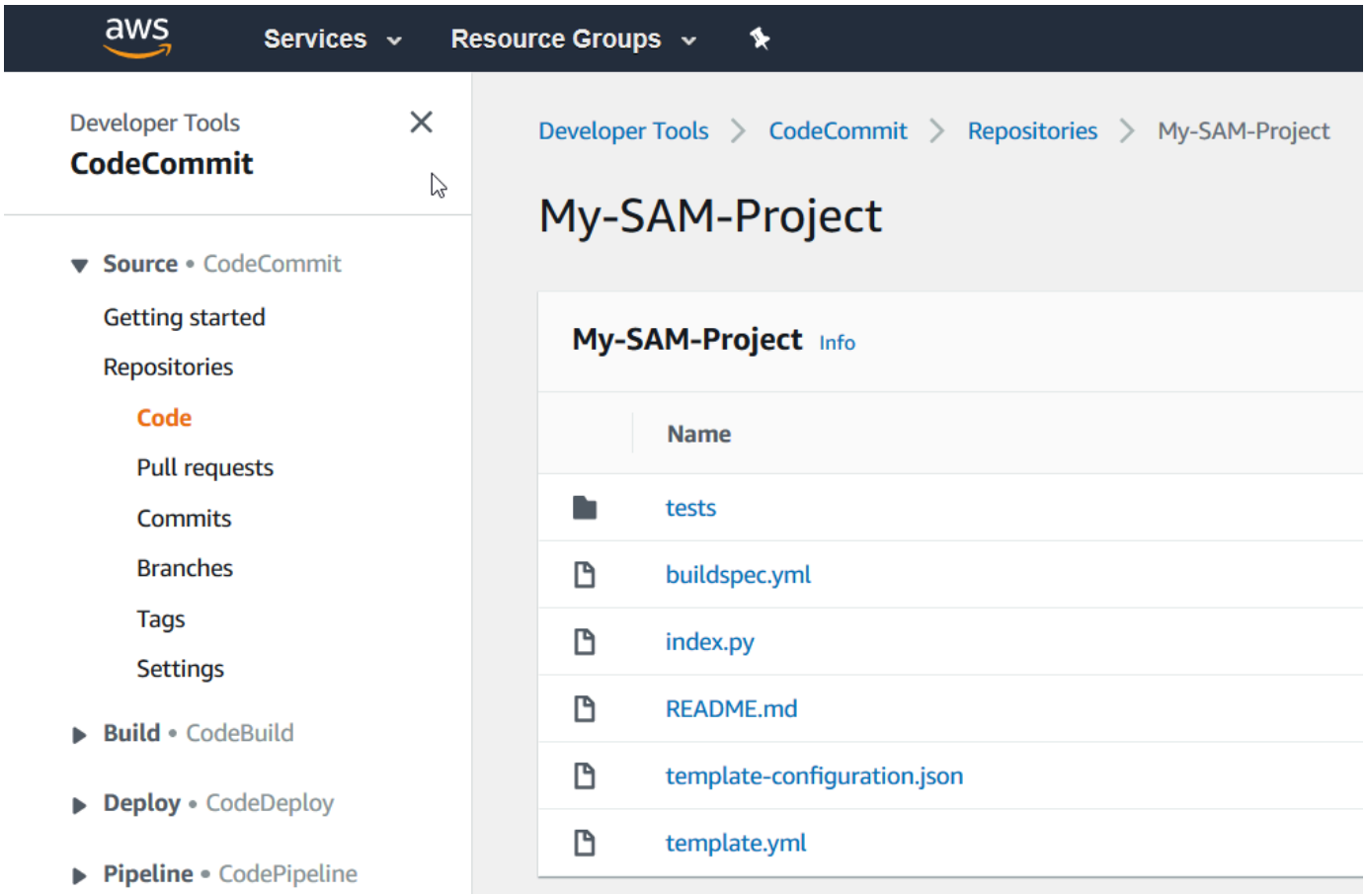
在此步骤中，将浏览项目的四个 AWS 资源，以便了解项目的工作原理：

- 存储项目源代码的 AWS CodeCommit 存储库。AWS CodeStar 为存储库指定名称 my-sam-project，其中 my-sam-project 是项目的名称。
- 使用 CodeBuild 和 AWS SAM 在 API Gateway 中自动构建和部署 Web 服务的 Lambda 函数和 API 的 AWS CodePipeline 管道。AWS CodeStar 为管道指定名称 my-sam-project--Pipeline，其中 my-sam-project 是项目的 ID。
- 包含 Web 服务逻辑的 Lambda 函数。AWS CodeStar 为函数指定名称 awscodestar-my-sam-project-lambda-HelloWorld-***RANDOM_ID***，其中：
  - my-sam-project 是项目的 ID。
  - HelloWorld 是在 AWS CodeCommit 存储库的 `template.yaml` 文件中指定的函数 ID。稍后将浏览该文件。
  - *RANDOM_ID* 是一个随机 ID，AWS SAM 向函数分配此 ID 来帮助确保唯一性。
- API Gateway 中用于方便调用 Lambda 函数的 API。AWS CodeStar 为该 API 指定名称 awscodestar-my-sam-project--lambda，其中 my-sam-project 是项目的 ID。

在 CodeCommit 中浏览源代码存储库

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择存储库。
2. 在存储库详情中选择指向您 CodeCommit 存储库的链接（**`My-SAM-Project`**）。
3. 在 CodeCommit 控制台的代码页面中，将显示项目的源代码文件：
   - `buildspec.yml`，CodePipeline 指示 CodeBuild 在构建阶段使用该文件，以便利用 AWS SAM 打包 Web 服务。
   - `index.py`，包含 Lambda 函数的逻辑。此函数只输出字符串 `Hello World` 及 ISO 格式的时间戳。
   - `README.md` – 包含有关存储库的一般信息。

- `template-configuration.json`，包含带有占位符的项目 ARN，用于使用项目 ID 标记资源
- `template.yml`，AWS SAM 使用它打包 Web 服务和在 API Gateway 中创建 API。



要查看某个文件的内容，请从列表中选择该文件。

有关 CodeCommit 控制台的更多信息，请参阅 AWS CodeCommit 用户指南。

在 CodePipeline 中浏览管道

1. 要查看有关管道的信息，在项目在 AWS CodeStar 控制台中打开的情况下，请在导航栏上选择管道，即可看到管道包括：
   - 源阶段 - 从 CodeCommit 获取源代码。
   - 构建阶段 - 使用 CodeBuild 构建源代码。
   - Deploy 阶段 - 使用 AWS SAM 部署构建好的源代码和 AWS 资源。

2. 要查看有关管道的更多信息，请在管道详细信息中，选择要在 CodePipeline 控制台中打开的管道。

有关 CodePipeline 控制台的信息，请参阅 AWS CodePipeline 用户指南。

在概览页面上浏览项目活动和 AWS 服务资源

1. 在 AWS CodeStar 控制台中打开您的项目，并从导航栏中选择概览。
2. 查看项目活动和项目资源列表。

在 Lambda 中浏览函数

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在侧导航栏上选择概览。
2. 在项目资源的 ARN 列中，选择 Lambda 函数的链接。

   函数代码显示在 Lambda 控制台中。

有关使用 Lambda 控制台的更多信息，请参阅 AWS Lambda 开发人员指南。

在 API Gateway 中浏览 API

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在侧导航栏上选择概览。
2. 在项目资源的 ARN 列中，选择 Amazon API Gateway API 的链接。

   该 API 的资源显示在 API Gateway 控制台中。

有关使用 API Gateway 控制台的更多信息，请参阅 API Gateway 开发人员指南。

步骤 3：测试 Web 服务

在此步骤中，将测试 AWS CodeStar 刚刚构建和部署的 Web 服务。

1. 在上一步中的项目仍然打开的情况下，在导航栏上选择管道。
2. 确保源、构建和部署阶段显示已成功，然后再继续。这可能需要花几分钟的时间。

> **Note**
>
> 如果任何阶段显示失败，请参阅以下内容获取问题排查帮助：

- 对于源阶段，请参阅 AWS CodeCommit 用户指南中的 AWS CodeCommit 问题排查。
- 对于构建阶段，请参阅 AWS CodeBuild 用户指南中的 AWS CodeBuild 问题排查。
- 对于部署阶段，请参阅 AWS CloudFormation 用户指南中的 AWS CloudFormation 问题排查。
- 有关其他问题，请参阅故障排除 AWS CodeStar。

3. 选择查看应用程序。

在 Web 浏览器中打开的新标签页上，Web 服务显示以下响应输出：

```
{"output": "Hello World", "timestamp": "2017-08-30T15:53:42.682839"}
```

步骤 4：设置本地工作站以编辑项目代码

在此步骤中，将设置本地工作站，以便编辑 AWS CodeStar 项目的源代码。本地工作站可以是运行 macOS、Windows 或 Linux 的物理或虚拟计算机。

1. 在上一步中的项目仍然打开的情况下，执行以下操作：
   - 在导航栏中，选择 IDE，然后展开访问您的项目代码。
   - 选择命令行界面下方的查看说明。

     如果安装了 Visual Studio 或 Eclipse，则改为选择 Visual Studio 或 Eclipse 下的查看说明，按照说明进行操作，然后跳到 步骤 5：向 Web 服务添加逻辑。
2. 按照说明完成以下任务：
   a. 在本地工作站上设置 Git。
   b. 使用 IAM 控制台为您的 IAM 用户生成 Git 凭证。
   c. 将项目的 CodeCommit 存储库克隆到本地工作站上。
3. 在左侧导航栏中，选择项目，返回到项目概述。

步骤 5：向 Web 服务添加逻辑

在此步骤中，将使用本地工作站向 Web 服务添加逻辑。具体来说，将添加一个 Lambda 函数，然后将其连接到 API Gateway 中的 API。

1. 在本地工作站上，转到包含克隆的源代码存储库的目录。
2. 在此目录中，创建名为 `hello.py` 的文件。添加以下代码，然后保存文件：

```
import json

def handler(event, context):
  data = {
    'output': 'Hello ' + event["pathParameters"]["name"]
  }
  return {
    'statusCode': 200,
    'body': json.dumps(data),
    'headers': {'Content-Type': 'application/json'}
  }
```

   上述代码将输出字符串 `Hello` 及调用方发送到此函数的字符串。

3. 在同一目录中，打开 `template.yml` 文件。将以下代码添加到文件末尾，然后保存文件：

```
  Hello:
    Type: AWS::Serverless::Function
    Properties:
      FunctionName: !Sub 'awscodestar-${ProjectId}-lambda-Hello'
      Handler: hello.handler
      Runtime: python3.7
      Role:
        Fn::GetAtt:
        - LambdaExecutionRole
        - Arn
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /hello/{name}
            Method: get
```

AWS SAM 将使用此代码在 Lambda 中创建一个函数，向 API Gateway 中的 API 添加一个新方法和路径，然后将此方法和路径连接到新函数。

> **Note**
>
> 上述代码的缩进很重要。如果添加的代码与所示代码有任何差异，都可能无法正确构建项目。

4. 运行 git add .，将文件更改添加到克隆存储库的暂存区域。不要忘记圆点 (.)，它表示添加所有更改过的文件。

> **Note**
>
> 如果使用 Visual Studio 或 Eclipse 而不是命令行，则使用 Git 的说明可能有所不同。请参阅 Visual Studio 或 Eclipse 文档。

5. 运行 git commit -m "Added hello.py and updated template.yaml." 以提交克隆存储库中的暂存文件
6. 运行 git push 以将提交推送到远程存储库。

> **Note**
>
> 系统可能会提示您输入之前为您生成的登录凭证。要避免每次与远程存储库交互时都显示提示，可以考虑安装和配置 Git 凭证管理器。例如，在 macOS 或 Linux 上，可以在终端中运行 git config credential.helper 'cache --timeout 900'，指示显示提示的间隔不得短于 15 分钟。或者，也可以运行 git config credential.helper 'store --file ~/.git-credentials' 永久关闭提示。Git 将凭证以明文形式存储在主目录中的纯文本文件中。有关更多信息，请参阅 Git 网站上的 Git 工具 - 凭证存储。

AWS CodeStar 检测到推送后，将指示 CodePipeline 使用 CodeBuild 和 AWS SAM 自动重新构建和重新部署 Web 服务。您可以在管道页面上查看部署进度。

AWS SAM 为新函数提供名称 awscodestar-my-sam-project-lambda-Hello-*RANDOM_ID*，其中：

- my-sam-project 是项目的 ID。
- Hello 是 `template.yaml` 文件中指定的函数 ID。
- *RANDOM_ID* 是一个随机 ID，AWS SAM 向函数分配此 ID 来确保唯一性。

步骤 6：测试增强型 Web 服务

在此步骤中，将测试 AWS CodeStar 基于在上一步中添加的逻辑构建和部署的增强型 Web 服务。

1. 在项目仍在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择管道。
2. 确保管道已再次运行，且源、构建和部署阶段显示已成功，然后再继续。这可能需要花几分钟的时间。

> **Note**
>
> 如果任何阶段显示失败，请参阅以下内容获取问题排查帮助：
>
> - 对于源阶段，请参阅 AWS CodeCommit 用户指南中的 AWS CodeCommit 问题排查。
> - 对于构建阶段，请参阅 AWS CodeBuild 用户指南中的 AWS CodeBuild 问题排查。
> - 对于部署阶段，请参阅 AWS CloudFormation 用户指南中的 AWS CloudFormation 问题排查。
> - 有关其他问题，请参阅故障排除 AWS CodeStar。

3. 选择查看应用程序。

   在 Web 浏览器中打开的新标签页上，Web 服务显示以下响应输出：

   ```
   {"output": "Hello World", "timestamp": "2017-08-30T15:53:42.682839"}
   ```

4. 在该标签的地址框中，将路径 **/hello/** 和您的名字添加到 URL 末尾（例如，https://***API_ID***.execute-api.***REGION_ID***.amazonaws.com/Prod/hello/***YOUR_FIRST_NAME***），然后按 Enter。

如果您的名字是 Mary，Web 服务将显示以下响应输出：

```
{"output": "Hello Mary"}
```

步骤 7：向 Web 服务添加单元测试

在此步骤中，将使用本地工作站添加 AWS CodeStar 对 Web 服务运行的测试。此测试将取代之前做过的手动测试。

1. 在本地工作站上，转到包含克隆的源代码存储库的目录。
2. 在此目录中，创建名为 `hello_test.py` 的文件。添加以下代码，然后保存文件。

```
from hello import handler

def test_hello_handler():

  event = {
    'pathParameters': {
      'name': 'testname'
    }
  }

  context = {}

  expected = {
    'body': '{"output": "Hello testname"}',
    'headers': {
      'Content-Type': 'application/json'
    },
    'statusCode': 200
  }

  assert handler(event, context) == expected
```

此测试检查 Lambda 函数输出是否为预期格式。如果是，则测试成功。否则，测试失败。

3. 在同一目录中，打开 `buildspec.yml` 文件。使用以下代码替换文件内容，然后保存文件。

```
version: 0.2

phases:
   install:
      runtime-versions:
         python: 3.7

      commands:
```

```
         - pip install pytest
         # Upgrade AWS CLI to the latest version
         - pip install --upgrade awscli


   pre_build:
      commands:
         - pytest

   build:
      commands:
         # Use AWS SAM to package the application by using AWS CloudFormation
         - aws cloudformation package --template template.yml --s3-bucket
 $S3_BUCKET --output-template template-export.yml

         # Do not remove this statement. This command is required for AWS CodeStar
 projects.
         # Update the AWS Partition, AWS Region, account ID and project ID in the
 project ARN on template-configuration.json file so AWS CloudFormation can tag
 project resources.
         - sed -i.bak 's/\$PARTITION\$/'${PARTITION}'/g;s/\$AWS_REGION
\$/'${AWS_REGION}'/g;s/\$ACCOUNT_ID\$/'${ACCOUNT_ID}'/g;s/\$PROJECT_ID\
$/'${PROJECT_ID}'/g' template-configuration.json

artifacts:
   type: zip
   files:
      - template-export.yml
      - template-configuration.json
```

此构建规范指示 CodeBuild 向构建环境中安装 Python 测试框架 pytest。CodeBuild 使用 pytest 运行单元测试。构建规范的其余部分与以前相同。

4. 使用 Git 将这些更改推送到远程存储库。

```
git add .

git commit -m "Added hello_test.py and updated buildspec.yml."

git push
```

步骤 8：查看单元测试结果

在此步骤中，将看到单元测试成功还是失败。

1. 在项目仍在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择管道。
2. 请确保管道已再次运行，然后再继续。这可能需要花几分钟的时间。

   如果单元测试成功，则构建阶段显示已成功。
3. 要查看单元测试结果的详细信息，请在构建阶段中选择 CodeBuild 链接。
4. 在 CodeBuild 控制台中，在构建项目: my-sam-project 页面上的构建历史记录中，选择表的构建运行列中的链接。
5. 在 my-sam-project:***BUILD_ID*** 页面的构建日志中，选择查看完整日志链接。
6. 在 Amazon CloudWatch Logs 控制台中，在日志输出中查找类似以下内容的测试结果。在以下测试结果中，测试通过：

   ```
   ...
   ============================= test session starts ==============================
   platform linux2 -- Python 2.7.12, pytest-3.2.1, py-1.4.34, pluggy-0.4.0
   rootdir: /codebuild/output/src123456789/src, inifile:
   collected 1 item

   hello_test.py .

   ========================== 1 passed in 0.01 seconds ===========================
   ...
   ```

   如果测试失败，日志输出中应包含详细信息，可帮助您进行问题排查。

第 9 步：清除

在此步骤中，将清理项目，以免继续产生与此项目相关的费用。

如果需要继续使用此项目，可以跳过这一步，但您的 AWS 账户可能会继续计费。

1. 在项目仍在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择设置。
2. 在项目详细信息页上，选择删除项目。
3. 输入 **delete**，让删除资源框处于选中状态，然后选择删除。

> **Important**
>
> 如果清除此框，将从 AWS CodeStar 中删除项目记录，但项目的许多 AWS 资源都将保留。您的 AWS 账户可能需要继续支付相应费用。

如果仍有 AWS CodeStar 为此账户创建的 Amazon S3 存储桶，请执行以下步骤将其删除：

1. 通过以下网址打开 Amazon S3 控制台：https://console.aws.amazon.com/s3/。
2. 在存储桶列表中，选择 aws-codestar-***REGION_ID*-*ACCOUNT_ID***-my-sam-project--pipe 旁边的图标，其中：
   - *REGION_ID* 是刚刚删除的项目的 AWS 区域。
   - *ACCOUNT_ID* 是您的 AWS 账户 ID。
   - my-sam-project 是刚刚删除的项目的 ID。
3. 选择清空存储桶。输入存储桶名称，然后选择确认。
4. 选择删除存储桶。输入存储桶名称，然后选择确认。

后续步骤

至此，您已完成本教程，建议您查看以下资源：

- 开始使用 AWS CodeStar 教程使用一个创建和部署基于 Node.js 的 Web 应用程序的项目，此 Web 应用程序运行在 Amazon EC2 实例上。
- AWS CodeStar 项目模板 介绍可以创建的其他类型的项目。
- 与 AWS CodeStar 团队协作 介绍如何邀请他人参与您的项目。

教程：使用 AWS CLI 在 AWS CodeStar 中创建项目

本教程向您介绍如何使用 AWS CLI 通过示例源代码和示例工具链模板创建 AWS CodeStar 项目。AWS CodeStar 预配置 AWS CloudFormation 工具链模板中指定的 AWS 基础设施和 IAM 资源。项目管理工具链资源以构建和部署源代码。

AWS CodeStar 使用 AWS CloudFormation 构建和部署示例代码。此示例代码创建托管在 AWS Lambda 中并可通过 Amazon API Gateway 进行访问的 Web 服务。

先决条件：

- 完成 设置 AWS CodeStar 中的步骤。
- 您必须已创建 Amazon S3 存储桶。在本教程中，您将示例源代码和工具链模板上传到此位置。

> **Note**
>
> 您的 AWS 账户可能产生与本教程相关的费用，包括 AWS CodeStar 使用 AWS 服务的费用。有关更多信息，请参阅 AWS CodeStar 定价。

主题

步骤 1：下载并查看示例源代码

在本教程中，有一个 zip 文件可供下载。它包含 Lambda 计算平台上的 Node.js 示例应用程序的示例源代码。将源代码放在存储库中后，其文件夹和文件如下所示：

```
tests/
app.js
buildspec.yml
index.js
package.json
README.md
template.yml
```

在示例源代码中表示了以下项目元素：

- `tests/`：为此项目的 CodeBuild 项目设置的单元测试。此文件夹包含在示例代码中，但它不是创建项目所必需的。
- `app.js`：项目的应用程序源代码。

- `buildspec.yml`：CodeBuild 资源的构建阶段的构建说明。此文件是包含 CodeBuild 资源的工具链模板所必需的。
- `package.json`：应用程序源代码的依赖项信息。
- `README.md`：包含在所有 AWS CodeStar 项目中的项目自述文件。此文件包含在示例代码中，但它不是创建项目所必需的。
- `template.yml`：包含在所有 AWS CodeStar 项目中的基础设施模板文件或 SAM 模板文件。这不同于您在本教程后面上传的工具链 template.yml。此文件包含在示例代码中，但它不是创建项目所必需的。

步骤 2：下载示例工具链模板

为本教程提供的示例工具链模板创建存储库 (CodeCommit)、管道 (CodePipeline) 和构建容器 (CodeBuild) 并使用 AWS CloudFormation 将您的源代码部署到 Lambda 平台。除了这些资源之外，还存在可用于限定运行时环境的权限范围的 IAM 角色、CodePipeline 用于存储您的部署项目的 Amazon S3 存储桶和用于在您将代码推送到存储库时触发管道部署的 CloudWatch Events 规则。为了符合 AWS IAM 最佳实践，请缩小此示例中定义的工具链角色的策略范围。

下载并解压缩 YAML 格式的示例 AWS CloudFormation 模板。

在本教程后面运行 create-project 命令时，此模板将在 AWS CloudFormation 中创建以下自定义工具链资源。有关在本教程中创建的资源的更多信息，请参阅 AWS CloudFormation 用户指南 中的以下主题：

- AWS::CodeCommit::Repository AWS CloudFormation 资源创建 CodeCommit 存储库。
- AWS::CodeBuild::Project AWS CloudFormation 资源创建 CodeBuild 构建项目。
- AWS::CodeDeploy::Application AWS CloudFormation 资源创建 CodeDeploy 应用程序。
- AWS::CodePipeline::Pipeline AWS CloudFormation 资源创建 CodePipeline 管道。
- AWS::S3::Bucket AWS CloudFormation 资源创建您管道的构件存储桶。
- AWS::S3::BucketPolicy AWS CloudFormation 资源为您管道的构件存储桶创建构件存储桶策略。
- AWS::IAM::Role AWS CloudFormation 资源创建 CodeBuild IAM 工作线程角色，该角色可向 AWS CodeStar 授予管理您的 CodeBuild 构建项目的权限。
- AWS::IAM::Role AWS CloudFormation 资源创建 CodePipeline IAM 工作线程角色，该角色可向 AWS CodeStar 授予创建您的管道的权限。
- AWS::IAM::Role AWS CloudFormation 资源创建 AWS CloudFormation IAM 工作线程角色，该角色可向 AWS CodeStar 授予创建您的资源堆栈的权限。

- AWS::IAM::Role AWS CloudFormation 资源创建 AWS CloudFormation IAM 工作线程角色，该角色可向 AWS CodeStar 授予创建您的资源堆栈的权限。
- AWS::IAM::Role AWS CloudFormation 资源创建 AWS CloudFormation IAM 工作线程角色，该角色可向 AWS CodeStar 授予创建您的资源堆栈的权限。
- AWS::Events::Rule AWS CloudFormation 资源创建监控您的存储库的推送事件的 CloudWatch Events 规则。
- AWS:::IAM::Role AWS CloudFormation 资源用于创建 CloudWatch Events IAM 角色。

步骤 3：在 AWS CloudFormation 中测试您的工具链模板

在上传您的工具链模板之前，可以在 AWS CloudFormation 中测试您的工具链模板并解决任何错误。

1. 将更新的模板保存到本地计算机，并打开 AWS CloudFormation 控制台。选择创建堆栈。您应在列表中看到新资源。
2. 查看您的堆栈中是否存在任何堆栈创建错误。
3. 在测试完成后，请删除堆栈。

> **Note**
>
> 请确保删除了您的堆栈和在 AWS CloudFormation 中创建的所有资源。否则，当您创建项目时，可能会遇到资源名称已使用的错误。

步骤 4：上传您的源代码和工具链模板

要创建 AWS CodeStar 项目，必须首先将您的源代码打包为 .zip 文件，然后将它放在 Amazon S3 中。AWS CodeStar 将使用这些内容初始化您的存储库。当您在 AWS CLI 中运行命令以创建项目时，在输入文件中指定此位置。

您还必须上传 `toolchain.yml` 文件并将它放在 Amazon S3 中。当您在 AWS CLI 中运行命令以创建项目时，在输入文件中指定此位置

上传您的源代码和工具链模板

1. 以下示例文件结构显示已准备好压缩和上传的源文件和工具链模板。示例代码包括 `template.yml` 文件。请记住，此文件与 `toolchain.yml` 文件不同。

```
ls
src toolchain.yml

ls src/
README.md    app.js        buildspec.yml    index.js    package.json
 template.yml    tests
```

2. 创建源代码文件的 .zip。

```
cd src; zip -r "../src.zip" *; cd ../
```

3. 使用 cp 命令并包含文件作为参数。

   以下命令将 .zip 文件和 `toolchain.yml` 上传到 Amazon S3。

```
aws s3 cp src.zip s3://MyBucket/src.zip
aws s3 cp toolchain.yml s3://MyBucket/toolchain.yml
```

配置您的 Amazon S3 存储桶以共享您的源代码

- 因为您将源代码和工具链存储在 Amazon S3 中，所以可以使用 Amazon S3 存储桶策略和对象 ACL 来确保其他 IAM 用户或 AWS 账户可以从您的示例创建项目。AWS CodeStar 可确保创建自定义项目的任何用户都有权访问要使用的工具链和源。

  要允许任何人使用您的示例，请运行以下命令：

```
aws s3api put-object-acl --bucket MyBucket --key toolchain.yml --acl public-read
aws s3api put-object-acl --bucket MyBucket --key src.zip --acl public-read
```

第 5 步：在 AWS CodeStar 中创建项目

使用以下步骤创建您的项目。

> **Important**
>
> 确保在 AWS CLI 中配置首选 AWS 区域。在 AWS CLI 中配置的 AWS 区域中创建您的项目。

1. 运行 create-project 命令并包含 `--generate-cli-skeleton` 参数：

```
aws codestar create-project --generate-cli-skeleton
```

输出中将显示 JSON 格式的数据。将数据复制到本地计算机上或安装 AWS CLI 的实例上某位置处的文件（如 *input.json*）中。按照下面所示修改复制的数据，并保存您的结果。为存储桶名称为 myBucket 的名为 MyProject 的项目配置此输入文件。

- 确保提供 `roleArn` 参数。对于自定义模板，如本教程中的示例模板，您必须提供角色。此角色必须拥有创建步骤 2：下载示例工具链模板中指定的所有资源的权限。
- 确保您在 `stackParameters` 下提供 `ProjectId` 参数。为本教程提供的示例模板需要此参数。

```
{
    "name": "MyProject",
    "id": "myproject",
    "description": "Sample project created with the CLI",
    "sourceCode": [
        {
            "source": {
                "s3": {
                    "bucketName": "MyBucket",
                    "bucketKey": "src.zip"
                }
            },
            "destination": {
                "codeCommit": {
                    "name": "myproject"

                }
            }
        }
    ],
    "toolchain": {
        "source": {
            "s3": {
                "bucketName": "MyBucket",
                "bucketKey": "toolchain.yml"
            }
        },
```

```
        "roleArn": "role_ARN",
        "stackParameters": {
            "ProjectId": "myproject"
        }
    }
}
```

2. 切换到包含您刚才保存的文件的目录，然后再次运行 create-project 命令。包含 `--cli-input-json` 参数。

```
aws codestar create-project --cli-input-json file://input.json
```

3. 如果成功，输出中将显示与以下内容类似的数据：

```
{
    "id": "project-ID",
    "arn": "arn"
}
```

- 输出包含有关新项目的信息：
  - `id` 值表示项目 ID。
  - `arn` 值表示项目的 ARN。

4. 使用 describe-project 命令检查您的项目创建状态。包含 `--id` 参数。

```
aws codestar describe-project --id <project_ID>
```

与以下内容类似的数据将显示在输出中：

```
{
    "name": "MyProject",
    "id": "myproject",
    "arn": "arn:aws:codestar:us-east-1:account_ID:project/myproject",
    "description": "",
    "createdTimeStamp": 1539700079.472,
    "stackId": "arn:aws:cloudformation:us-east-1:account_ID:stack/awscodestar-
myproject/stack-ID",
    "status": {
        "state": "CreateInProgress"
    }
}
```

- 输出包含有关新项目的信息：
  - `id` 值表示唯一项目 ID。
  - `state` 值表示项目创建状态，如 `CreateInProgress` 或 `CreateComplete`。

在创建项目时，您可以从命令行或您常用的 IDE 为项目存储库添加成员或配置访问权限。

教程：在 AWS CodeStar 中创建 Alexa 技能项目

AWS CodeStar 是 AWS 上的一项基于云的开发服务，它为您提供在 AWS 上快速开发、构建和部署应用程序所需的各种工具。借助 AWS CodeStar，您可以在几分钟内建立完整的持续交付工具链，从而以更快的速度开始发布代码。利用 AWS CodeStar 上的 Alexa 技能项目模板，您只需单击几下即可在您的 AWS 账户内创建简单的“Hello World”Alexa 技能。这些模板还创建一个基本部署管道，使您能够开始使用持续集成 (CI) 工作流来进行技能开发。

从 AWS CodeStar 创建 Alexa 技能的主要好处是，您可以从 AWS 中开始进行技能开发并将您的 Amazon 开发人员账户连接到项目，以便直接从 AWS 将技能部署到开发阶段。您还可以获得一个随时可用的部署 (CI) 管道，此管道具有一个包含项目的所有源代码的存储库。您可以使用首选 IDE 配置此存储库，从而使用您熟悉的工具创建技能。

先决条件

- 通过转到 https://developer.amazon.com 来创建 Amazon 开发人员账户。注册是免费的。此账户拥有您的 Alexa 技能。
- 如果您还没有 AWS 账户，请使用以下过程创建一个。

  注册 AWS

  1. 打开 https://aws.amazon.com/，然后选择创建 AWS 账户。

     > **Note**
     >
     > 如果您之前曾使用 AWS 账户根用户 凭证登录 AWS Management Console，请选择登录其他账户。如果您之前已使用 IAM 凭证登录控制台，请选择使用 AWS 账户根用户 凭证登录。选择创建新的 AWS 账户。

  2. 按照屏幕上的说明进行操作。

> **Important**
>
> 创建 Alexa 技能项目后，仅在项目存储库中进行所有编辑。我们建议您不要使用任何其他 Alexa Skills Kit 工具（例如 ASK CLI 或 ASK 开发人员控制台）直接编辑此技能。这些工具不会与项目存储库集成。使用它们会导致技能和存储库代码不同步。

步骤 1：创建项目并连接您的 Amazon 开发人员账户

在本教程中，您可使用在 AWS Lambda 上运行的 Node.js 创建技能。对于其他语言来说，大多数步骤都相同，但技能名称会有所不同。有关您选择的特定项目模板的详细信息，请参阅项目存储库中的 README.md 文件。

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。
2. 选择要在其中创建项目及其资源的 AWS 区域。Alexa 技能运行时在以下 AWS 区域可用：
   - 亚太地区（东京）
   - 欧洲（爱尔兰）
   - 美国东部（弗吉尼亚州北部）
   - 美国西部（俄勒冈州）
3. 请选择创建项目。
4. 在选择项目模板页面上：
   a. 对于应用程序类别，选择 Alexa 技能。
   b. 对于编程语言，选择 Node.js。
5. 选择您所需项对应的框。
6. 对于项目名称，输入项目的名称（例如，**`My Alexa Skill`**）。如果您使用不同的名称，请确保在本教程中始终使用它。AWS CodeStar 将为项目 ID 选择此项目的相关标识符（例如 my-alexa-skill）。如果看到不同于示例所用的项目 ID，请务必在本教程中通篇使用它。
7. 在本教程中为存储库选择 AWS CodeCommit，并且不要更改存储库名称值。
8. 选择连接 Amazon 开发人员账户以链接到用于托管技能的 Amazon 开发人员账户。如果您没有 Amazon 开发人员账户，请先创建一个账户，然后在 Amazon 开发人员处完成注册。
9. 使用 Amazon 开发人员凭证进行登录。选择允许，然后选择确认以完成连接。
10. 如果您有多个与 Amazon 开发人员账户关联的供应商 ID，请选择要用于此项目的 ID。确保您使用已分配管理员或开发人员角色的账户。

11选择下一步。

12.（可选）如果这是您第一次在此 AWS 区域中使用 AWS CodeStar，请输入希望 AWS CodeStar 用于您的 IAM 用户的显示名称和电子邮件地址。选择下一步。

13等待 AWS CodeStar 创建项目。这可能需要花几分钟的时间。请在看到项目已配置横幅后再继续。

步骤 2：在 Alexa 模拟器中测试您的技能

在第一步中，AWS CodeStar 已为您创建一个技能并已将其部署到 Alexa 技能开发阶段。接下来，您将在 Alexa 模拟器中测试该技能。

1. 在 AWS CodeStar 控制台的项目中，选择查看应用程序。这将在 Alexa 模拟器中打开一个新的选项卡。
2. 使用已在步骤 1 中连接到项目的账户的 Amazon 开发人员凭证进行登录。
3. 在测试下，选择开发以启用测试。
4. 输入 `ask hello node hello`。您的技能的默认调用名称为 `hello node`。
5. 您的技能应响应 `Hello World!`。

在 Alexa 模拟器中启用该技能后，您还可以在已注册到您的 Amazon 开发人员账户的支持 Alexa 的设备上调用它。要在设备上测试您的技能，请说 Alexa, ask hello node to say hello。

有关 Alexa 模拟器的更多信息，请参阅在开发人员控制台中测试您的技能。

步骤 3：浏览您的项目资源

作为项目创建过程的一部分，AWS CodeStar 还代表您创建了 AWS 资源。这些资源包含使用 CodeCommit 的项目存储库、使用 CodePipeline 的部署管道和 AWS Lambda 函数。您可以从导航栏访问这些资源。例如，选择存储库会显示有关 CodeCommit 存储库的详细信息。您可以在管道页面中查看管道部署状态。您可以通过选择导航栏中的概览来查看作为项目的一部分创建的 AWS 资源的完整列表。此列表包含指向每个资源的链接。

步骤 4：在您的技能的响应中进行更改

在此步骤中，您可对技能的响应进行细微更改以了解迭代周期。

1. 在导航栏中，选择存储库。选择存储库名称下的链接，您的项目存储库将在新的选项卡或窗口中打开。此存储库包含构建规范 (buildspec.yml)、AWS CloudFormation 应用程序堆栈 (template.yml)、自述文件以及采用技能包格式（项目结构）的技能的源代码。

2. 导航到文件 lambda > 自定义 > index.js（如果是 Node.js）。此文件包含使用 ASK 开发工具包的请求处理代码。
3. 选择编辑。
4. 将第 24 行中的字符串 `Hello World!` 替换为字符串 `Hello. How are you?`。
5. 向下滚动到文件末尾。输入作者姓名和电子邮件地址及可选的提交消息。
6. 选择提交更改以将更改提交到存储库。
7. 返回 AWS CodeStar 中的项目并查看管道页面。现在您应看到管道正在部署。
8. 管道完成部署后，在 Alexa 模拟器中再次测试您的技能。您的技能现在应使用 `Hello. How are you?` 进行响应

步骤 5：设置您的本地工作站以连接到您的项目存储库

之前，您已从 CodeCommit 控制台直接对源代码进行了少量更改。在此步骤中，您使用本地工作站配置项目存储库，以便能从命令行或常用 IDE 编辑和管理代码。以下步骤说明如何设置命令行工具。

1. 如有必要，导航到 AWS CodeStar 中的项目控制面板。
2. 在导航栏中，选择 IDE。
3. 在访问您的项目代码中，查看命令行界面下方的说明。
4. 按照说明完成以下任务：
   a. 从 Git 下载等网站将 Git 安装在您的本地工作站上。
   b. 安装 AWS CLI。有关信息，请参阅安装 AWS 命令行界面。
   c. 使用您的 IAM 用户访问密钥和私有密钥配置 AWS CLI。有关信息，请参阅配置 AWS CLI。
   d. 将项目的 CodeCommit 存储库克隆到本地工作站上。有关更多信息，请参阅连接到 CodeCommit 存储库。

后续步骤

本教程介绍如何开始使用基本技能。要继续您的技能开发之旅，请参阅以下资源。

- 通过在 Alexa Developers YouTube 频道上观看 Alexa 技能工作原理及其他视频来了解技能的基础知识。
- 通过查看技能包格式、技能清单架构和交互模式架构的相关文档来了解您的技能的各种组件。
- 通过查看 Alexa Skills Kit 和 ASK 开发工具包的相关文档来将您的想法变成技能。

教程：使用 GitHub 源代码库创建项目

使用 AWS CodeStar，您可以设置存储库，以便与项目团队一起创建、审阅和合并拉取请求。

在本教程中，您将创建一个项目，其中包含 GitHub 存储库中的示例 Web 应用程序源代码、用于部署更改的管道以及在云中托管应用程序的 EC2 实例。项目创建完成后，本教程将向您展示如何创建和合并 GitHub 拉取请求，以更改您的 Web 应用程序的主页。

主题

步骤 1：创建项目并创建您的 GitHub 存储库

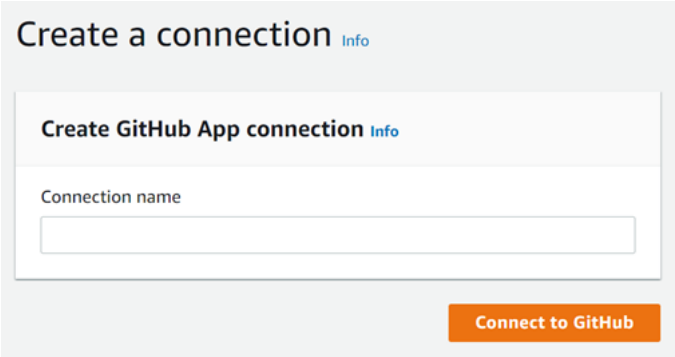
在此步骤中，使用控制台创建项目并创建与新 GitHub 存储库的连接。要访问您的 GitHub 存储库，您需要创建一个 AWS CodeStar 用于管理 GitHub 授权的连接资源。创建项目后，系统会为您配置项目的其他资源。

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。
2. 选择要在其中创建项目及其资源的 AWS 区域。
3. 在 AWS CodeStar 页面上，选择创建项目。
4. 在选择项目模板页面上，选中 Web 应用程序、Node.js 和 Amazon EC2 复选框。随后，从可用于此选项集的模板中进行选择。

   有关更多信息，请参阅AWS CodeStar 项目模板。
5. 选择下一步。
6. 对于项目名称，输入项目的名称（例如，**`MyTeamProject`**）。如果您使用其他名称，请确保在本教程中通篇使用它。
7. 在项目存储库下，选择 GitHub。
8. 如果您选择 GitHub，则需要选择或创建连接资源。如果您已有连接，请在搜索栏中选择该连接。否则，您将在此处创建一个新连接。选择连接到 GitHub。

   创建连接页面随即显示。

> **Note**
>
> 要创建连接，您必须拥有 GitHub 账户。您必须是组织所有者才能为组织创建连接。



a. 在创建 GitHub 应用程序连接的连接名称中，输入连接的名称。选择连接到 GitHub。

   连接到 GitHub 页面将出现，并显示 GitHub 应用程序字段。

b. 在 GitHub 应用程序下，选择一个应用程序安装或选择安装新应用程序来创建一个应用程序安装。

> **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 AWS Connector for GitHub 应用程序，选择它并跳过此步骤。

c. 在安装 AWS Connector for GitHub 页面中，选择要在其中安装应用程序的账户。

> **Note**
>
> 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。

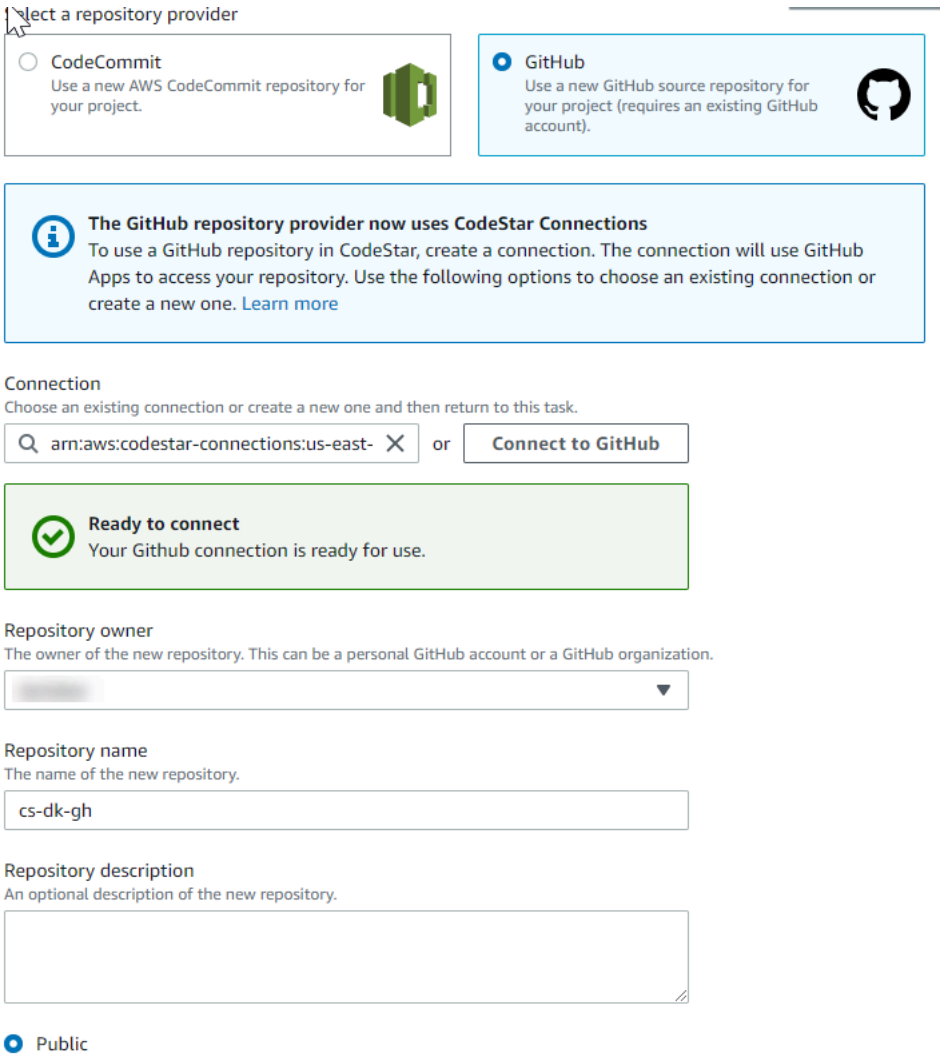
d. 如果显示确认密码以继续页面，请输入您的 GitHub 密码，然后选择登录。

e. 在安装 AWS Connector for GitHub 页面中，保留所有默认设置，然后选择安装。

f. 在连接到 GitHub 页面上，新安装的连接 ID 将显示在 GitHub 应用程序中。

   成功创建连接后，在 CodeStar“创建项目”页面中，将显示准备连接消息。

> **Note**
>
> 您可以在“开发人员工具”控制台的“设置”下查看连接。有关更多信息，请参阅开始使用连接。



g. 对于存储库所有者，请选择 GitHub 组织或您的个人 GitHub 账户。

h. 对于存储库名称，请接受默认的 GitHub 存储库名称，或输入其他名称。

i. 选择 公共或私有。

> **Note**
>
> 如果您想使用 AWS Cloud9 作为开发环境，则必须选择一个公共存储库。

j. （可选）对于存储库描述，请为 GitHub 存储库输入描述。

9. 如果您的项目已部署到 Amazon EC2 实例并且您想进行更改，请在 Amazon EC2 配置中配置您的 Amazon EC2 实例。例如，您可以从项目的可用实例类型中进行选择。

   在密钥对中，选择在 步骤 4：为 AWS CodeStar 项目创建 Amazon EC2 密钥对 创建的 Amazon EC2 密钥对。选择我确认我有权访问私钥文件。

10. 选择下一步。
11. 查看资源和配置详细信息。
12. 选择 Next 或 Create project。（显示的选择取决于您的项目模板。）

    等待几分钟，项目会创建完毕。

13. 创建项目后，选择查看应用程序以查看您的 Web 应用程序。

步骤 2：查看源代码

在此步骤中，您将查看源代码和可用于源存储库的工具。

1. 在项目的导航栏中，选择存储库。

   要在 GitHub 中查看提交列表，请选择查看提交。这将在 GitHub 中打开您的提交历史记录。

   要查看问题，请选择项目的问题选项卡。要在 GitHub 中创建新问题，请选择创建 GitHub 问题。这将在 GitHub 中打开您的存储库问题表单。

2. 在存储库选项卡下，选择存储库名称下的链接，您的项目存储库将在新的选项卡或窗口中打开。此存储库包含您项目的源代码。

步骤 3：创建 GitHub 拉取请求

在此步骤中，您将对源代码进行细微更改并创建拉取请求。

1. 在 GitHub 中，在存储库中创建一个新的功能分支。选择主分支下拉字段，然后在名为 `feature-branch` 的字段中输入新的分支。选择创建新分支。该分支已为您创建并签出。

2. 在 GitHub 中，在 `feature-branch` 分支中进行更改。打开公共文件夹并打开 `index.html` 文件。
3. 如要在 GitHub 中创建拉取请求，请在 AWS CodeStar 控制台的拉取请求下，选择创建拉取请求。这将在 GitHub 中打开您的存储库拉取请求表单。在 GitHub 中，选择铅笔图标来编辑文件。

   出现 `Congratulations!` 之后，添加字符串 `Well done, <name>!` 并将 `<name>` 替换为您的名字。选择提交更改。此时更改已提交到您的功能分支。
4. 在 AWS CodeStar 控制台中，选择您的项目。选择存储库选项卡。在“拉取请求”中，选择创建拉取请求。

   表格将在 GitHub 中打开。将主分支留在基础分支中。在比较对象中，选择您的功能分支。查看差异。
5. 在 GitHub 中，选择创建拉取请求。系统将创建一个名为 Update index.html 的拉取请求。
6. 在 AWS CodeStar 控制台中，查看新的拉取请求。选择合并更改，将更改提交到存储库，并将拉取请求与存储库的主分支合并。
7. 返回 AWS CodeStar 中的项目并查看管道页面。现在您应看到管道正在部署。
8. 创建项目后，选择查看应用程序以查看您的 Web 应用程序。

AWS CodeStar 项目模板

通过 AWS CodeStar 项目模板，您可以从一个示例应用程序开始，使用事先创建的 AWS 资源部署该应用程序，以支持您的开发项目。当您选择一个 AWS CodeStar 项目模板时，系统将会为您预配置应用程序类型、编程语言和计算平台。创建具有 Web 应用程序、Web 服务，Alexa 技能和静态网页的项目后，您便可将示例应用程序替换为自己的应用程序。

在 AWS CodeStar 创建项目后，您可以修改支持应用程序交付的 AWS 资源。AWS CodeStar 与 AWS CloudFormation 配合使用，允许您使用代码在云创建支持服务和服务器/无服务器平台。借助 AWS CloudFormation，您可将整个基础设施建模在一个文本文件中。

主题

AWS CodeStar 项目文件和资源

AWS CodeStar 项目是源代码和为部署代码而创建的资源的组合。帮助您构建、发布和部署您的代码的资源集合称为工具链资源。在创建项目期间，AWS CloudFormation 模板将工具链资源预配置在一个持续集成/持续部署 (CI/CD) 管道中。

根据您创建 AWS 资源的经验水平，您可以通过两种方式使用 AWS CodeStar 创建项目：

- 当您使用控制台创建项目时，AWS CodeStar 会为您创建具有存储库的工具链资源，并用示例应用程序代码和项目文件填充存储库。使用控制台可根据一组预配置的项目选项快速设置示例项目。
- 当您使用 CLI 创建项目时，您将提供创建工具链资源的 AWS CloudFormation 模板和应用程序源代码。使用 CLI 时允许 AWS CodeStar 从模板创建项目，然后用示例代码填充存储库。

一个提供单一管理点的 AWS CodeStar 项目。您可以使用控制台中的创建项目向导设置示例项目。然后，可将该项目用作管理您团队的权限和资源的协作平台。有关更多信息，请参阅什么是 AWS CodeStar？。当您使用控制台创建项目时，会将您的源代码作为示例代码提供，并会为您创建 CI/CD 工具链资源

在控制台中创建项目时，AWS CodeStar 预配置以下资源：

- GitHub 或 CodeCommit 中的代码存储库。
- 项目存储库中的 `README.md` 文件，提供文件和目录的详细信息。
- 项目存储库中的 `template.yml` 文件，存储应用程序运行时堆栈的定义。使用此文件添加或修改不属于工具链资源的项目资源，如用于通知、数据库支持、监控和跟踪的 AWS 资源。
- 与管道一起创建的 AWS 服务和资源，如 Amazon S3 构件存储桶、Amazon CloudWatch Events 及相关的服务角色。
- 具有完整源代码和公有 HTTP 终端节点的一个运行中的示例应用程序。
- AWS 计算资源，基于 AWS CodeStar 项目模板类型：
  - Lambda 函数。
  - 一个 Amazon EC2 实例。
  - 一个 AWS Elastic Beanstalk 环境。
- 自太平洋夏令时 2018 年 12 月 6 日起：
  - 一个权限边界，它是用于控制对项目资源的访问的专用 IAM 策略。默认情况下，权限边界附加到示例项目中的角色。有关更多信息，请参阅工作线程角色的 IAM 权限边界。
  - 用于使用 AWS CloudFormation 创建项目资源的 AWS CloudFormation IAM 角色，其中包括所有 AWS CloudFormation 支持的资源（包括 IAM 角色）的权限。
  - 工具链 IAM 角色。
  - 应用程序堆栈中定义的 Lambda 的执行角色，您可以修改它。
- 在太平洋夏令时 2018 年 12 月 6 日前：
  - 用于创建项目资源的 AWS CloudFormation IAM 角色，支持一组有限的 AWS CloudFormation 资源。
  - 用于创建 CodePipeline 资源的 IAM 角色。
  - 用于创建 CodeBuild 资源的 IAM 角色。
  - 用于创建 CodeDeploy 资源的 IAM 角色（如果适用于您的项目类型）。
  - 用于创建 Amazon EC2 Web 应用程序的 IAM 角色（如果适用于您的项目类型）。
  - 用于创建 CloudWatch Events 资源的 IAM 角色。
  - Lambda 的执行角色，可动态修改以包含部分资源集。

该项目包括详细信息页面，其中显示了状态，包括指向团队管理的链接、指向 IDE 或存储库设置说明的链接以及存储库中源代码更改的提交历史记录。此外，您还可以选择工具来连接到外部问题跟踪工具（如 Jira）。

入门：选择项目模板

在控制台中选择 AWS CodeStar 项目时，您将从具有示例代码和资源的一组预配置选项中进行选择，以便快速入门。这些选项称为项目模板。每个 AWS CodeStar 项目模板由编程语言、应用程序类型和计算平台组成。该项目模板由您选择的组合确定。

选择模板计算平台

每个模板都会配置以下计算平台类型之一：

- 当您选择 AWS Elastic Beanstalk 项目时，您将部署到云中 Amazon Elastic Compute Cloud 实例上的 AWS Elastic Beanstalk 环境。
- 当您选择 Amazon EC2 项目时，AWS CodeStar 会创建 Linux EC2 实例以在云中托管您的应用程序。您的项目团队成员可以访问这些实例，并且您的团队使用您提供给 SSH 的密钥对连接到您的 Amazon EC2 实例。AWS CodeStar 还具有使用团队成员权限来管理密钥对连接的托管 SSH。
- 当您选择 AWS Lambda 时，AWS CodeStar 会创建通过 Amazon API Gateway 访问的无服务器环境，其中没有实例或服务器，无需进行维护。

选择模板应用程序类型

每个模板会配置以下应用程序类型之一：

- Web 服务

  Web 服务用于在后台运行的任务（如调用 API）。在 AWS CodeStar 创建示例 Web 服务项目后，您可以选择终端节点 URL 查看 Hello World 输出，但此应用程序类型的主要用途不是用作用户界面 (UI)。此类别中的 AWS CodeStar 项目模板支持使用 Ruby、Java、ASP.NET、PHP、Node.js 等编程语言进行的开发。
- Web 应用程序

  Web 应用程序提供 UI。在 AWS CodeStar 创建示例 Web 应用程序项目后，您可以选择终端节点 URL 查看交互式 Web 应用程序。此类别中的 AWS CodeStar 项目模板支持使用 Ruby、Java、ASP.NET、PHP、Node.js 等编程语言进行的开发。
- 静态网页

  如果您希望项目用于 HTML 网站，请选择此模板。此类别中的 AWS CodeStar 项目模板支持使用 HTML5 进行的开发。

- Alexa 技能

  如果您需要面向具有 AWS Lambda 函数的 Alexa 技能的项目，请选择此模板。当您创建技能项目时，AWS CodeStar 会返回可用作服务端点的 Amazon 资源名称 (ARN)。有关更多信息，请参阅托管自定义技能作为 AWS Lambda 函数。

  > **Note**
  >
  > 仅美国东部（弗吉利亚北部）、美国西部（俄勒冈）、欧洲（爱尔兰）和亚太地区（东京）区域支持 Alexa 技能的 Lambda 函数。

- Config 规则

  如果您希望项目用于一个 AWS Config 规则（可在您账户的 AWS 资源之间自动执行规则），请选择此模板。该函数返回可用作规则的服务终端节点的 ARN。

选择模板编程语言

当您选择项目模板时，将选择一种编程语言，如 Ruby、Java、ASP.NET、PHP、Node.js 等。

如何对您的 AWS CodeStar 项目进行更改

您可以通过修改以下内容来更新您的项目：

- 用于您应用程序的示例代码和编程语言资源。
- 存储和部署您应用程序的基础设施所包含的资源（操作系统、支持应用程序和服务、部署参数和云计算平台）。您可以在 `template.yml` 文件中修改应用程序资源。这是对您应用程序的运行时环境进行建模的 AWS CloudFormation 文件。

> **Note**
>
> 如果您使用的是 Alexa 技能 AWS CodeStar 项目，则不能对 AWS CodeStar 源存储库之外的技能（CodeCommit 或 GitHub）进行更改。如果您在 Alexa 开发人员门户中编辑技能，则更改可能在源存储库中不可见，并且两个版本将不同步。

更改应用程序源代码并推送更改

要修改示例源代码、脚本及其他应用程序源文件，请通过以下方法编辑源存储库中的文件：

- 在 CodeCommit 或 GitHub 中使用编辑模式。
- 在 IDE（如 AWS Cloud9）中打开项目。
- 在本地克隆存储库，然后提交并推送您的更改。有关信息，请参阅 步骤 4：提交更改。

使用 Template.yml 文件更改应用程序资源

无需手动修改基础设施资源，而是使用 AWS CloudFormation 对应用程序的运行时资源进行建模并加以部署。

您可以通过编辑项目存储库中的 `template.yml` 文件，在运行时堆栈中修改或添加应用程序资源（如 Lambda 函数）。您可以添加可用作 AWS CloudFormation 资源的任何资源。

要更改 AWS Lambda 函数的代码或设置，请参阅将资源添加到项目。

修改项目存储库中的 `template.yml` 文件，以添加作为应用程序资源的 AWS CloudFormation 资源的类型。当您将应用程序资源添加到 `template.yml` 文件的 `Resources` 部分时，AWS CloudFormation 和 AWS CodeStar 会为您创建资源。有关 AWS CloudFormation 资源及其必需属性的列表，请参阅 AWS 资源类型参考。有关更多信息，请参阅步骤 1：在 IAM 中编辑 CloudFormation 工作线程角色中的此示例。

借助 AWS CodeStar，您可以对应用程序运行时环境进行配置和建模，从而实施最佳做法。

如何管理更改应用程序资源的权限

当您使用 AWS CloudFormation 添加运行时应用程序资源（如 Lambda 函数）时，AWS CloudFormation 工作线程角色可以使用其已有的权限。对于某些运行时应用程序资源，您必须先手动调整 AWS CloudFormation 工作线程角色的权限，然后再编辑 `template.yml` 文件。

有关更改 AWS CloudFormation 工作线程角色的权限的示例，请参阅步骤 5：使用内联策略添加资源权限。

AWS CodeStar 最佳实践

AWS CodeStar 与多种产品和服务集成在一起。以下几节介绍了 AWS CodeStar 以及相关产品和服务的最佳实践。

主题

AWS CodeStar 资源的安全最佳实践

您应该定期对应用程序所使用的依赖项应用补丁并审查安全最佳实践。使用以下安全最佳实践来更新您的示例代码并在生产环境中维护您的项目：

- 跟踪您的框架的持续安全公告和更新。
- 在部署项目之前，请遵循为您的框架开发的最佳实践。
- 定期审查您的框架的依赖项并根据需要进行更新。
- 每个 AWS CodeStar 模板包含适用于您的编程语言的配置说明。请参阅您的项目的源存储库中的 `README.md` 文件。
- 作为隔离项目资源的最佳实践，请使用 AWS CodeStar 中的安全性 中介绍的多账户策略来管理对 AWS 资源的最低权限访问权限。

有关为依赖项设置版本的最佳实践

您的 AWS CodeStar 项目中的示例源代码使用源存储库中的 `package.json` 文件中列出的依赖项。作为最佳实践，始终将您的依赖项设置为指向特定版本。这称为固定版本。建议不要将版本设置为 `latest`，因为这会引入可能不通知即中断您的应用程序的更改。

AWS CodeStar 资源的监控和日志记录最佳实践

您可以使用 AWS 中的日志记录功能来确定用户在您的账户中进行了哪些操作，以及使用了哪些资源。日志文件显示：

- 操作的时间和日期。
- 操作的源 IP 地址。
- 由于权限不足而失败的操作。

AWS CloudTrail 可以用于记录由某个 AWS 账户发出或代表该账户发出的 AWS API 调用和相关事件。有关更多信息，请参阅使用 AWS CloudTrail 记录 AWS CodeStar API 调用。

在 AWS CodeStar 中处理项目

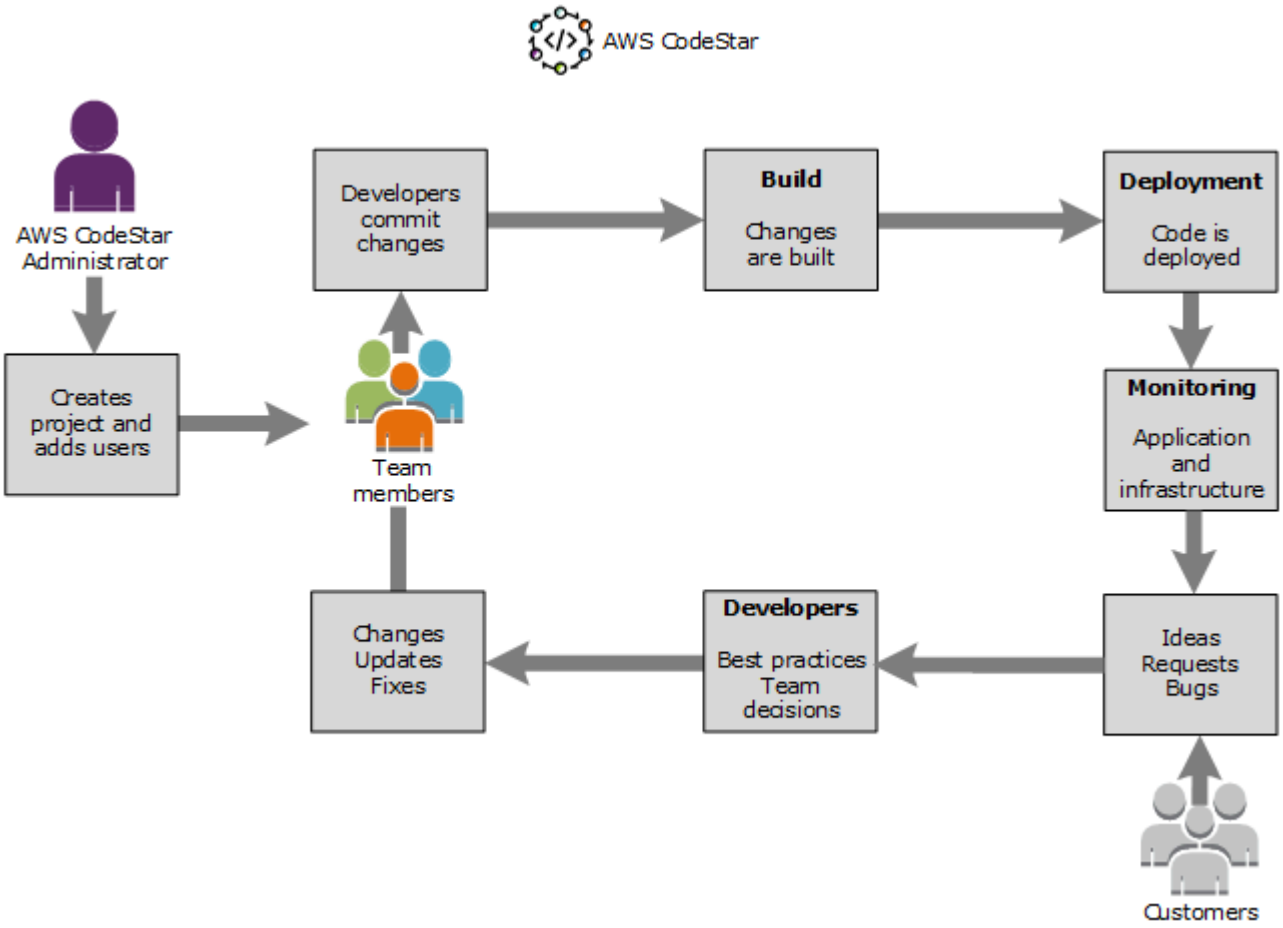
在使用 AWS CodeStar 项目模板时，您可以快速创建已使用所需资源配置的项目，包括：

- 源存储库
- 构建环境
- 部署和托管资源
- 编程语言

模板甚至包括示例源代码，因此您可以立即开始使用您的项目。

在拥有项目后，可添加或删除资源、自定义您的项目控制面板及监控进度。

下图演示了 AWS CodeStar 项目中的基本工作流程。



图中的基本工作流程展示应用了 AWSCodeStarFullAccess 策略的开发人员如何创建一个项目并向该项目添加团队成员。这些团队成员一起编写、构建、测试和部署代码。项目控制面板提供了一些工

具，可用于实时查看应用程序活动和监控生成、通过部署管道的代码流等。该团队使用“Team wiki”磁贴共享信息、最佳实践和链接。他们整合其问题跟踪软件来帮助其跟踪进度和任务。在客户提供请求和反馈时，该团队将此信息添加到项目并将此信息集成到其项目规划和开发中。随着项目的进行，该团队将添加更多团队成员来支持其代码库。

在 AWS CodeStar 中创建项目

您可使用 AWS CodeStar 控制台创建项目。如果您使用了项目模板，它将为您设置所需的资源。该模板还包括可供您用于开始编码的示例代码。

要创建项目，请以具有 `AWSCodeStarFullAccess` 策略或同等权限的 IAM 用户的身份登录 AWS Management Console。有关更多信息，请参阅设置 AWS CodeStar。

> **Note**
>
> 在完成本主题的步骤之前，您必须完成 设置 AWS CodeStar 中的步骤。

主题

在 AWS CodeStar 中创建项目（控制台）

使用 AWS CodeStar 控制台创建项目。

在 AWS CodeStar 中创建项目

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。

   确保您已登录要在其中创建项目及其资源的 AWS 区域。例如，要在美国东部（俄亥俄州）中创建项目，请确保您已选择 AWS 区域。有关 AWS CodeStar 在其中可用的 AWS 区域的信息，请参阅 AWS 一般参考中的区域和端点。
2. 在 AWS CodeStar 页面上，选择创建项目。
3. 在创建项目模板页面上，从 AWS CodeStar 项目模板的列表中选择项目类型。您可使用筛选栏缩小所选内容的范围。例如，对于要部署到 Amazon EC2 实例、用 Node.js 编写的 Web 应用程序项

目，请选中 Web 应用程序、Node.js 和 Amazon EC2 复选框。随后，从可用于此选项集的模板中进行选择。

有关更多信息，请参阅AWS CodeStar 项目模板。

4. 选择下一步。
5. 在项目名称文本输入字段中，输入项目的名称，例如*#######*。在项目 ID 中，项目的 ID 派生自此项目名称，但限制为 15 个字符。

   例如，名为*#######*的项目的默认 ID 是 *my-first-projec*。此项目 ID 是与项目关联的所有资源的名称的基础。AWS CodeStar 使用此项目 ID 作为代码存储库 URL 的一部分，以及 IAM 中相关安全访问角色和策略的名称的一部分。创建项目后，项目 ID 便无法更改。要在创建项目之前编辑项目 ID，请在项目 ID 中输入要使用的 ID。

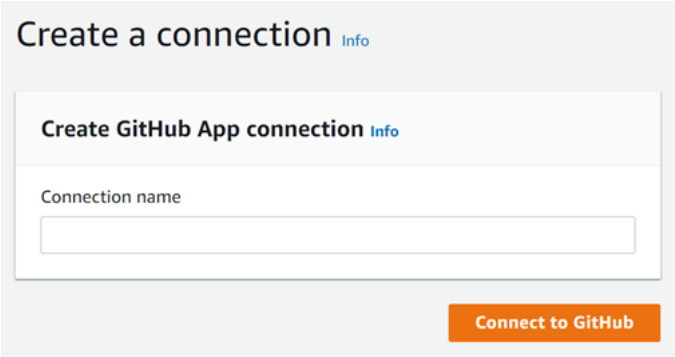
   有关项目名称和项目 ID 的限制的信息，请参阅 AWS CodeStar 中的限制。

   > **Note**
   >
   > 对于您的 AWS 账户来说，项目 ID 在某个 AWS 区域必须唯一。

6. 选择存储库提供商 AWS CodeCommit 或 GitHub。
7. 如果您选择了 AWS CodeCommit，则对于存储库名称，请接受默认的 AWS CodeCommit 存储库名称，或输入其他名称。然后跳至步骤 9。
8. 如果您选择 GitHub，则需要选择或创建连接资源。如果您已有连接，请在搜索栏中选择该连接。否则，立即创建新连接。选择连接到 GitHub。

   创建连接页面随即显示。

   > **Note**
   >
   > 要创建连接，您必须拥有 GitHub 账户。您必须是组织所有者才能为组织创建连接。



a. 在创建 GitHub 应用程序连接下连接名称的输入文本字段中，输入连接的名称。选择连接到 GitHub。

   连接到 GitHub 页面将出现，并显示 GitHub 应用程序字段。

b. 在 GitHub 应用程序下，选择一个应用程序安装或选择安装新应用程序来创建一个应用程序安装。

> **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 AWS Connector for GitHub 应用程序，选择它并跳过此步骤。

c. 在安装 AWS Connector for GitHub 页面中，选择要在其中安装应用程序的账户。

> **Note**
>
> 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。

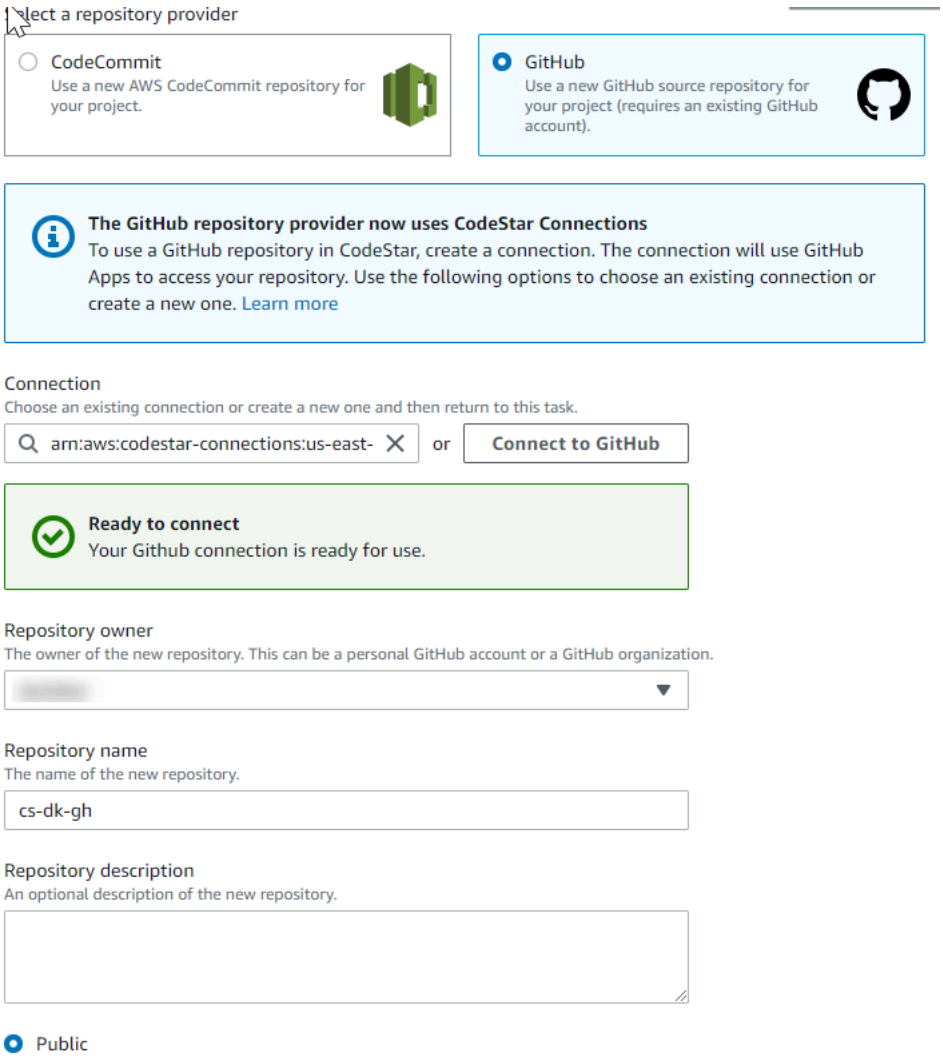
d. 如果显示确认密码以继续页面，请输入您的 GitHub 密码，然后选择登录。

e. 在安装 AWS Connector for GitHub 页面中，保留所有默认设置，然后选择安装。

f. 在连接到 GitHub 页面上，新安装的连接 ID 将显示在 GitHub 应用程序文本输入字段中。

   创建连接后，在 CodeStar“创建项目”页面中，将显示准备连接消息。

> **Note**
>
> 您可以在开发人员工具控制台的设置下查看您的连接。有关更多信息，请参阅开始使用连接。



g. 对于存储库所有者，请选择 GitHub 组织或您的个人 GitHub 账户。

h. 对于存储库名称，请接受默认的 GitHub 存储库名称，或输入其他名称。

i. 选择 公共或私有。

> **Note**
>
> 此外，如果您想使用 AWS Cloud9 作为开发环境，则必须选择公共。

j. （可选）对于存储库描述，请为 GitHub 存储库输入描述。

> **Note**
>
> 如果您选择 Alexa Skill 项目模板，则需要关联 Amazon 开发人员账户。有关如何使用 Alexa Skill 项目的更多信息，请参阅 教程：在 AWS CodeStar 中创建 Alexa 技能项目。

9. 如果您的项目已部署到 Amazon EC2 实例，并且您想进行更改，请在 Amazon EC2 配置中配置您的 Amazon EC2 实例。例如，您可以从项目的可用实例类型中进行选择。

> **Note**
>
> 不同的 Amazon EC2 实例类型将提供不同级别的计算能力，并且可能具有不同的关联成本。有关更多信息，请参阅 Amazon EC2 实例类型和 Amazon EC2 定价。
> 如果您在 Amazon Virtual Private Cloud 中创建了多个虚拟私有云 (VPC) 或多个子网，则还可选择要使用的 VPC 和子网。但是，如果您选择专用实例上不支持的 Amazon EC2 实例类型，则无法选择其实例租赁设置为专用的 VPC。
> 有关更多信息，请参阅什么是 Amazon VPC？和专用实例基本信息。

在密钥对中，选择在 步骤 4：为 AWS CodeStar 项目创建 Amazon EC2 密钥对 创建的 Amazon EC2 密钥对。选择我确认我有权访问私钥文件。

10. 选择下一步。
11. 查看资源和配置详细信息。
12. 选择 Next 或 Create project。（显示的选择取决于您的项目模板。）

    创建项目（包括存储库）可能需要几分钟时间。
13. 在项目拥有存储库后，您可以使用存储库页面来配置对它的访问权限。使用后续步骤中的链接来配置 IDE，设置问题跟踪或向项目中添加团队成员。

在创建项目时，您可以从命令行或您常用的 IDE 为项目存储库添加成员或配置访问权限。

在 AWS CodeStar 中创建项目 (AWS CLI)

AWS CodeStar 项目是源代码和为部署代码而创建的资源的组合。帮助您构建、发布和部署您的代码的资源集合称为工具链资源。在创建项目期间，AWS CloudFormation 模板将工具链资源预配置在一个持续集成/持续部署 (CI/CD) 管道中。

当您使用控制台创建项目时，将为您创建工具链模板。当您使用 AWS CLI 创建项目时，创建用于创建您的工具链资源的工具链模板。

完整的工具链需要以下推荐的资源：

1. 包含您的源代码的 CodeCommit 或 GitHub 存储库。
2. 配置为侦听存储库更改的 CodePipeline 管道。
   a. 当您使用 CodeBuild 运行单元测试或集成测试时，建议您向管道中添加构建阶段以创建构建构件。
   b. 建议您向管道中添加使用 CodeDeploy 或 AWS CloudFormation 将构建构件和源代码部署到运行时基础设施的部署阶段。

> **Note**
>
> 因为 CodePipeline 要求管道中至少具有两个阶段，并且第一个阶段必须是源阶段，所以请添加构建或部署阶段作为第二个阶段。

AWS CodeStar 工具链被定义为 CloudFormation 模板。

有关说明如何完成此任务和设置示例资源的教程，请参阅教程：使用 AWS CLI 在 AWS CodeStar 中创建项目。

先决条件：

当您创建项目时，在输入文件中提供以下参数。如果未提供以下参数，AWS CodeStar 将创建一个空项目。

- 源代码。如果此参数包含在您的请求中，则您还必须包括工具链模板。
  - 您的源代码必须包含运行项目所需的应用程序代码。
  - 您的源代码必须包含任何所需配置文件，如 CodeBuild 项目的 buildspec.yml 或 CodeDeploy 部署的 appspec.yml。
  - 您可以在源代码中包含可选项，如 README 或非工具链 AWS 资源的 template.yml。

- 工具链模板。您的工具链模板预配置要为您的项目管理的 AWS 资源和 IAM 角色。
- 源位置。如果您为项目指定源代码和工具链模板，则必须提供位置。将您的源文件和工具链模板上传到 Amazon S3 存储桶。AWS CodeStar 将检索文件并使用它们创建项目。

> **Important**
>
> 确保在 AWS CLI 中配置首选的 AWS 区域。在 AWS CLI 中配置的 AWS 区域中创建您的项目。

1. 运行 create-project 命令并包含 `--generate-cli-skeleton` 参数：

```
aws codestar create-project --generate-cli-skeleton
```

输出中将显示 JSON 格式的数据。将数据复制到本地计算机上或安装 AWS CLI 的实例上某位置处的文件（如 *input.json*）中。按照下面所示修改复制的数据，并保存您的结果。

```
{
    "name": "project-name",
    "id": "project-id",
    "description": "description",
    "sourceCode": [
        {
            "source": {
                "s3": {
                    "bucketName": "s3-bucket-name",
                    "bucketKey": "s3-bucket-object-key"
                }
            },
            "destination": {
                "codeCommit": {
                    "name": "codecommit-repository-name"
                },
                "gitHub": {
                    "name": "github-repository-name",
                    "description": "github-repository-description",
                    "type": "github-repository-type",
                    "owner": "github-repository-owner",
                    "privateRepository": true,
                    "issuesEnabled": true,
```

```
                    "token": "github-personal-access-token"
                }
            }
        }
    ],
    "toolchain": {
        "source": {
            "s3": {
                "bucketName": "s3-bucket-name",
                "bucketKey": "s3-bucket-object-key"
            }
        },
        "roleArn": "service-role-arn",
        "stackParameters": {
            "KeyName": "key-name"
        }
    },
    "tags": {
        "KeyName": "key-name"
    }
}
```

替换以下内容：

- *project-name*：必需。此 AWS CodeStar 项目的友好名称。
- *project-id*：必需。此 AWS CodeStar 项目的项目 ID。

> **Note**
>
> 在创建项目时，您必须具有唯一的项目 ID。如果您提交的输入文件中的项目 ID 已存在，则您会收到错误。

- *description*：可选。此 AWS CodeStar 项目的描述。
- *sourceCode*：可选。为项目提供的源代码的配置信息。目前，仅支持单个 `sourceCode` 对象。每个 `sourceCode` 对象包含有关 AWS CodeStar 检索源代码的位置和源代码的填充目标的信息。
  - *source*：必需。这定义您将源代码上传到的位置。唯一受支持的源是 Amazon S3。在创建您的项目后，AWS CodeStar 检索源代码并将其包含在存储库中。
    - *S3*：可选。您的源代码的 Amazon S3 位置。
      - *bucket-name*：包含您的源代码的存储桶。

- *bucket-key*：指向包含您的源代码的 .zip 文件（例如，`src.zip`）的存储桶前缀和对象键。

- *destination*：可选。在创建项目时，将您的源代码填充到的目标位置。您的源代码的受支持目标是 CodeCommit 和 GitHub。

  您只能提供下面两个选项之一：

  - *codeCommit*：唯一的必需属性是应包含您的源代码的 CodeCommit 存储库的名称。此存储库应在您的工具链模板中。

    > **Note**
    >
    > 对于 CodeCommit，您必须提供在工具链堆栈中定义的存储库的名称。AWS CodeStar 将使用您在 Amazon S3 中提供的源代码初始化此存储库。

  - *gitHub*：此对象表示创建 GitHub 存储库并使用源代码为其做种所需的信息。如果您选择 GitHub 存储库，则需要以下值。

    > **Note**
    >
    > 对于 GitHub，您无法指定现有的 GitHub 存储库。AWS CodeStar 会为您创建一个存储库并使用您上传到 Amazon S3 的源代码填充它。AWS CodeStar 使用以下信息在 GitHub 中创建您的存储库。

    - *name*：必需。您的 GitHub 存储库的名称。
    - *description*：必需。您的 GitHub 存储库的描述。
    - *type*：必需。GitHub 存储库的类型。有效值为 User 或 Organization。
    - *owner*：必需。您的存储库所有者的 GitHub 用户名称。如果存储库应由 GitHub 组织所有，请提供组织名称。
    - *privateRepository*：必需。您希望此存储库是私有的还是公有的。有效值为 true 或 false。
    - *issuesEnabled*：必需。您是否希望在 GitHub 中对此存储库启用问题。有效值为 true 或 false。
    - *token*：可选。这是 AWS CodeStar 用于访问您的 GitHub 账户的个人访问令牌。此令牌必须包含以下范围：存储库、用户和 admin:repo_hook。要从 GitHub 检索个人访问令牌，请参阅 GitHub 网站上的为命令行创建个人访问令牌。

> **Note**
>
> 如果您使用 CLI 创建包含 GitHub 源存储库的项目，则 AWS CodeStar 使用您的令牌通过 OAuth 应用程序访问存储库。如果您使用控制台创建包含 GitHub 源存储库的项目，则 AWS CodeStar 使用连接资源，该资源通过 GitHub 应用程序访问存储库。

- *toolchain*：有关要在创建项目时设置的 CI/CD 工具链的信息。这包括您上传工具链模板的位置。模板将创建 AWS CloudFormation 堆栈，其中包含您的工具链资源。这还包括供 AWS CloudFormation 引用的任何参数覆盖和用于创建堆栈的角色。AWS CodeStar 检索模板并使用 AWS CloudFormation 运行模板。
  - *source*：必需。您工具链模板的位置。Amazon S3 是唯一受支持的源位置。
    - *S3*：可选。您将工具链模板上传到的 Amazon S3 位置。
      - *bucket-name*：Amazon S3 存储桶的名称。
      - *bucket-key*：指向包含您的工具链模板的 .yml 或 .json 文件（例如，`files/toolchain.yml`）的存储桶前缀和对象键。
  - *stackParameters*：可选。包含传递到 AWS CloudFormation 的键-值对。这些是您的工具链模板设置为引用的参数（如果有）。
  - *role*：可选。用于在您的账户中创建工具链资源的角色。角色是必填项，如下所示：
    - 如果未提供角色，则 AWS CodeStar 使用工具链为 AWS CodeStar 快速启动模板时为您的账户创建的默认服务角色。如果您的账户中不存在服务角色，您可以创建一个。有关信息，请参阅 步骤 2：创建 AWS CodeStar 服务角色。
    - 如果要上传并使用自己的自定义工具链模板，则必须提供角色。您可以根据 AWS CodeStar 服务角色和策略语句创建一个角色。有关此策略声明的示例，请参阅 AWSCodeStarServiceRole 策略。
- *tags*：可选。附加到您的 AWS CodeStar 项目的标签。

> **Note**
>
> 这些标签不会附加到项目中包含的资源。

2. 切换到包含您刚才保存的文件的目录，然后再次运行 create-project 命令。包含 `--cli-input-json` 参数。

```
aws codestar create-project --cli-input-json file://input.json
```

3. 如果成功，输出中将显示与以下内容类似的数据：

```
{
    "id": "project-ID",
    "arn": "arn"
}
```

- 输出包含有关新项目的信息：
  - id 值表示项目 ID。
  - arn 值表示项目的 ARN。

4. 使用 describe-project 命令检查您的项目创建状态。包含 --id 参数。

```
aws codestar describe-project --id <project_ID>
```

与以下内容类似的数据将显示在输出中：

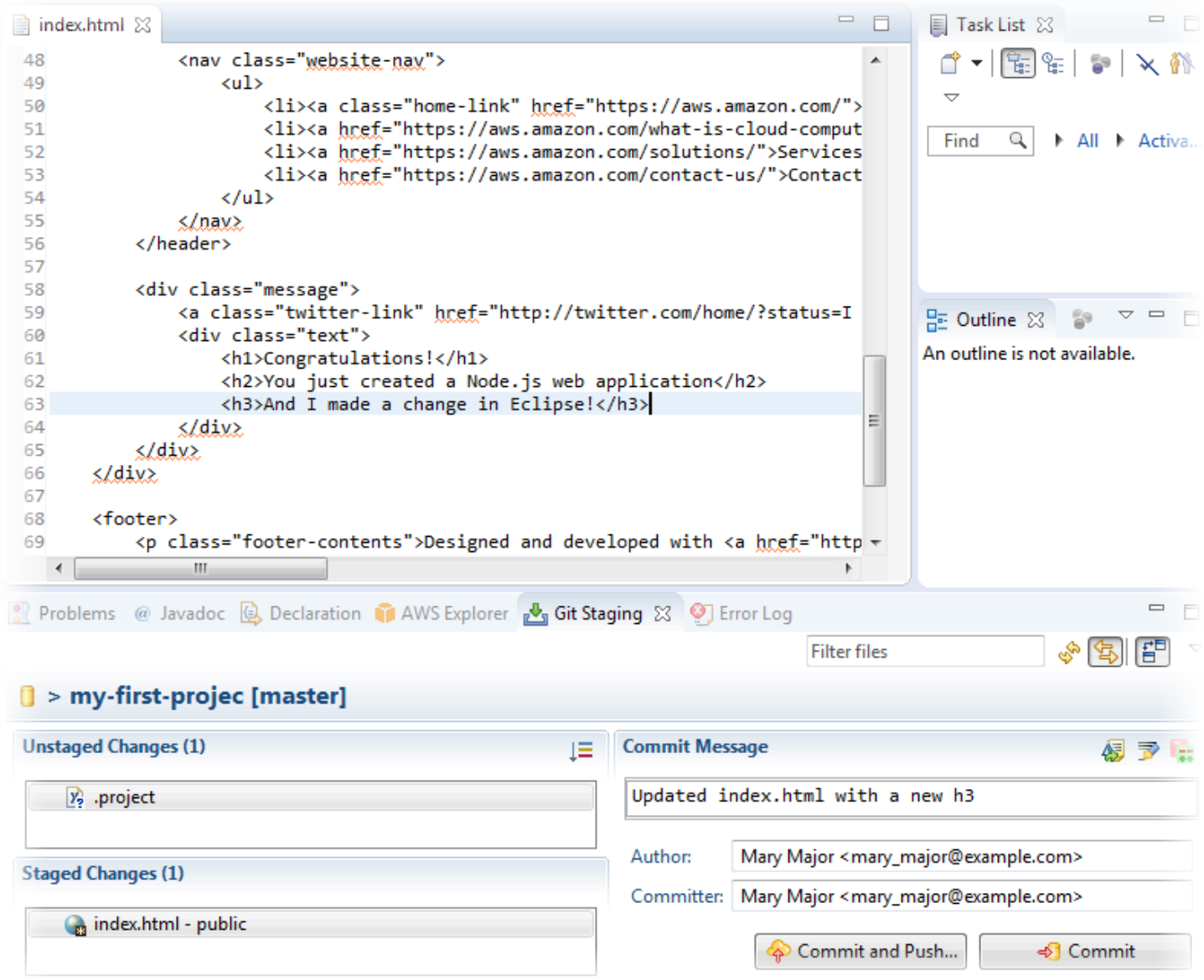
```
{
    "name": "MyProject",
    "id": "myproject",
    "arn": "arn:aws:codestar:us-east-1:account_ID:project/myproject",
    "description": "",
    "createdTimeStamp": 1539700079.472,
    "stackId": "arn:aws:cloudformation:us-east-1:account_ID:stack/awscodestar-
myproject/stack-ID",
    "status": {
        "state": "CreateInProgress"
    }
}
```

- 输出包含有关新项目的信息：
  - state 值表示项目创建状态，如 CreateInProgress 或 CreateComplete。

在创建项目时，您可以从命令行或您常用的 IDE 为项目存储库添加成员或配置访问权限。

将 AWS CodeStar 与 IDE 配合使用

在将 IDE 与 AWS CodeStar 集成时，您可以继续在您的首选环境中编写和开发代码。每次提交和推送代码时，您所做的更改都将包含在 AWS CodeStar 项目中。



主题

- 将 AWS Cloud9 与 AWS CodeStar 结合使用
- 将 AWS CodeStar 与 Eclipse 配合使用
- 将 Visual Studio 与 AWS CodeStar 一起使用

将 AWS Cloud9 与 AWS CodeStar 结合使用

在 AWS CodeStar 项目中，您可以使用 AWS Cloud9 来更改代码和开发软件。AWS Cloud9 是可通过 Web 浏览器访问的在线 IDE。该 IDE 提供丰富的代码编辑体验，对多种编程语言和运行时调试程序的支持以及内置终端。在后台，Amazon EC2 实例将托管 AWS Cloud9 开发环境。此环境提供 AWS Cloud9 IDE 以及对 AWS CodeStar 项目代码文件的访问。有关更多信息，请参阅 AWS Cloud9 用户指南。

您可以使用 AWS CodeStar 控制台或 AWS Cloud9 控制台为在 CodeCommit 中存储代码的项目创建 AWS Cloud9 开发环境。对于在 GitHub 中存储其代码的 AWS CodeStar 项目，您只能使用 AWS Cloud9 控制台。本主题介绍了这两个控制台的使用方法。

要使用 AWS Cloud9，您需要：

- 一名已作为团队成员添加到 AWS CodeStar 项目的 IAM 用户。
- 如果 AWS CodeStar 项目将其源代码存储在 CodeCommit 中，则需要该 IAM 用户的 AWS 凭证。

主题

为项目创建 AWS Cloud9 环境

按照以下步骤为 AWS CodeStar 项目创建 AWS Cloud9 开发环境。

1. 如果要创建新项目，请按照 创建项目 中的步骤进行操作。
2. 在 AWS CodeStar 控制台中打开项目。在导航栏上，选择 IDE。选择创建环境，然后执行以下步骤。

> **Important**
>
> 如果项目位于不支持 AWS Cloud9 的 AWS 区域中，侧导航栏上的 IDE 选项卡将不显示 AWS Cloud9 选项。但是，您可以使用 AWS Cloud9 控制台创建一个开发环境，打开这一

新环境，然后将其连接到项目的 AWS CodeCommit 存储库。跳过以下步骤，并参阅AWS Cloud9 用户指南中的创建环境、打开环境和 AWS CodeCommit 示例。有关受支持的 AWS 区域的列表，请参阅 Amazon Web Services 一般参考 中的 AWS Cloud9。

在创建 AWS Cloud9 环境中，自定义项目默认值。

1. 要更改用于托管环境的 Amazon EC2 实例的默认类型，在实例类型中选择实例的类型。
2. AWS Cloud9 使用您 AWS 账户中的 Amazon Virtual Private Cloud (Amazon VPC) 与实例进行通信。根据您 AWS 账户中 Amazon VPC 的设置方式，请执行下列操作之一。

该账户是否具有 VPC，且该 VPC 包含至少一个子网？	您希望 AWS Cloud9 使用的 VPC 是不是账户中的默认 VPC？	该 VPC 是否有单个子网？	请执行该操作
否	—	—	如果不存在 VPC，则创建一个。展开网络设置。对于网络 (VPC) 选择创建 VPC，然后按页面上的说明操作。有关更多信息，请参阅 AWS Cloud9 用户指南中的为 AWS Cloud9 创建 Amazon VPC。 如果存在 VPC 但没有子网，请创建一个子网。展开网络设置。为网络 (VPC) 选择创建子网，然后按照说明操作。有关更多信息，请参阅 AWS Cloud9 用户指南中的为 AWS Cloud9 创建子网。
是	是	是	跳至本过程的步骤 4。（AWS Cloud9 使用默认 VPC 及其单个子网。）
是	是	否	对于子网，选择希望 AWS Cloud9 在预先选择的默认 VPC 中使用的子网。

该账户是否具有 VPC，且该 VPC 包含至少一个子网？	您希望 AWS Cloud9 使用的 VPC 是不是账户中的默认 VPC？	该 VPC 是否有单个子网？	请执行该操作
是	否	是或否	对于网络 (VPC)，选择希望 AWS Cloud9 使用的 VPC。对于子网，选择您希望 AWS Cloud9 在该 VPC 中使用的子网。

有关更多信息，请参阅 AWS Cloud9 用户指南中的 AWS Cloud9 开发环境的 Amazon VPC 设置。

3. 输入环境名称，并可选择添加环境描述。

> **Note**
>
> 每个用户的环境名称必须是唯一的。

4. 要更改 AWS Cloud9 在环境未被使用时将环境关闭的默认时间段，请展开成本节省设置，然后更改该设置。
5. 选择创建环境。

要打开环境，请参阅为项目打开 AWS Cloud9 环境。

您可以使用上述步骤来为项目创建多个环境。例如，您可能希望使用一个环境来处理一部分代码，使用另一个环境通过不同设置来处理同一部分代码。

为项目打开 AWS Cloud9 环境

按照以下步骤打开您为 AWS CodeStar 项目创建的 AWS Cloud9 开发环境。

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在导航栏上选择 IDE。

> **Important**
>
> 如果项目的源代码存储在 GitHub 中，您在导航栏上将看不到 IDE。但是，您可以使用 AWS Cloud9 控制台来打开现有环境。跳过此过程的其余部分，参见 AWS Cloud9用户指南中的打开环境和 将 GitHub 与 AWS Cloud9 结合使用。

2. 对于您的 AWS Cloud9 环境或共享 AWS Cloud9 环境，请为要打开的环境选择 Open IDE。

您可以使用 AWS Cloud9 IDE 立即开始处理项目的 AWS CodeCommit 存储库中的代码。有关更多信息，请参阅 AWS Cloud9用户指南中的环境窗口、编辑器、选项卡和窗格以及终端，还有 AWS CodeCommit 用户指南中的基本 Git 命令。

与项目团队成员共享 AWS Cloud9 环境

为 AWS CodeStar 项目创建 AWS Cloud9 开发环境后，您可以邀请包括项目团队成员在内的其他用户跨 AWS 账户访问该环境。这对配对编程特别有用，其中两个程序员轮流编写代码，并通过屏幕共享或位于相同工作站中来对相同代码提出建议。环境成员可以使用共享的 AWS Cloud9 IDE 在代码编辑器中查看突出显示的、每个成员所做的代码更改，还可在编码的同时与其他成员进行文字聊天。

将团队成员添加到项目不会自动允许该成员参与该项目的任何相关 AWS Cloud9 开发环境。要邀请项目团队成员访问项目的环境，您需要确定正确的环境成员访问角色、对该用户应用 AWS 托管策略，并邀请该用户访问您的环境。有关更多信息，请参阅 AWS Cloud9 用户指南中的关于环境成员访问角色和邀请 IAM 用户加入您的环境。

当您邀请项目团队成员访问项目的环境时，AWS CodeStar 控制台将向该团队成员显示环境。该环境显示在项目 AWS CodeStar 控制台中 IDE 选项卡上的共享环境列表中。要显示此列表，可让团队成员在控制台中打开该项目，然后在导航栏中选择 IDE。

> **Important**
>
> 如果项目的源代码存储在 GitHub 中，您在导航栏上将看不到 IDE。但是，您可以使用 AWS Cloud9 控制台来邀请包括项目团队成员在内的其他用户跨 AWS 账户访问环境。为此，请参阅本指南中的 将 GitHub 与 AWS Cloud9 结合使用，并参阅 AWS Cloud9 用户指南中的关于环境成员访问角色和邀请 IAM 用户加入您的环境。

您还可以邀请不是项目团队成员的用户访问环境。例如，您可能希望用户能处理项目的代码但没有该项目的其他访问权限。要邀请此类型用户，请参阅 AWS Cloud9 用户指南中的关于环境成员访问角

色和邀请 IAM 用户加入您的环境。当您邀请不是项目团队成员的用户访问项目的环境时，该用户可以使用 AWS Cloud9 控制台来访问该环境。有关更多信息，请参阅 AWS Cloud9 用户指南中的打开环境。

从项目中删除 AWS Cloud9 环境

当您从 AWS CodeStar 中删除某个项目及其所有 AWS 资源时，同时将删除使用 AWS CodeStar 控制台创建的所有相关的 AWS Cloud9 开发环境，并且无法恢复。您可以删除项目中的开发环境，而不删除项目。

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在导航栏中选择 IDE。

   > **Important**
   >
   > 如果项目的源代码存储在 GitHub 中，您在导航栏上将看不到 IDE。但是，您可以使用 AWS Cloud9 控制台来删除开发环境。跳过此过程的其余部分，参见 AWS Cloud9 用户指南中的删除环境。

2. 在 Cloud9 环境中选择要删除的环境，然后选择删除
3. 输入 **`delete`** 以确认删除开发环境，然后选择删除。

   > **Warning**
   >
   > 开发环境在删除后不能恢复。环境中所有未提交的代码更改都将丢失。

将 GitHub 与 AWS Cloud9 结合使用

对于将源代码存储在 GitHub 中的 AWS CodeStar 项目，AWS CodeStar 控制台不支持直接使用 AWS Cloud9 开发环境。但是，您可以使用 AWS Cloud9 控制台来处理 GitHub 存储库中的源代码。

1. 使用 AWS Cloud9 控制台创建 AWS Cloud9 开发环境。有关信息，请参阅 AWS Cloud9 用户指南中的创建环境。
2. 使用 AWS Cloud9 控制台打开开发环境。有关信息，请参阅 AWS Cloud9 用户指南中的打开环境。
3. 在 IDE 中，使用终端会话连接到 GitHub 存储库（名为克隆 的过程）。如果终端会话未运行，请在 IDE 中的菜单栏上选择窗口、新建终端。有关用于克隆 GitHub 存储库的命令，请参阅 GitHub 帮助网站上的克隆存储库。

要导航到 GitHub 存储库的主页，请在项目已在 AWS CodeStar 控制台中打开的情况下，在侧导航栏上选择代码。

4. 使用 IDE 中的环境 窗口和编辑器选项卡查看、更改和保存代码。有关更多信息，请参阅 AWS Cloud9 用户指南中的环境窗口和编辑器、选项卡和窗格。
5. 在 IDE 的终端会话中使用 Git 将代码更改推送到存储库，同时定期从存储库拉取其他人对代码的更改。有关更多信息，请参阅 GitHub 帮助网站上的推送到远程存储库和获取远程存储库。有关 Git 命令，请参阅 GitHub 帮助网站上的 Git 备忘单。

> **Note**
>
> 要阻止 Git 在您每次向存储库推送或从存储库拉取代码时提示您输入 GitHub 登录凭证，您可以使用凭证助手。有关更多信息，请参阅 GitHub 帮助网站上的在 Git 中缓存 GitHub 密码。

其他资源

有关使用 AWS Cloud9 的更多信息，请参阅 AWS Cloud9 用户指南 中的以下内容。

- 教程
- 处理环境
- 使用 IDE
- 示例

将 AWS CodeStar 与 Eclipse 配合使用

在 AWS CodeStar 项目中，您可以使用 Eclipse 来更改代码和开发软件。您可以使用 Eclipse 编辑您的 AWS CodeStar 项目代码，然后将您的更改提交并推送到 AWS CodeStar 项目的源存储库。

> **Note**
>
> 本主题中的信息仅适用于在 CodeCommit 中存储源代码的 AWS CodeStar 项目。如果 AWS CodeStar 项目的源代码存储在 GitHub 中，则可以使用 EGit for Eclipse 等工具。有关更多信息，请参阅 EGit 网站上的 EGit 文档。

如果 AWS CodeStar 项目的源代码存储在 CodeCommit 中，您必须安装支持 AWS CodeStar 的 AWS Toolkit for Eclipse 版本。您还必须是具有所有者或贡献者角色的 AWS CodeStar 项目团队的成员。

要使用 Eclipse，您还需要：

- 一名已作为团队成员添加到 AWS CodeStar 项目的 IAM 用户。
- 如果 AWS CodeStar 项目将其源代码存储在 CodeCommit 中，则需要该 IAM 用户的 Git 凭证（登录凭证）。
- 在您的本地计算机上安装 Eclipse 和 AWS Toolkit for Eclipse 的足够权限。

主题

步骤 1：安装 AWS Toolkit for Eclipse

Toolkit for Eclipse 是一个可添加到 Eclipse 的软件包。它的安装和托管方式与 Eclipse 中的其他软件包相同。AWS CodeStar 工具包作为 Toolkit for Eclipse 的一部分提供。

使用 AWS CodeStar 模块安装 Toolkit for Eclipse

1. 在本地计算机上安装 Eclipse。Eclipse 的受支持版本包括 Luna、Mars 和 Neon。
2. 下载并安装 Toolkit for Eclipse。有关更多信息，请参阅 AWS Toolkit for Eclipse 入门指南。
3. 在 Eclipse 中，选择 Help，然后选择 Install New Software。
4. 在 Available Software 中，选择 Add。
5. 在 Add Repository 中，选择 Archive，浏览到您保存 .zip 文件的位置，然后打开文件。将名称留空，然后选择确定。
6. 在可用软件中，选择全选以选择 AWS 核心管理工具和开发人员工具，然后选择下一步。
7. 在安装详细信息中，选择下一步。
8. 在 Review Licenses 中，查看许可证协议。选择我接受许可协议的条款，然后选择完成。重新启动 Eclipse。

步骤 2：将您的 AWS CodeStar 项目添加到 Eclipse

安装 Toolkit for Eclipse 后，您可以导入 AWS CodeStar 项目并从 IDE 中编辑、提交和推送代码。

> **Note**
>
> 您可以将多个 AWS CodeStar 项目添加到 Eclipse 中的一个工作区，但如果您这样做，当您从一个项目更改到另一个项目时，就必须更新您的项目凭证。

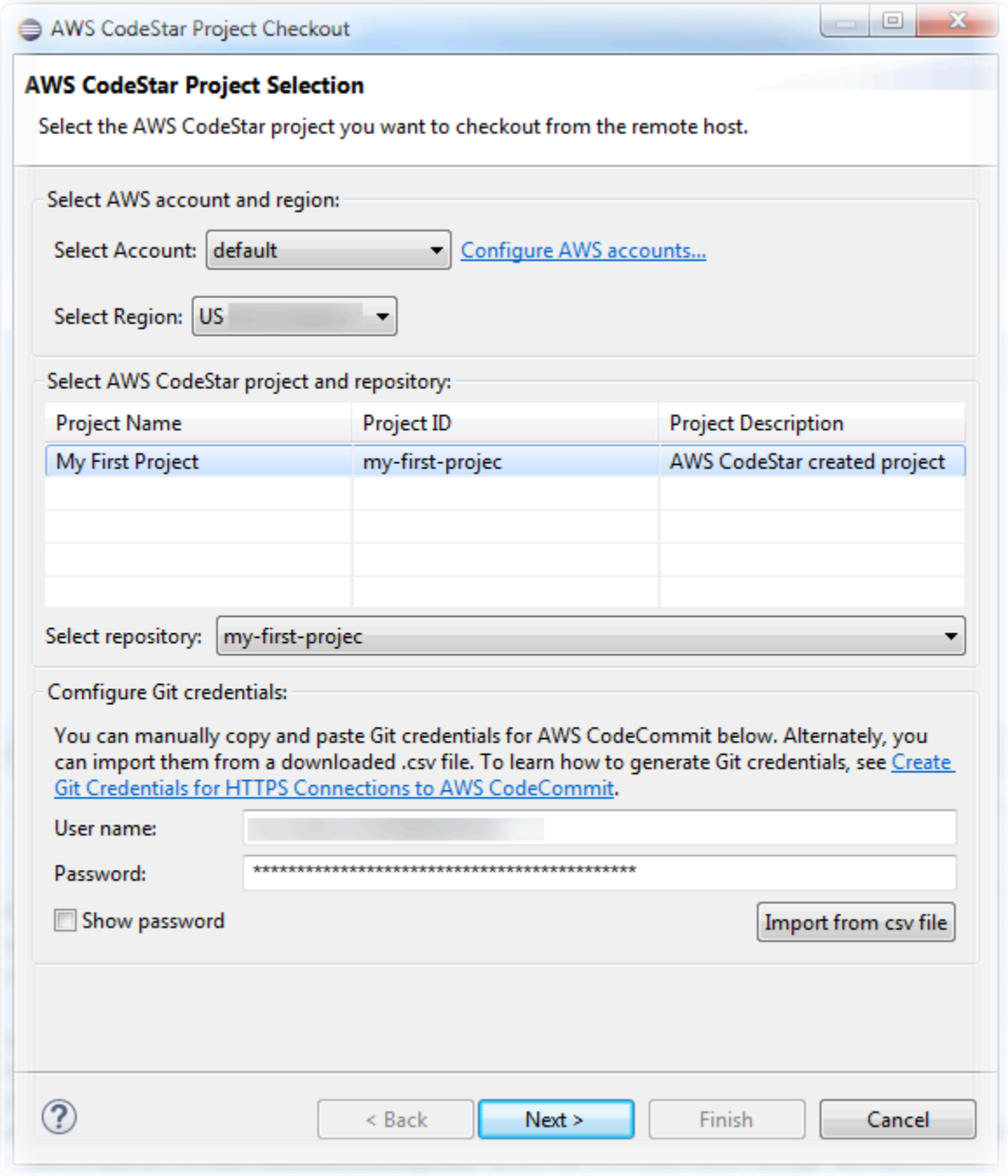
导入 AWS CodeStar 项目

1. 从 AWS 菜单，选择导入 AWS CodeStar 项目。或者，选择 File，然后选择 Import。在选择中，展开 AWS，然后选择 AWS CodeStar 项目。

   选择下一步。

2. 在 AWS CodeStar 项目选择中，选择您的 AWS 配置文件和托管 AWS CodeStar 项目的 AWS 区域。如果您的计算机上没有配置了访问密钥和私有密钥的 AWS 配置文件，请选择配置 AWS 账户并按照说明操作。

   在选择 AWS CodeStar 项目和存储库中，选择您的 AWS CodeStar 项目。在 Git 凭证中，输入您为访问项目的存储库而生成的登录凭证。（如果您没有 Git 凭证，请参阅开始使用。）选择下一步。



3. 默认情况下，项目的存储库的所有分支均处于选中状态。如果您不想导入一个或多个分支，请清除这些框，然后选择 Next。
4. 在本地目标中，选择导入向导将在您的计算机上创建本地存储库的目标位置，然后选择完成。
5. 在项目资源管理器中，展开项目树以浏览 AWS CodeStar 项目中的文件。

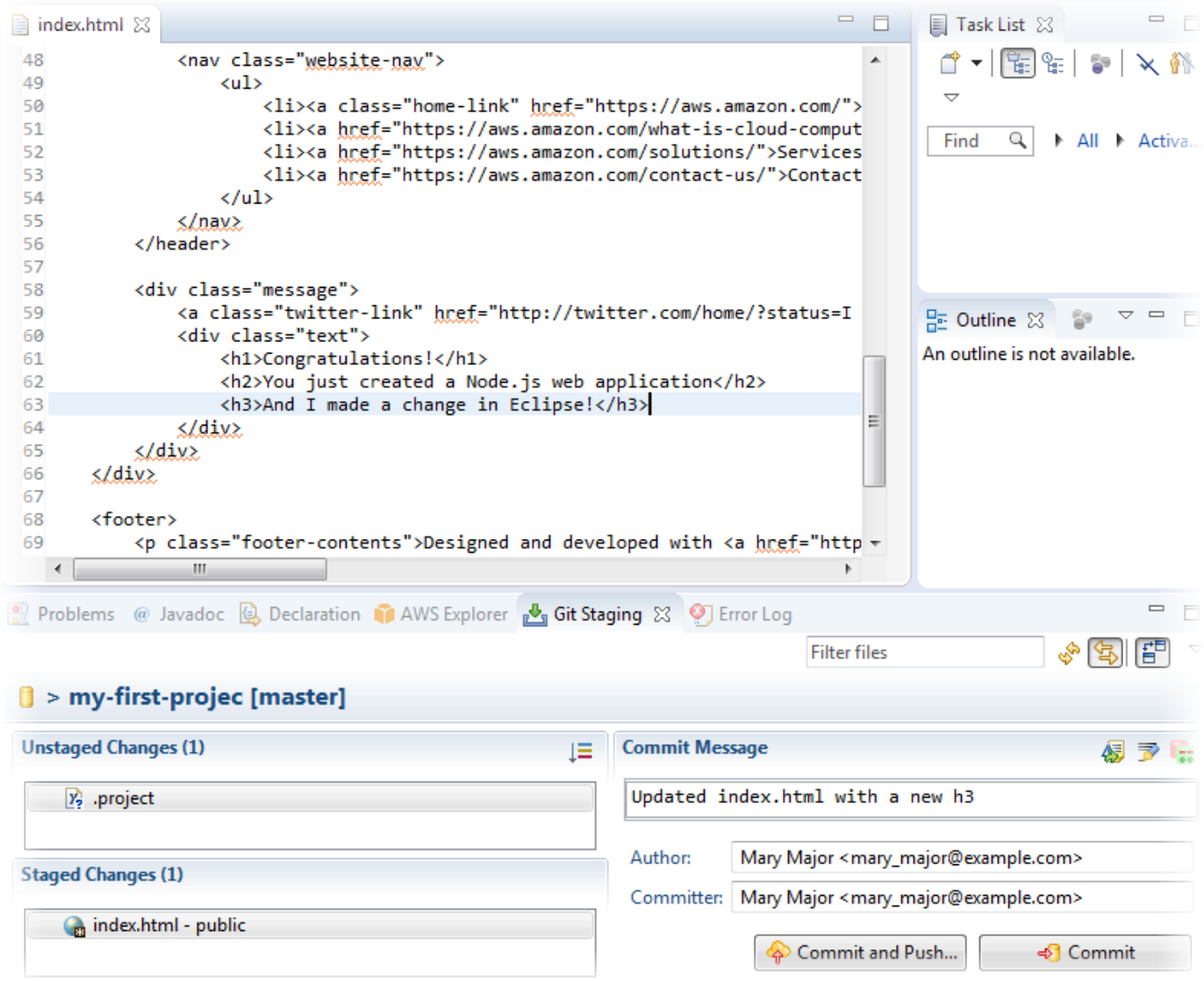
步骤 3：在 Eclipse 中编辑 AWS CodeStar 项目代码

将 AWS CodeStar 项目导入到 Eclipse 工作区中后，您可以为该项目编辑代码，保存您的更改，然后将您的代码提交并推送到该项目的源存储库。此过程与您对使用适用于 Eclipse 的 EGit 插件的任何 Git 存储库采用的过程相同。有关更多信息，请参阅 Eclipse 网站上的 EGit 用户指南。

编辑项目代码并对 AWS CodeStar 项目的源存储库进行第一次提交

1. 在项目资源管理器中，展开项目树以浏览 AWS CodeStar 项目中的文件。
2. 编辑一个或多个代码文件并保存更改。
3. 准备好提交更改后，打开该文件的上下文菜单，选择 Team，然后选择 Commit。

   如果您已在项目视图中打开 Git 暂存窗口，则可以跳过此步骤。
4. 在 Git 暂存中，通过将更改的文件移动到暂存的更改中来暂存更改。在提交消息中输入提交消息，然后选择提交并推送。



要查看代码更改的部署，请返回项目的控制面板。有关更多信息，请参阅步骤 3：查看您的项目。

将 Visual Studio 与 AWS CodeStar 一起使用

在 AWS CodeStar 项目中，您可以使用 Visual Studio 来更改代码和开发软件。

> **Note**
>
> 适用于 Mac 的 Visual Studio 不支持该 AWS 工具包，因此无法将其与 AWS CodeStar 一起使用。
>
> 本主题中的信息仅适用于在 CodeCommit 中存储源代码的 AWS CodeStar 项目。如果 AWS CodeStar 项目的源代码存储在 GitHub 中，则可以使用 GitHub Extension for Visual Studio 等

工具。有关更多信息，请参阅 GitHub Extension for Visual Studio 网站上的 Overview 页面及 GitHub 网站上的 Getting Started with GitHub for Visual Studio。

要在 AWS CodeStar 项目的源存储库中使用 Visual Studio 来编辑代码，您必须安装支持 AWS CodeStar 的 AWS Toolkit for Visual Studio 版本。您必须是具有所有者或贡献者角色的 AWS CodeStar 项目团队的成员。

要使用 Visual Studio，您还需要：

- 一名已作为团队成员添加到 AWS CodeStar 项目的 IAM 用户。
- IAM 用户的 AWS 凭证（例如，您的访问密钥和私有密钥）。
- 在您的本地计算机上安装 Visual Studio 和 AWS Toolkit for Visual Studio 的足够权限。

Toolkit for Visual Studio 是可以添加到 Visual Studio 中的一个软件包。它的安装和托管方式与 Visual Studio 中的其他软件包相同。

安装包含 AWS CodeStar 模块的 Toolkit for Visual Studio 并配置对项目存储库的访问权限

1. 在您的本地计算机上安装 Visual Studio。
2. 下载并安装 Toolkit for Visual Studio 并将 .zip 文件保存到本地文件夹或目录。在 AWS Toolkit for Visual Studio 入门页面上，输入或导入您的 AWS 凭证，然后选择保存并关闭。
3. 在 Visual Studio 中打开 Team Explorer。在托管服务提供商 中，找到 CodeCommit，然后选择连接。
4. 在 Manage Connections 中，选择 Clone。选择您的项目的存储库以及您要将存储库克隆到的本地计算机文件夹，然后选择确定。
5. 如果系统提示您创建 Git 凭证，请选择 Yes。该工具包将尝试代表您创建凭证。将凭证文件保存到安全的位置。这是您保存这些凭证的唯一机会。如果工具包无法代表您创建凭证，或者您选择了否，那么您必须创建并提供您自己的 Git 凭证。有关更多信息，请参阅设置您的计算机以提交更改（IAM 用户）或按照在线指示操作。

   完成克隆项目后，您就可以开始在 Visual Studio 中编辑代码并将更改提交和推送到 CodeCommit 中的项目存储库。

更改 AWS CodeStar 项目中的 AWS 资源

在 AWS CodeStar 中创建项目后，您可以更改 AWS CodeStar 添加到项目的默认 AWS 资源集。

支持的资源更改

下表列出了对 AWS CodeStar 项目中的默认 AWS 资源的受支持的更改。

更改	注意
向 AWS CodePipeline 添加阶段。	请参阅向 AWS CodePipeline 添加阶段。
更改 Elastic Beanstalk 环境设置。	请参阅更改 AWS Elastic Beanstalk 环境设置。
在 Amazon API Gateway 中更改 AWS Lambda 函数的代码或设置、其 IAM 角色或其 API。	请参阅更改源代码中的 AWS Lambda 函数。
将资源添加到 AWS Lambda 项目并扩展权限以创建和访问新资源。	请参阅将资源添加到项目。
为 AWS Lambda 函数添加使用 CodeDeploy 转移流量的功能。	请参阅转移 AWS Lambda 项目的流量。
添加 AWS X-Ray 支持	请参阅启用项目跟踪。
编辑您的项目的 buildspec.yml 文件以便为要运行的 AWS CodeBuild 添加单元测试构建阶段。	请参阅无服务器项目教程中的步骤 7：向 Web 服务添加单元测试。
将自己的 IAM 角色添加到您的项目。	请参阅向项目添加 IAM 角色。
更改 IAM 角色定义。	对于应用程序堆栈中定义的角色。您无法更改工具链或 AWS CloudFormation 堆栈中定义的角色。
修改您的 Lambda 项目以添加终端节点。	
修改您的 EC2 项目以添加终端节点。	
修改您的 Elastic Beanstalk 项目以添加终端节点。	

更改	注意
编辑您的项目以添加 Prod 阶段和终端节点。	请参阅将 Prod 阶段和终端节点添加到项目。
在 AWS CodeStar 项目中安全地使用 SSM 参数。	请参阅the section called “在 AWS CodeStar 项目中安全地使用 SSM 参数。”。

不支持以下更改。

- 切换到不同的部署目标（例如，部署到 AWS Elastic Beanstalk 而不是 AWS CodeDeploy）。
- 添加友好的 Web 终端节点名称。
- 更改 CodeCommit 存储库名称（对于连接到 CodeCommit 的 AWS CodeStar 项目）。
- 对于连接到 GitHub 的 AWS CodeStar 项目，断开 GitHub 存储库连接，然后将存储库重新连接到此项目，或将任何其他存储库连接到此项目。在管道的源阶段，可以使用 CodePipeline 控制台（而不是 AWS CodeStar 控制台）断开并重新连接 GitHub。但是，如果将源阶段重新连接到其他 GitHub 存储库，则在此项目的 AWS CodeStar 控制面板中，存储库和问题磁贴中的信息可能是错误的或过时的。断开 GitHub 存储库连接不会从 AWS CodeStar 项目控制面板中的提交历史记录和 GitHub 问题磁贴中删除此存储库的信息。要删除此信息，请使用 GitHub 网站禁用从 AWS CodeStar 项目对 GitHub 的访问。要撤销访问权限，请在 GitHub 网站上使用您的 GitHub 账户资料信息设置页面中的授权的 OAuth 应用程序部分。
- 断开 CodeCommit 存储库连接（对于连接到 CodeCommit 的 AWS CodeStar 项目），然后将存储库重新连接到此项目，或将任何其他存储库连接到此项目。

向 AWS CodePipeline 添加阶段

您可以向 AWS CodeStar 在项目中创建的管道添加新阶段。有关更多信息，请参阅 AWS CodePipeline 用户指南中的在 AWS CodePipeline 中编辑管道。

> **Note**
>
> 如果新阶段依赖 AWS CodeStar 未创建的任何 AWS 资源，则管道可能会中断。这是因为，默认情况下，AWS CodeStar 为 AWS CodePipeline 创建的 IAM 角色可能无权访问这些资源。要尝试向 AWS CodePipeline 提供对 AWS CodeStar 未创建的 AWS 资源的访问权限，您可能需要更改 AWS CodeStar 创建的 IAM 角色。这不受支持，因为 AWS CodeStar 可能会在对项目执行定期更新检查时删除您对 IAM 角色的更改。

更改 AWS Elastic Beanstalk 环境设置

您可以更改 AWS CodeStar 在项目中创建的 Elastic Beanstalk 环境的设置。例如，您可能需要将 AWS CodeStar 项目中的默认 Elastic Beanstalk 环境从“单一实例”更改为“负载均衡”。为此，请在项目的存储库中编辑 `template.yml` 文件。您可能还需要更改项目工作线程角色的权限。在您推送模板更改后，AWS CodeStar 和 AWS CloudFormation 会为您预置资源。

有关编辑 `template.yml` 文件的更多信息，请参阅 使用 Template.yml 文件更改应用程序资源。有关 Elastic Beanstalk 环境的更多信息，请参阅 AWS Elastic Beanstalk 开发人员指南中的 AWS Elastic Beanstalk 环境管理控制台。

更改源代码中的 AWS Lambda 函数

您可以更改 Lambda 函数的代码或设置，或者 AWS CodeStar 在项目中创建的其 IAM 角色或 API Gateway API。为此，建议您使用 AWS 无服务器应用程序模型 (AWS SAM) 以及项目的 CodeCommit 存储库中的 `template.yaml` 文件。此 `template.yaml` 文件定义 API Gateway 中的函数名称、处理程序、运行时、IAM 角色和 API。有关更多信息，请参阅 GitHub 网站上的如何使用 AWS SAM 创建无服务器应用程序。

启用项目跟踪

AWS X-Ray 提供了跟踪，可用于分析分布式应用程序的性能行为（例如，响应时间延迟）。向 AWS CodeStar 项目添加跟踪后，您可以使用 AWS X-Ray 控制台查看应用程序视图和响应时间。

> **Note**
>
> 您可以将下述步骤用于在带有以下项目支持更改的情况下创建的以下项目：
>
> - 任何 Lambda 项目。
> - 对于在 2018 年 8 月 3 日之后创建的 Amazon EC2 或 Elastic Beanstalk 项目，AWS CodeStar 在项目存储库中预配置了 `/template.yml` 文件。

每个 AWS CodeStar 项目模板均包含一个 AWS CloudFormation 文件，该文件用于为您的应用程序的 AWS 运行时依赖项（如数据库表和 Lambda 函数）建模。此文件存储在源存储库中的 `/template.yml` 文件中。

您可以通过将 AWS X-Ray 资源添加到 `Resources` 部分来修改此文件以添加跟踪。然后，为您的项目修改 IAM 权限以允许 AWS CloudFormation 创建资源。有关模板元素和格式的信息，请参阅 AWS 资源类型参考。

以下是自定义您的模板时应遵循的概要步骤。

1. 步骤 1：在 IAM 中编辑工作线程角色以便跟踪
2. 步骤 2：为跟踪修改 template.yml 文件
3. 步骤 3：为跟踪提交并推送您的模板更改
4. 步骤 4：为跟踪监控 AWS CloudFormation 堆栈更新

步骤 1：在 IAM 中编辑工作线程角色以便跟踪

您必须以管理员身份登录才能执行步骤 1 和 4。此步骤显示了编辑 Lambda 项目权限的示例。

> **Note**
>
> 如果您的项目已预配置权限边界策略，则可以跳过此步骤。
> 对于在太平洋夏令时 2018 年 12 月 6 日之后创建的项目，AWS CodeStar 已为您的项目预配置权限边界策略。

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。
2. 使用 `template.yml file` 创建项目或选择现有项目，然后打开项目资源页面。
3. 在项目资源下，在资源列表中找到为 CodeStarWorker/Lambda 角色创建的 IAM 角色。角色名称遵循以下格式：`role/CodeStarWorker-`*`Project_name`*`-lambda-`*`Function_name`*。选择此角色的 ARN。
4. 此角色将在 IAM 控制台中打开。选择附加策略。搜索 `AWSXrayWriteOnlyAccess` 策略，选中它旁边的框，然后选择附加策略。

步骤 2：为跟踪修改 template.yml 文件

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 选择您的无服务器项目，然后打开代码页面。在您的存储库的顶层，找到并编辑 `template.yml` 文件。在 `Resources` 下，将资源粘贴到 `Properties` 部分中。

```
      Tracing: Active
```

以下示例显示已修改的模板：

```
Resources:
  GetHelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.get
      Runtime: nodejs4.3
      Tracing: Active # Enable X-Ray tracing for the function
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
```

步骤 3：为跟踪提交并推送您的模板更改

- 提交并推送 `template.yml` 文件中的更改。

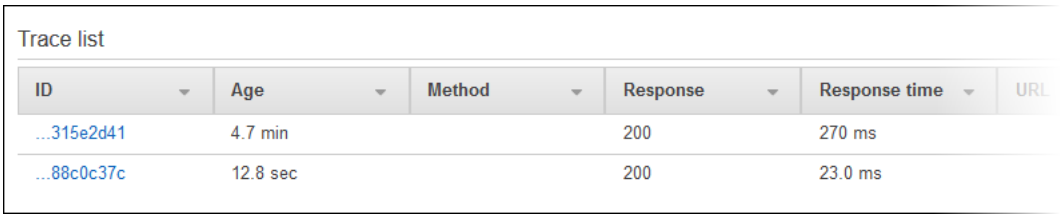
> **Note**
>
> 这会启动您的管道。如果您在更新 IAM 权限之前提交更改，则您的管道会启动，AWS CloudFormation 堆栈更新会遇到错误，并且堆栈更新会回滚。如果发生这种情况，请更正权限，然后重新启动您的管道。

步骤 4：为跟踪监控 AWS CloudFormation 堆栈更新

1. 在您的项目的管道启动部署阶段时，AWS CloudFormation 堆栈更新启动。要查看堆栈更新的状态，请在 AWS CodeStar 控制面板上选择您的管道中的 AWS CloudFormation 阶段。

   如果 AWS CloudFormation 中的堆栈更新返回错误，请参阅 AWS CloudFormation：由于缺少权限，堆栈创建已回滚中的故障排除指南。如果工作线程角色缺少权限，请编辑附加到项目的 Lambda 工作线程角色的策略。请参阅步骤 1：在 IAM 中编辑工作线程角色以便跟踪。
2. 使用控制面板查看您的管道的成功完成。现已在您的应用程序上启用跟踪。
3. 通过在 Lambda 控制台中查看您的函数的详细信息来验证是否已启用跟踪。

4. 为您的项目选择应用程序终端节点。将跟踪与您的应用程序进行的此交互。您可以在 AWS X-Ray 控制台中查看跟踪信息。



将资源添加到项目

所有项目的每个 AWS CodeStar 模板均包含一个 AWS CloudFormation 文件，该文件用于为您的应用程序的 AWS 运行时依赖项（如数据库表和 Lambda 函数）建模。此项存储在源存储库中的 `/template.yml` 文件中。

> **Note**
>
> 您可以将下述步骤用于在带有以下项目支持更改的情况下创建的以下项目：
>
> - 任何 Lambda 项目。
> - 对于在 2018 年 8 月 3 日之后创建的 Amazon EC2 或 Elastic Beanstalk 项目，AWS CodeStar 在项目存储库中预配置了 `/template.yml` 文件。

您可以通过将 AWS CloudFormation 资源添加到 `Resources` 部分来修改此文件。修改 `template.yml` 文件将允许 AWS CodeStar 和 AWS CloudFormation 将新资源添加到您的项目。一些资源需要您将其他权限添加到项目的 CloudFormation 工作线程角色的策略。有关模板元素和格式的信息，请参阅 AWS 资源类型参考。

在您确定必须将哪些资源添加到项目中后，以下是自定义模板时应遵循的概要步骤。有关 AWS CloudFormation 资源及其必需属性的列表，请参阅 AWS 资源类型参考。

1. 步骤 1：在 IAM 中编辑 CloudFormation 工作线程角色（如有必要）
2. 步骤 2：修改 template.yml 文件
3. 步骤 3：提交并推送您的模板更改
4. 步骤 4：监控 AWS CloudFormation 堆栈更新
5. 步骤 5：使用内联策略添加资源权限

按照此部分中的步骤修改您的 AWS CodeStar 项目模板以添加资源，然后在 IAM 中扩展项目的 CloudFormation 工作线程角色的权限。在此示例中，将 AWS::SQS::Queue 资源添加到 `template.yml` 文件中。此更改在 AWS CloudFormation 中启动将 Amazon Simple Queue Service 队列添加到您的项目中的自动响应。

步骤 1：在 IAM 中编辑 CloudFormation 工作线程角色

您必须以管理员身份登录才能执行步骤 1 和 5。

> **Note**
>
> 如果您的项目已预配置权限边界策略，则可以跳过此步骤。
> 对于在太平洋夏令时 2018 年 12 月 6 日之后创建的项目，AWS CodeStar 已为您的项目预配置权限边界策略。

1. 登录 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。
2. 使用 `template.yml file` 创建项目或选择现有项目，然后打开项目资源页面。
3. 在项目资源下，在资源列表中找到为 CodeStarWorker/AWS CloudFormation 角色创建的 IAM 角色。角色名称遵循以下格式：`role/CodeStarWorker-`*`Project_name`*`-CloudFormation`。
4. 此角色将在 IAM 控制台中打开。在权限选项卡上的内联策略中，展开您的服务角色策略所在的行，然后选择编辑策略。
5. 选择 JSON 选项卡以编辑策略。

   > **Note**
   >
   > 附加到工作线程角色的策略为 `CodeStarWorkerCloudFormationRolePolicy`。

6. 在 JSON 字段中，在 `Statement` 元素中添加以下策略语句。

```
{
  "Action": [
    "sqs:CreateQueue",
    "sqs:DeleteQueue",
    "sqs:GetQueueAttributes",
```

```
      "sqs:SetQueueAttributes",
      "sqs:ListQueues",
      "sqs:GetQueueUrl"
    ],
    "Resource": [
      "*"
    ],
    "Effect": "Allow"
}
```

7. 选择查看策略以确保策略不包含任何错误，然后选择保存更改。

步骤 2：修改 template.yml 文件

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 选择您的无服务器项目，然后打开代码页面。在您的存储库的顶层中，记下 `template.yml` 的位置。
3. 在本地存储库中使用 IDE、控制台或命令行编辑存储库中的 `template.yml` 文件。将资源粘贴到 `Resources` 部分中。在此示例中，在复制以下文本时，会添加 `Resources` 部分。

```
Resources:
  TestQueue:
    Type: AWS::SQS::Queue
```

以下示例显示已修改的模板：

步骤 3：提交并推送您的模板更改

- 提交并推送您在步骤 2 中保存的 `template.yml` 文件中的更改。

> **Note**
>
> 这会启动您的管道。如果您在更新 IAM 权限之前提交更改，则您的管道会启动，并且 AWS CloudFormation 堆栈更新会遇到错误，这会导致堆栈更新回滚。如果发生这种情况，请更正权限，然后重新启动您的管道。

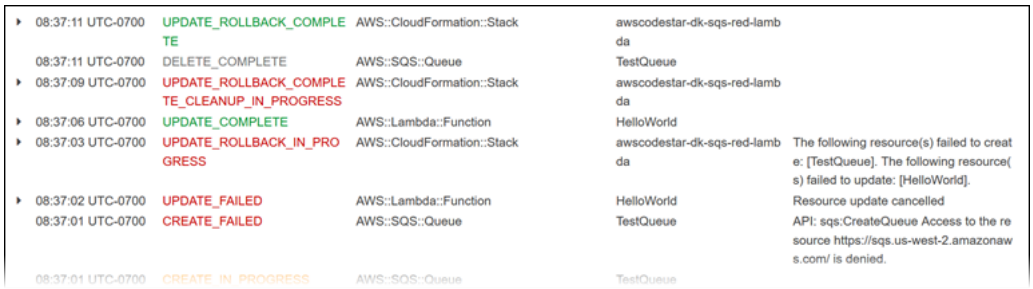
步骤 4：监控 AWS CloudFormation 堆栈更新

1. 在您的项目的管道启动部署阶段时，AWS CloudFormation 堆栈更新启动。您可以在 AWS CodeStar 控制面板上选择您的管道中的 AWS CloudFormation 阶段以查看堆栈更新。

   故障排除：

   如果缺少所需资源权限，则堆栈更新会失败。在项目的管道的 AWS CodeStar 控制面板视图中查看失败状态。

   选择管道的部署阶段中的 CloudFormation 链接以在 AWS CloudFormation 控制台中对失败进行问题排查。在控制台的事件列表中，选择您的项目以查看堆栈创建详细信息。有一条包含失败详细信息的消息。在此示例中，缺少 `sqs:CreateQueue` 权限。

08:37:11 UTC-0700	UPDATE_ROLLBACK_COMPLETE	AWS::CloudFormation::Stack	awscodestar-dk-sqs-red-lambda	
08:37:11 UTC-0700	DELETE_COMPLETE	AWS::SQS::Queue	TestQueue	
08:37:09 UTC-0700	UPDATE_ROLLBACK_COMPLETE_CLEANUP_IN_PROGRESS	AWS::CloudFormation::Stack	awscodestar-dk-sqs-red-lambda	
08:37:06 UTC-0700	UPDATE_COMPLETE	AWS::Lambda::Function	HelloWorld	
08:37:03 UTC-0700	UPDATE_ROLLBACK_IN_PROGRESS	AWS::CloudFormation::Stack	awscodestar-dk-sqs-red-lambda	The following resource(s) failed to create: [TestQueue]. The following resource(s) failed to update: [HelloWorld].
08:37:02 UTC-0700	UPDATE_FAILED	AWS::Lambda::Function	HelloWorld	Resource update cancelled
08:37:01 UTC-0700	CREATE_FAILED	AWS::SQS::Queue	TestQueue	API: sqs:CreateQueue Access to the resource https://sqs.us-west-2.amazonaws.com/ is denied.
08:37:01 UTC-0700	CREATE_IN_PROGRESS	AWS::SQS::Queue	TestQueue	

   通过编辑附加到项目的 AWS CloudFormation 工作线程角色的策略来添加任何缺少的权限。请参阅步骤 1：在 IAM 中编辑 CloudFormation 工作线程角色。

2. 在您的管道成功运行后，将在 AWS CloudFormation 堆栈中创建资源。在 AWS CloudFormation 的资源列表中，查看为您的项目创建的资源。在此示例中，TestQueue 队列在资源部分中列出。

   AWS CloudFormation 中提供了队列 URL。队列 URL 遵循以下格式：

```
https://{REGION_ENDPOINT}/queue.|api-domain|/{YOUR_ACCOUNT_NUMBER}/
{YOUR_QUEUE_NAME}
```

有关更多信息，请参阅发送 Amazon SQS 消息、从 Amazon SQS 队列接收消息和从 Amazon SQS 队列删除消息。

步骤 5：使用内联策略添加资源权限

通过向用户的角色添加适当的内联策略来授予团队成员对您的新资源的访问权限。并非所有资源都需要您添加权限。要执行以下步骤，您必须已作为根用户、账户中的管理员用户或者具有 `AdministratorAccess` 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

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：

```
 {
  "Action": [
    "sqs:CreateQueue",
    "sqs:DeleteQueue",
    "sqs:GetQueueAttributes",
    "sqs:SetQueueAttributes",
    "sqs:ListQueues",
    "sqs:GetQueueUrl"
  ],
  "Resource": [
    "*"
  ],
  "Effect": "Allow"
}
```

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您的策略授予的权限。
8. 选择创建策略可保存您的新策略。

向项目添加 IAM 角色

从太平洋夏令时 2018 年 12 月 6 日开始，您可以在应用程序堆栈 (template.yml) 中定义自己的角色和策略。为了降低权限升级和破坏性操作的风险，您需要为您创建的每个 IAM 实体设置项目特定的权限边界。如果您有包含多个函数的 Lambda 项目，最好为每个函数创建一个 IAM 角色。

向您的项目添加 IAM 角色

1. 编辑您的项目的 `template.yml` 文件。
2. 在 `Resources:` 部分中，使用下例中的格式添加您的 IAM 资源：

```
  SampleRole:
  Description: Sample Lambda role
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
      - Effect: Allow
        Principal:
          Service: [lambda.amazonaws.com]
        Action: sts:AssumeRole
    ManagedPolicyArns:
      -  arn:aws:iam::aws:policy/service-role/AWSLambdaBasicExecutionRole
    PermissionsBoundary: !Sub 'arn:${AWS::Partition}:iam::${AWS::AccountId}:policy/
CodeStar_${ProjectId}_PermissionsBoundary'
```

3. 通过管道发布您的更改，并验证是否成功。

将 Prod 阶段和终端节点添加到项目

使用此部分中的过程，将新的生产 (Prod) 阶段添加到您的管道，并在您的管道的 Deploy 和 Prod 阶段之间添加一个手动审批阶段。这样会在您的项目管道运行时创建一个额外的资源堆栈。

> **Note**
>
> 在以下情况下，您可以使用这些过程：
>
> - 对于在 2018 年 8 月 3 日之后创建的项目，AWS CodeStar 已在项目存储库中为您的 Amazon EC2、Elastic Beanstalk 或 Lambda 项目预配置一个 `/template.yml` 文件。
> - 对于在太平洋夏令时 2018 年 12 月 6 日之后创建的项目，AWS CodeStar 已为您的项目预配置权限边界策略。

所有 AWS CodeStar 项目使用一个 AWS CloudFormation 模板文件，该文件用于为您的应用程序的 AWS 运行时依赖项（如 Linux 实例和 Lambda 函数）建模。`/template.yml` 文件存储在您的源存储库中。

在 `/template.yml` 文件中，使用 `Stage` 参数为项目管道中的新阶段添加一个资源堆栈。

```
 Stage:
   Type: String
   Description: The name for a project pipeline stage, such as Staging or Prod, for
which resources are provisioned and deployed.
   Default: ''
```

该 `Stage` 参数应用于具有资源中引用的项目 ID 的所有指定资源。例如，以下角色名称是模板中的指定资源：

```
RoleName: !Sub 'CodeStar-${ProjectId}-WebApp${Stage}'
```

先决条件

使用 AWS CodeStar 控制台中的模板选项创建项目。

确保您的 IAM 用户具有以下权限：

- 项目 AWS CloudFormation 角色的 `iam:PassRole`。
- 项目工具链角色的 `iam:PassRole`。
- `cloudformation:DescribeStacks`
- `cloudformation:ListChangeSets`

仅适用于 Elastic Beanstalk 或 Amazon EC2 项目：

- `codedeploy:CreateApplication`
- `codedeploy:CreateDeploymentGroup`
- `codedeploy:GetApplication`
- `codedeploy:GetDeploymentConfig`
- `codedeploy:GetDeploymentGroup`
- `elasticloadbalancing:DescribeTargetGroups`

主题

步骤 1：在 CodeDeploy 中新建部署组（仅限 Amazon EC2 项目）

选择您的 CodeDeploy 应用程序，然后添加与新实例关联的新部署组。

> **Note**
>
> 如果您的项目是 Lambda 或 Elastic Beanstalk 项目，可以跳过此步骤。

1. 从 https://console.aws.amazon.com/codedeploy 打开 CodeDeploy 控制台。
2. 选择在 AWS CodeStar 中创建项目时为项目生成的 CodeDeploy 应用程序。
3. 在 Deployment groups 下，选择 Create deployment group。
4. 在部署组名称中，输入 ***<project-id>-prod-Env***。

5. 在服务角色中，为您的 AWS CodeStar 项目选择工具链工作线程角色。
6. 在部署类型下，选择就地。
7. 在环境配置下，选择 Amazon EC2 实例 选项卡。
8. 在标签组下的密钥下，选择 `aws:cloudformation:stack-name`。在值下，选择 `awscodestar-<projectid>-infrastructure-prod`（要为 GenerateChangeSet 操作创建的堆栈）。
9. 在部署设置中，选择 `CodeDeployDefault.AllAtOnce`。
10. 清除选择负载均衡器。
11. 选择创建部署组。

    现在，您的第二个部署组已创建。

步骤 2：为 Prod 阶段添加新的管道阶段

添加一个与您项目的 Deploy 阶段具有一组相同部署操作的新阶段。例如，Amazon EC2 项目的新的 Prod 阶段应与为该项目创建的部署阶段具有相同的操作。

从 Deploy 阶段复制参数和字段

1. 从您的 AWS CodeStar 项目控制面板中，选择管道详细信息以在 CodePipeline 控制台中打开您的管道。
2. 选择编辑。
3. 在 Deploy 阶段中，选择编辑阶段。
4. 选择 GenerateChangeSet 操作上的编辑图标。记下以下字段中的值。在创建您的新操作时，将使用这些值。

   - 堆栈名称
   - 更改集名称
   - 模板
   - 模板配置
   - 输入构件
5. 展开高级，然后在参数中，复制您的项目的参数。将这些参数粘贴到您的新操作。例如，以 JSON 格式复制此处显示的参数：

   - Lambda 项目：

```
{
  "ProjectId":"MyProject"
}
```

- Amazon EC2 项目：

```
{

  "ProjectId":"MyProject",
  "InstanceType":"t2.micro",
  "WebAppInstanceProfile":"awscodestar-MyProject-WebAppInstanceProfile-
EXAMPLEY5VSFS",
  "ImageId":"ami-EXAMPLE1",
  "KeyPairName":"my-keypair",
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE1"
}
```

- Elastic Beanstalk 项目：

```
{
  "ProjectId":"MyProject",
  "InstanceType":"t2.micro",
  "KeyPairName":"my-keypair",
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE",
  "SolutionStackName":"64bit Amazon Linux 2018.03 v3.0.5 running Tomcat 8 Java
 8",
  "EBTrustRole":"CodeStarWorker-myproject-EBService",
  "EBInstanceProfile":"awscodestar-myproject-EBInstanceProfile-11111EXAMPLE"
}
```

6. 在阶段编辑窗格中，选择取消。

在新的 Prod 阶段中创建 GenerateChangeSet 操作

> **Note**
>
> 添加新操作后但在仍处于编辑模式时，如果重新打开新操作进行编辑，则可能不会显示某些字段。您还可能看到以下内容：堆栈 stack-name 不存在

此错误不会阻止您保存管道。但是，要还原缺少的字段，您必须删除并重新添加新操作。保存和运行管道后，堆栈状态将被识别和错误不会重新出现。

1. 如果未显示您的管道，则从 AWS CodeStar 项目控制面板中，选择管道详细信息以在控制台中打开您的管道。
2. 选择编辑。
3. 在示意图底部，选择 + 添加阶段。
4. 输入阶段名称（例如 **Prod**），然后选择 + 添加操作组。
5. 在操作名称中，输入名称（例如 **GenerateChangeSet**）。
6. 在操作提供商中，选择 AWS CloudFormation。
7. 在操作模式中，选择创建或替换更改集。
8. 在堆栈名称中，为将要通过此操作创建的 AWS CloudFormation 堆栈输入一个新名称。从与 Deploy 堆栈名称完全相同的名称开始，然后添加 **-prod**：

   - Lambda 项目：`awscodestar-<project_name>-lambda-prod`
   - Amazon EC2 和 Elastic Beanstalk 项目：`awscodestar-<project_name>-infrastructure-prod`

   > **Note**
   >
   > 堆栈名称必须完全以 **awscodestar-<project_name>-** 开头，否则堆栈创建会失败。

9. 在更改集名称中，输入与现有 Deploy 阶段中提供的更改集名称相同的名称（例如 **pipeline-changeset**）。
10. 在输入项目 中，选择构建项目。
11. 在模板中，输入与现有 Deploy 阶段中提供的模板名称相同的名称（例如 **<project-ID>-BuildArtifact::template.yml**）。
12. 在模板配置中，输入与 Deploy 阶段中提供的更改模板配置文件名相同的名称（例如 **<project-ID>-BuildArtifact::template-configuration.json**）。
13. 在功能中，选择 CAPABILITY_NAMED_IAM。
14. 在角色名称中，选择您项目的 AWS CloudFormation 工作线程角色的名称。
15. 展开高级，然后在参数 中，为您的项目粘贴参数。为 Amazon EC2 项目包含 `Stage` 参数（此处所示为 JSON 格式）：

```
{

  "ProjectId":"MyProject",
  "InstanceType":"t2.micro",
  "WebAppInstanceProfile":"awscodestar-MyProject-WebAppInstanceProfile-
EXAMPLEY5VSFS",
  "ImageId":"ami-EXAMPLE1",
  "KeyPairName":"my-keypair",
  "SubnetId":"subnet-EXAMPLE",
  "VpcId":"vpc-EXAMPLE1",
  "Stage":"Prod"
}
```

> **Note**
>
> 确保为该项目粘贴所有参数，而不仅仅是新参数或您想要更改的参数。

16. 选择保存。
17. 在 AWS CodePipeline 窗格中，选择保存管道更改，然后选择保存更改。

> **Note**
>
> 可能会显示一条消息，通知您更改检测资源已被删除和添加。确认消息并继续本教程的下一步。

查看已更新的管道。

在新的 Prod 阶段中创建 ExecuteChangeSet 操作

1. 如果您尚未查看您的管道，则从 AWS CodeStar 项目控制面板中，选择管道详细信息以在控制台中打开您的管道。
2. 选择编辑。
3. 在您的新 Prod 阶段中，在新的 GenerateChangeSet 操作后面，选择 + 添加操作组。
4. 在操作名称中，输入名称（例如 **`ExecuteChangeSet`**）。
5. 在操作提供商中，选择 AWS CloudFormation。
6. 在操作模式中，选择执行更改集。

7. 在堆栈名称中，输入您在 GenerateChangeSet 操作中输入的 AWS CloudFormation 堆栈的新名称（例如 **awscodestar-<project-ID>-infrastructure-prod**）。
8. 在更改集名称中，输入与部署阶段中使用的更改集名称相同的名称（例如 **pipeline-changeset**）。
9. 选择完成。
10. 在 AWS CodePipeline 窗格中，选择保存管道更改，然后选择保存更改。

> **Note**
>
> 可能会显示一条消息，通知您更改检测资源已被删除和添加。确认消息并继续本教程的下一步。

查看已更新的管道。

在新的 Prod 阶段中创建 CodeDeploy 部署操作（仅限 Amazon EC2 项目）

1. 在您的 Prod 阶段中的新操作后面，选择 + 操作。
2. 在操作名称中，输入名称（例如 **Deploy**）。
3. 在操作提供商中，选择 AWS CodeDeploy。
4. 在应用程序名称中，为您的项目选择 CodeDeploy 应用程序的名称。
5. 在部署组中，选择您在步骤 2 中创建的新 CodeDeploy 部署组的名称。
6. 在输入项目中，选择在现有阶段中使用的相同构建项目。
7. 选择完成。
8. 在 AWS CodePipeline 窗格中，选择保存管道更改，然后选择保存更改。查看已更新的管道。

步骤 3：添加手动审批阶段

作为最佳实践，在您的新生产阶段前面添加手动审批阶段。

1. 在左上方，选择编辑。
2. 在您的管道图表中的 Deploy 和 Prod 部署阶段之间，选择 + 添加阶段。
3. 在编辑阶段上，输入阶段名称（例如 **Approval**），然后选择 + 添加操作组。
4. 在操作名称中，输入名称（例如 **Approval**）。

5. 在 Approval type 中，选择 Manual approval。
6. （可选）在配置下的 SNS 主题 ARN 中，选择您已创建并订阅的 SNS 主题。
7. 选择 Add Action。
8. 在 AWS CodePipeline 窗格中，选择保存管道更改，然后选择保存更改。查看已更新的管道。
9. 要提交所做的更改并开始管道构建，请选择发布更改，然后选择发布。

步骤 4：推送更改和监控 AWS CloudFormation 堆栈更新

1. 当您的管道运行时，您可以使用此处的步骤来跟踪新阶段的堆栈和端点创建过程。
2. 在管道启动 Deploy 阶段时，AWS CloudFormation 堆栈更新将启动。您可以在 AWS CodeStar 控制面板上选择您的管道中的 AWS CloudFormation 阶段，以查看堆栈更新通知。要查看堆栈创建详细信息，请在控制台的事件列表中选择您的项目。
3. 在您的管道成功完成后，将在 AWS CloudFormation 堆栈中创建资源。在 AWS CloudFormation 控制台中，选择您的项目的基础设施堆栈。堆栈名称采用以下格式：

   - Lambda 项目：`awscodestar-<project_name>-lambda-prod`
   - Amazon EC2 和 Elastic Beanstalk 项目：`awscodestar-<project_name>-infrastructure-prod`

   在 AWS CloudFormation 控制台的资源列表中，查看为您的项目创建的资源。在本示例中，新的 Amazon EC2 实例显示在资源部分中。
4. 访问您的生产阶段的终端节点：

   - 对于 Elastic Beanstalk 项目，请在 AWS CloudFormation 控制台中打开新堆栈并展开资源。选择 Elastic Beanstalk 应用程序。此链接将打开 Elastic Beanstalk 控制台。选择环境。在 URL 中选择 URL，以便在浏览器中打开终端节点。
   - 对于 Lambda 项目，请在 AWS CloudFormation 控制台中打开新堆栈并展开资源。选择 API Gateway 资源。该链接将在 API Gateway 控制台中打开。选择阶段。在调用 URL 中选择 URL，以便在浏览器中打开终端节点。
   - 对于 Amazon EC2 项目，请在 AWS CodeStar 控制台的项目资源列表中选择新的 Amazon EC2 实例。该链接将在 Amazon EC2 控制台的实例页面中打开。选择说明选项卡，复制公有 DNS (IPv4) 中的 URL，然后在浏览器中打开该 URL。
5. 验证您的更改是否已部署。

在 AWS CodeStar 项目中安全地使用 SSM 参数。

许多客户都会在 Systems Manager Parameter Store 参数中存储机密信息，例如凭证。现在，您可以在 AWS CodeStar 项目中安全地使用这些参数。例如，您可能希望在为 CodeBuild 构建规范时，或者在工具链堆栈 (template.yml) 中定义应用程序资源时使用 SSM 参数。

要在 AWS CodeStar 项目中使用 SSM 参数，必须使用 AWS CodeStar 项目 ARN 手动标记参数。您还必须为 AWS CodeStar 工具链工作线程角色提供适当的权限，以访问您标记的参数。

开始前的准备工作

- 创建新的或识别现有 Systems Manager 参数，其中包含您要访问的信息。
- 确定您要使用的 AWS CodeStar 项目，或创建新项目。
- 记下 CodeStar 项目 ARN。它类似于以下内容：`arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*

使用 AWS CodeStar 项目 ARN 标记参数

有关分步指导，请参阅标记 Systems Manager 参数。

1. 在键中，输入 `awscodestar:projectArn`。
2. 在值中，输入 CodeStar 的项目 ARN：`arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*。
3. 选择保存。

现在，您可以在 template.yml 文件中引用 SSM 参数。如果要将其与工具链工作线程角色一起使用，您将需要授予其他权限。

授予权限以在您的 AWS CodeStar 项目工具链中使用标记的参数

> **Note**
>
> 这些步骤仅适用于在太平洋夏令时 2018 年 12 月 6 日之后创建的项目。

1. 为您要使用的项目打开 AWS CodeStar 项目控制面板。

2. 单击项目以查看创建的资源列表，并找到工具链工作线程角色。它是一个 IAM 资源，其名称的格式为：role/CodeStarWorker-*project-id*-ToolChain。
3. 在 IAM 控制台中单击 ARN 以将其打开。
4. 找到 ToolChainWorkerPolicy 并展开它（如有必要）。
5. 单击编辑策略。
6. 在 Action: 下，添加以下行：

   ```
   ssm:GetParameter*
   ```
7. 单击“查看策略”，然后单击“保存更改”。

对于在太平洋夏令时 2018 年 12 月 6 日之前创建的项目，您需要将以下权限添加到每个服务的工作线程角色。

```
        {
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "*",
            "Effect": "Allow",
            "Condition": {
                "StringEquals": {
                    "ssm:ResourceTag/awscodestar:projectArn": "arn:aws:codestar:region-
id:account-id:project/project-id"
                }
            }
        }
```

转移 AWS Lambda 项目的流量

AWS CodeDeploy 支持 AWS CodeStar 无服务器项目中的 AWS Lambda 函数的函数版本部署。AWS Lambda 部署可将传入流量从现有 Lambda 函数转移到更新的 Lambda 函数版本。您可能希望测试更新的 Lambda 函数，方法是部署单独版本，然后将部署回滚到第一个版本（如果需要）。

按照此部分中的步骤修改 AWS CodeStar 项目模板并更新 CodeStarWorker 角色的 IAM 权限。此任务在 AWS CloudFormation 中启动创建别名的 AWS Lambda 函数的自动响应，然后指示 AWS CodeDeploy 将流量转移到更新的环境。

> **Note**
>
> 仅当您在 2018 年 12 月 12 日之前已创建 AWS CodeStar 项目的情况下完成这些步骤。

AWS CodeDeploy 有三个部署选项，允许您将流量转移到应用程序中的 AWS Lambda 函数版本：

- Canary：流量将通过两次递增进行转移。您可以从预定义的金丝雀部署选项中选择，这些选项指定在第一次增量中转移到更新后的 Lambda 函数版本的流量百分比以及以分钟为单位的间隔；然后指定在第二次增量中转移剩余的流量。
- 线性部署：流量使用相等的增量转移，在每次递增之间间隔的分钟数相同。您可以从预定义的线性选项中进行选择，这些选项指定在每次增量中转移的流量百分比以及每次增量之间的分钟数。流量使用相等的增量转移，在每次递增之间间隔的分钟数相同。您可以从预定义的线性选项中进行选择，这些选项指定在每次增量中转移的流量百分比以及每次增量之间的分钟数。
- 一次性部署：所有流量均从原始 Lambda 函数一次性地转移到更新后的 Lambda 函数版本。

部署首选项类型
Canary10Percent30Minutes
Canary10Percent5Minutes
Canary10Percent10Minutes
Canary10Percent15Minutes
Linear10PercentEvery10Minutes
Linear10PercentEvery1Minute
Linear10PercentEvery2Minutes
Linear10PercentEvery3Minutes
AllAtOnce

有关 AWS Lambda 计算平台上的 AWS CodeDeploy 部署的更多信息，请参阅 AWS Lambda 计算平台上的部署。

有关 AWS SAM 的更多信息，请参阅 GitHub 上的 AWS 无服务器应用程序模型（AWS SAM）。

先决条件：

在创建无服务器项目时，请选择使用 Lambda 计算平台的任何模板。您必须以管理员身份登录才能执行步骤 4-6。

步骤 1：修改 SAM 模板以添加 AWS Lambda 版本部署参数

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 使用 `template.yml` 文件创建项目或选择现有项目，然后打开代码页面。在存储库的顶层中，记下要修改的名为 `template.yml` 的 SAM 模板的位置。
3. 打开 IDE 或本地存储库中的 `template.yml` 文件。复制下面的文本以向该文件中添加 `Globals` 部分。本教程中的示例文本选择 `Canary10Percent5Minutes` 选项。

```
Globals:
  Function:
    AutoPublishAlias: live
    DeploymentPreference:
      Enabled: true
      Type: Canary10Percent5Minutes
```

此示例显示了添加 `Globals` 部分后的已修改模板：

```
AWSTemplateFormatVersion: 2010-09-09
Transform:
- AWS::Serverless-2016-10-31
- AWS::CodeStar

Parameters:
  ProjectId:
    Type: String
    Description: CodeStar projectId used to associate new resources to team members

Globals:
  Function:
    AutoPublishAlias: live
    DeploymentPreference:
      Enabled: true
      Type: Canary10Percent5Minutes

Resources:
  HelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.6
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
```

有关更多信息，请参阅 SAM 模板的 Globals 部分参考指南。

步骤 2：编辑 AWS CloudFormation 角色以添加权限

1. 登录到 AWS Management Console，然后通过以下网址打开 AWS CodeStar 控制台：https://console.aws.amazon.com/codestar/。

   > **Note**
   >
   > 您必须使用与在 设置 AWS CodeStar 中创建或标识的 IAM 用户关联的凭证登录 AWS Management Console。此用户必须附加了名为 AWS**AWSCodeStarFullAccess** 的 托管策略。

2. 选择您的现有无服务器项目，然后打开项目资源页面。
3. 在资源下，选择为 CodeStarWorker/AWS CloudFormation 角色创建的 IAM 角色。此角色将在 IAM 控制台中打开。
4. 在 Permissions 选项卡上的 Inline Policies 中，选择您的服务角色策略所在行中的 Edit Policy。选择 JSON 选项卡以编辑 JSON 格式的策略。

   > **Note**
   >
   > 您的服务角色名为 `CodeStarWorkerCloudFormationRolePolicy`。

5. 在 JSON 字段中，在 `Statement` 元素中添加以下策略语句。将 *`region`* 和 *`id`* 占位符替换为您的区域和账户 ID。

```
{
  "Action": [
    "s3:GetObject",
    "s3:GetObjectVersion",
    "s3:GetBucketVersioning"
  ],
  "Resource": "*",
  "Effect": "Allow"
},
{
  "Action": [
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::codepipeline*"
```

```
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "lambda:*"
    ],
    "Resource": [
      "arn:aws:lambda:region:id:function:*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "apigateway:*"
    ],
    "Resource": [
      "arn:aws:apigateway:region::*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:GetRole",
      "iam:CreateRole",
      "iam:DeleteRole",
      "iam:PutRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::id:role/*"
    ],
    "Effect": "Allow"
  },
  {
    "Action": [
      "iam:AttachRolePolicy",
      "iam:DeleteRolePolicy",
      "iam:DetachRolePolicy"
    ],
    "Resource": [
      "arn:aws:iam::id:role/*"
    ],
    "Effect": "Allow"
  },
```

```
{
  "Action": [
    "iam:PassRole"
  ],
  "Resource": [
    "*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "codedeploy:CreateApplication",
    "codedeploy:DeleteApplication",
    "codedeploy:RegisterApplicationRevision"
  ],
  "Resource": [
    "arn:aws:codedeploy:region:id:application:*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "codedeploy:CreateDeploymentGroup",
    "codedeploy:CreateDeployment",
    "codedeploy:DeleteDeploymentGroup",
    "codedeploy:GetDeployment"
  ],
  "Resource": [
    "arn:aws:codedeploy:region:id:deploymentgroup:*"
  ],
  "Effect": "Allow"
},
{
  "Action": [
    "codedeploy:GetDeploymentConfig"
  ],
  "Resource": [
    "arn:aws:codedeploy:region:id:deploymentconfig:*"
  ],
  "Effect": "Allow"
}
```

6. 选择查看策略以确保策略不包含错误。当策略正确无误时，选择保存更改。

步骤 3：提交并推送模板更改以启动 AWS Lambda 版本转移

1. 提交并推送您在步骤 1 中保存的 `template.yml` 文件中的更改。

   > **Note**
   >
   > 这会启动您的管道。如果您在更新 IAM 权限之前提交更改，则您的管道会启动，并且 AWS CloudFormation 堆栈更新会遇到回滚堆栈更新的错误。如果发生这种情况，请在更正权限后重新启动您的管道。

2. 在您的项目的管道启动部署阶段时，AWS CloudFormation 堆栈更新启动。要在部署启动时查看堆栈更新通知，请在 AWS CodeStar 控制面板上选择您的管道中的 AWS CloudFormation 阶段。

   在堆栈更新期间，AWS CloudFormation 会自动更新项目资源，如下所示：

   - AWS CloudFormation 通过创建别名的 Lambda 函数、事件挂钩和资源来处理 `template.yml` 文件。
   - AWS CloudFormation 调用 Lambda 来创建函数的新版本。
   - AWS CloudFormation 创建 AppSpec 文件并调用 AWS CodeDeploy 来转移流量。

   有关在 SAM 中发布别名的 Lambda 函数的更多信息，请参阅 AWS 无服务器应用程序模型 (SAM) 模板参考。有关 AWS CodeDeploy AppSpec 文件中的事件挂钩和资源的更多信息，请参阅 AppSpec 的“resources”部分（仅限 AWS Lambda 部署）和用于 AWS Lambda 部署的 AppSpec 的“hooks”部分。

3. 在您的管道成功完成后，将在 AWS CloudFormation 堆栈中创建资源。在项目页面上的项目资源列表中，查看 AWS CodeDeploy 应用程序、AWS CodeDeploy 部署组以及为您的项目创建的 AWS CodeDeploy 服务角色资源。

4. 要创建新版本，请更改存储库中的 Lambda 函数。新部署将根据 SAM 模板中指示的部署类型启动并转移流量。要查看正在转移到新版本的流量状态，请在项目页面上的项目资源列表中，选择 AWS CodeDeploy 部署的链接。

5. 要查看有关每个修订的详细信息，请在修订下，选择 AWS CodeDeploy 部署组的链接。

6. 在您的本地工作目录中，您可以更改 AWS Lambda 函数并将更改提交到项目存储库。AWS CloudFormation 支持 AWS CodeDeploy 以相同方式管理下一个修订。有关重新部署、停止或回滚 Lambda 部署的更多信息，请参阅 AWS Lambda 计算平台上的部署。

将您的 AWS CodeStar 项目转换为生产用途

使用 AWS CodeStar 项目创建应用程序并查看 AWS CodeStar 提供的内容后，您需要将项目转换为生产用途。实现此目标的一种方法是在 AWS CodeStar 外部复制应用程序的 AWS 资源。您仍然需要存储库、构建项目、管道和部署，但无需让 AWS CodeStar 为您创建，您可以使用 AWS CloudFormation 重新创建它们。

> **Note**
>
> 首先使用任一 AWS CodeStar 快速入门创建或查看类似项目，将其用作您自己的项目的模板，以确保包括所需的资源和策略，这可能会有所帮助。

AWS CodeStar 项目是源代码和为部署代码而创建的资源的组合。帮助您构建、发布和部署您的代码的资源集合称为工具链资源。在创建项目期间，AWS CloudFormation 模板将工具链资源预配置在一个持续集成/持续部署 (CI/CD) 管道中。

当您使用控制台创建项目时，将为您创建工具链模板。当您使用 AWS CLI 创建项目时，创建用于创建您的工具链资源的工具链模板。

完整的工具链需要以下推荐的资源：

1. 包含您的源代码的 CodeCommit 或 GitHub 存储库。
2. 配置为侦听存储库更改的 CodePipeline 管道。
   a. 当您使用 AWS CodeBuild 运行单元测试或集成测试时，建议您向管道中添加构建阶段以创建构建构件。
   b. 建议您向管道中添加使用 CodeDeploy 或 AWS CloudFormation 将构建构件和源代码部署到运行时基础设施的部署阶段。

   > **Note**
   >
   > 因为 CodePipeline 要求管道中至少具有两个阶段，并且第一个阶段必须是源阶段，所以请添加构建或部署阶段作为第二个阶段。

主题

创建 GitHub 存储库

您可以通过在工具链模板中定义 GitHub 库来创建它。确保您已经为包含源代码的 ZIP 文件创建了位置，以便可以将代码上传到存储库。此外，您必须已经在 GitHub 中创建了个人访问令牌，以便 AWS 可以代表您连接到 GitHub。除了 GitHub 的个人访问令牌外，您还必须具有传入的 `Code` 对象的 `s3.GetObject` 权限。

要指定公共 GitHub 存储库，请将如下代码添加到 AWS CloudFormation 中的工具链模板中。

```
GitHubRepo:
Condition: CreateGitHubRepo
Description: GitHub repository for application source code
Properties:
  Code:
    S3:
      Bucket: MyCodeS3Bucket
      Key: MyCodeS3BucketKey
  EnableIssues: true
  IsPrivate: false
  RepositoryAccessToken: MyGitHubPersonalAccessToken
  RepositoryDescription: MyAppCodeRepository
  RepositoryName: MyAppSource
  RepositoryOwner: MyGitHubUserName
Type: AWS::CodeStar::GitHubRepository
```

该代码指定以下信息：

- 您要包含的代码的位置，该位置必须是 Amazon S3 存储桶。
- 是否要在 GitHub 存储库上启用问题。
- GitHub 存储库是否为私有。
- 您创建的 GitHub 个人访问令牌。
- 您要创建的存储库的描述、名称和拥有者。

有关要指定哪些信息的完整详细信息，请参阅 AWS CloudFormation 用户指南中的 AWS::CodeStar::GitHubRepository。

在 AWS CodeStar 中使用项目标签

在 AWS CodeStar 中，可以为项目关联标签。标签可帮助您管理项目。例如，您可能向组织为测试版而进行的任何项目添加键为 `Release`、值为 `Beta` 的标签。

向项目添加标签

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在侧导航窗格中选择设置。
2. 在标签中，选择编辑。
3. 在键中，输入标签的名称。在键中输入标签的值。
4. （可选）选择添加标签，以添加更多标签。
5. 添加完标签后，选择保存。

从项目中删除标签

1. 在项目在 AWS CodeStar 控制台中打开的情况下，在侧导航窗格中选择设置。
2. 在标签中，选择编辑。
3. 在标签中找到要移除的标签，然后选择移除标签。
4. 选择保存。

获取项目的标签列表

使用 AWS CLI 运行 AWS CodeStar list-tags-for-project 命令，指定项目的名称：

```
aws codestar list-tags-for-project --id my-first-projec
```

如果成功，输出中将显示标签列表，类似于以下内容：

```
{
  "tags": {
    "Release": "Beta"
  }
}
```

删除 AWS CodeStar 项目

如果您不再需要某个项目，则可删除此项目及其资源以便您在 AWS 中不会产生任何额外费用。删除一个项目时，将删除该项目中的所有团队成员。这将从团队成员的 IAM 用户中删除其项目角色，但不会更改其在 AWS CodeStar 中的用户配置文件。您可以使用 AWS CodeStar 控制台或 AWS CLI 删除项目。删除项目需要 AWS CodeStar 服务角色 `aws-codestar-service-role`，该角色必须是未经修改，并且可以由 AWS CodeStar 代入。

Important

从 AWS CodeStar 中删除项目这一操作是无法撤消的。默认情况下，项目的所有 AWS 资源都将从您的 AWS 账户中删除，包括：

- 项目的 CodeCommit 存储库以及该存储库中存储的所有内容。
- 为项目及其资源配置的 AWS CodeStar 项目角色和关联的 IAM 策略。
- 为项目创建的所有 Amazon EC2 实例。
- 部署应用程序和关联资源，例如：
  - CodeDeploy 应用程序和关联的部署组。
  - AWS Lambda 函数和关联的 API Gateway API。
  - AWS Elastic Beanstalk 应用程序和关联的环境。
- CodePipeline 中项目的持续部署管道。
- 与项目关联的 AWS CloudFormation 堆栈。
- 使用 AWS CodeStar 控制台创建的任何 AWS Cloud9 开发环境。环境中所有未提交的代码更改都将丢失。

要删除所有项目资源以及项目，请选中删除资源复选框。如果清除此选项，将从 AWS CodeStar 中删除项目，并从 IAM 中删除支持访问这些资源的项目角色，但将保留所有其他资源。AWS 中的这些资源可能继续产生费用。如果您决定不再需要这些资源中的一个或多个资源，则必须手动将其删除。有关更多信息，请参阅项目删除：已删除 AWS CodeStar 项目，但资源仍存在。
如果您决定在删除项目时保留资源，作为最佳实践，请复制项目详细信息页中的资源列表。这样一来，您将获得已保留的所有资源的记录，即使项目不再存在。

主题

在 AWS CodeStar 中删除项目（控制台）

您可以使用 AWS CodeStar 控制台删除项目。

在 AWS CodeStar 中删除项目

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 在导航窗格中，选择项目。
3. 选择要删除的项目，然后选择删除。

   或者打开项目，并在控制台左侧的导航窗格中选择设置。在项目详细信息页上，选择 Delete project。
4. 在删除确认页面中，选择删除。如果您要删除项目资源，请选中删除资源。选择删除。

   可能需要花费几分钟的时间才能删除项目。删除项目后，项目将不再显示在 AWS CodeStar 控制台中的项目列表中。

> **Important**
>
> 如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），即使选中此复选框，也不会删除这些资源。
> 如果任何 AWS CodeStar 托管策略已手动附加到非 IAM 用户的角色，则无法删除您的项目。如果您已将项目的托管策略附加到联合身份用户的角色，则必须先分离策略才能删除项目。有关更多信息，请参阅???。

在 AWS CodeStar 中删除项目 (AWS CLI)

您可以使用 AWS CLI 删除项目。

在 AWS CodeStar 中删除项目

1. 在终端（Linux、macOS 或 Unix）或命令提示符处 (Windows)，运行 delete-project 命令，包括项目的名称。例如，要删除 ID 为 *my-2nd-project* 的项目，请执行以下操作：

```
aws codestar delete-project --id my-2nd-project
```

该命令会返回类似以下内容的输出：

```
{
    "projectArn":"arn:aws:codestar:us-east-2:111111111111:project/my-2nd-project"
}
```

项目不会立即删除。

2. 运行 describe-project 命令，包括项目的名称。例如，检查 ID 为 *my-2nd-project* 的项目的状态：

```
    aws codestar describe-project --id my-2nd-project
```

如果尚未删除项目，则此命令返回类似于以下内容的输出：

```
{
    "name": "my project",
    "id": "my-2nd-project",
    "arn": "arn:aws:codestar:us-west-2:123456789012:project/my-2nd-project",
    "description": "My second CodeStar project.",
    "createdTimeStamp": 1572547510.128,
    "status": {
        "state": "CreateComplete"
    }
}
```

如果删除了项目，则此命令将返回类似于以下内容的输出：

```
    An error occurred (ProjectNotFoundException) when calling the DescribeProject
operation: The project ID was not found: my-2nd-project. Make sure that the
project ID is correct and then try again.
```

3. 运行 list-projects 命令并确认已删除的项目不再出现在与 AWS 账户关联的项目的列表中。

```
aws codestar list-projects
```

与 AWS CodeStar 团队协作

在创建开发项目后，您将需要向其他人授予访问权限以便进行协作。在 AWS CodeStar 中，每个项目均有一个项目团队。一个用户可属于多个 AWS CodeStar 项目并在每个项目中拥有不同的 AWS CodeStar 角色（因而拥有不同的权限）。在 AWS CodeStar 控制台中，用户可查看与您的 AWS 账户关联的所有项目，但他们只能查看和处理他们身为团队成员的那些项目。

团队成员可以为自己选择一个友好名称。他们还可以添加电子邮件地址，以便其他团队成员联系他们。不是所有者的团队成员无法更改其在项目中的 AWS CodeStar 角色。

AWS CodeStar 中的每个项目均有三个角色：

AWS CodeStar 项目中的角色和权限

角色名称	查看项目控制面板和状态	添加/删除/访问项目资源	添加/删除团队成员	删除项目
所有者	x	x	x	x
贡献者	x	x		
查看者	x			

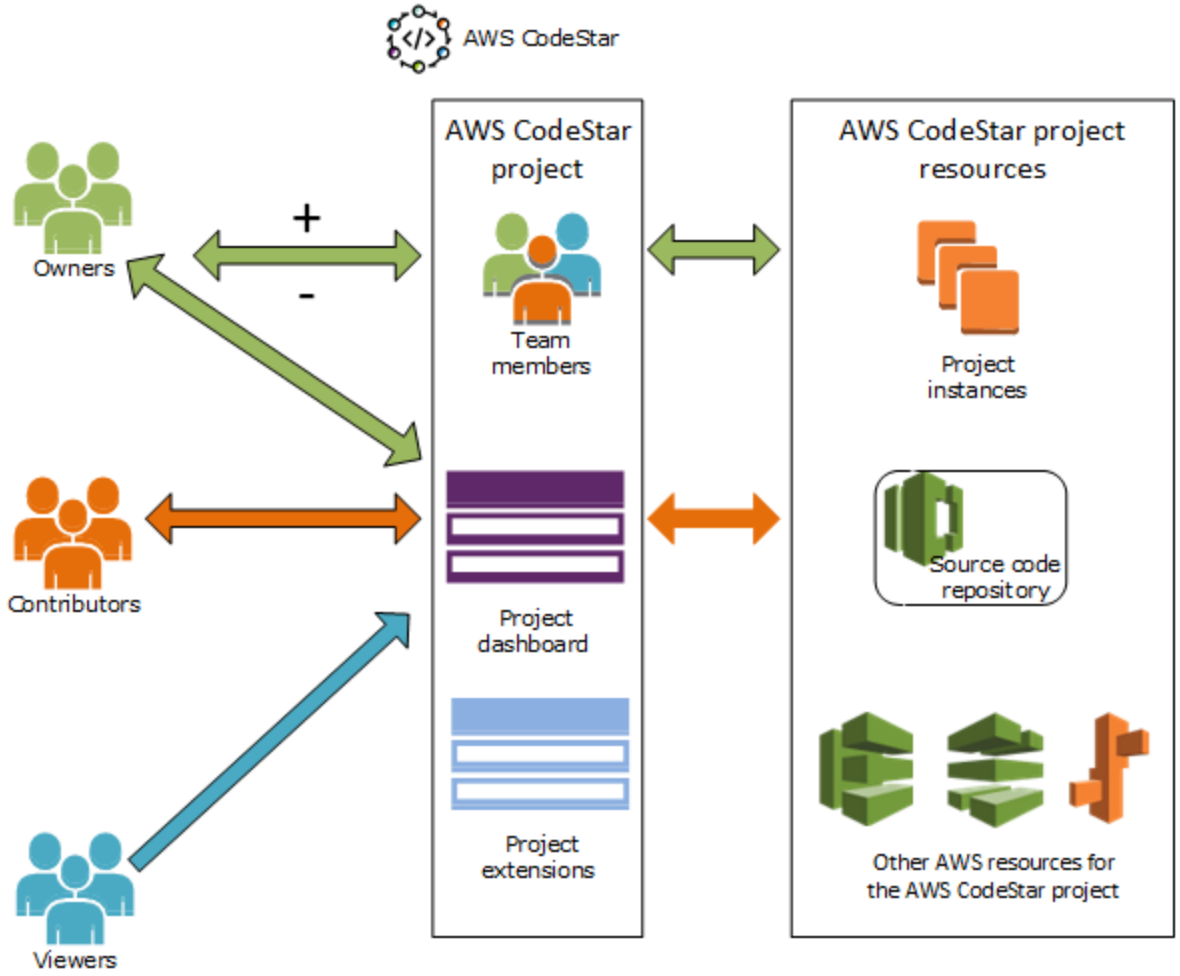
- 所有者：可添加和删除其他团队成员，向项目存储库提供代码（如果代码存储在 CodeCommit 中），授予或拒绝其他团队成员对运行 Linux 且与项目关联的任何 Amazon EC2 实例的远程访问，配置项目控制面板以及删除项目。
- 贡献者：可添加和删除控制面板资源（如 JIRA 磁贴），向项目存储库提供代码（如果代码存储在 CodeCommit 中）以及与控制面板完全交互。无法添加或删除团队成员、授予或拒绝对资源的远程访问或者删除项目。这是应为大多数团队成员选择的角色。
- 查看者：可查看项目控制面板、代码（如果存储在 CodeCommit 中）以及控制面板磁贴上项目及其资源的状态。

> **Important**
>
> 如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），对这些资源的访问由资源提供者而非 AWS CodeStar 控制。有关更多信息，请参阅资源提供者的文档。

有权访问 AWS CodeStar 项目的任何人都可以使用 AWS CodeStar 控制台访问位于 AWS 以外但与此项目有关的资源。
AWS CodeStar 不会自动允许项目团队成员参与项目的任何相关 AWS Cloud9 开发环境。要允许团队成员参与共享环境，请参阅与项目团队成员共享 AWS Cloud9 环境。

每个项目角色均关联一个 IAM 策略。此策略针对您的项目进行自定义以反映其资源。有关这些策略的更多信息，请参阅 AWS CodeStar 基于身份的策略示例。

下图显示每个角色与 AWS CodeStar 项目之间的关系。



主题

- 向 AWS CodeStar 项目添加团队成员
- 管理 AWS CodeStar 团队成员的权限
- 从 AWS CodeStar 项目中删除团队成员

向 AWS CodeStar 项目添加团队成员

如果您在 AWS CodeStar 项目中具有所有者角色或已将 `AWSCodeStarFullAccess` 策略应用于 IAM 用户，则可将其他 IAM 用户添加到项目团队。这是一个将 AWS CodeStar 角色（所有者、贡献者或查看者）应用于用户的简单过程。这些角色基于每个项目并且已进行自定义。例如，项目 A 中的贡献者团队成员对资源具有的权限可能不同于项目 B 中的贡献者团队成员对资源具有的权限。一个团队成员在项目中只能有一个角色。在您添加一个团队成员后，此成员可立即在角色所定义的级别与您的项目进行交互。

AWS CodeStar 角色和团队成员资格的好处包括：

- 您无需在 IAM 中为团队成员手动配置权限。
- 您可以轻松更改团队成员具有的项目访问权的级别。
- 用户仅在为团队成员时才能在 AWS CodeStar 控制台中访问项目。
- 用户具有的项目访问权限由角色定义。

有关团队和 AWS CodeStar 角色的更多信息，请参阅与 AWS CodeStar 团队协作和使用您的 AWS CodeStar 用户配置文件 。

要向项目添加团队成员，您必须具有项目的 AWS CodeStar 所有者角色或具有 `AWSCodeStarFullAccess` 策略。

> **Important**
>
> 添加团队成员不影响此成员访问 AWS 以外的资源（例如，GitHub 存储库或 Atlassian JIRA 中的问题）。这些访问权限由资源提供者而非 AWS CodeStar 控制。有关更多信息，请参阅资源提供者的文档。
>
> 有权访问 AWS CodeStar 项目的任何人都可以使用 AWS CodeStar 控制台访问位于 AWS 以外但与此项目有关的资源。
>
> 将团队成员添加到项目不会自动允许该成员参与该项目的任何相关 AWS Cloud9 开发环境。要允许团队成员参与共享环境，请参阅与项目团队成员共享 AWS Cloud9 环境。
>
> 授予联合身份用户对项目的访问权限涉及将 AWS CodeStar 所有者、贡献者或查看者托管策略手动附加到联合身份用户所代入的角色。有关更多信息，请参阅联合身份用户对 AWS CodeStar 的访问权限。

主题

添加团队成员（控制台）

您可以使用 AWS CodeStar 控制台向项目添加团队成员。如果要添加的人员已有 IAM 用户，则可添加该 IAM 用户。否则，您可以在将该人员添加到项目时为其创建 IAM 用户。

将团队成员添加到 AWS CodeStar 项目（控制台）

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您的项目。
3. 在项目的侧导航栏中，选择团队。
4. 在 Team members 页面上，选择 Add team member。
5. 在 Choose user 中，执行下列操作之一：

   - 如果要添加的人员已有 IAM 用户，请从列表中选择该 IAM 用户名。

   > **Note**
   >
   > 已添加到另一个 AWS CodeStar 项目的用户将显示在现有 AWS CodeStar 用户列表中。

   在项目角色中，选择要此用户的 AWS CodeStar 角色（所有者、贡献者或查看者）。这是只能由项目所有者更改的 AWS CodeStar 项目级角色。此角色在应用于 IAM 用户时将提供访问 AWS CodeStar 项目资源所需的所有权限。它会应用执行以下操作所需的策略：在 IAM 中为存储在 CodeCommit 中的代码创建和管理 Git 凭证，或在 IAM 中为用户上传 Amazon EC2 SSH 密钥。

   > **Important**
   >
   > 您无法提供或更改 IAM 用户的显示名称或电子邮件信息，除非您已经以该用户身份登录到控制台。有关更多信息，请参阅管理 AWS CodeStar 用户配置文件的显示信息 。

   选择添加团队成员。

- 如果要添加到项目的人员没有 IAM 用户，请选择创建新 IAM 用户。您将被重定向到 IAM 控制台，可以在其中创建新的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的创建 IAM 用户。创建 IAM 用户后，返回 AWS CodeStar 控制台，刷新用户列表，然后从下拉列表中选择您创建的 IAM 用户。输入要应用于此新用户的 AWS CodeStar 显示名称、电子邮件地址和项目角色，然后选择添加团队成员。

> **Note**
>
> 为了便于管理，应向至少一个用户分配了该项目的“所有者”角色。

6. 向新团队成员发送以下信息：

- 您的 AWS CodeStar 项目的连接信息。
- 为了从本地计算机访问 CodeCommit 存储库而使用 Git 凭证设置访问权限的说明（如果源代码存储在 CodeCommit 中）。
- 有关用户如何管理其显示名称、电子邮件地址和公有 Amazon EC2 SSH 密钥的信息，如 使用您的 AWS CodeStar 用户配置文件 中所述。
- 一次性密码和连接信息，前提是用户是首次使用 AWS，并且您已为该人员创建 IAM 用户。此密码将在用户首次登录后过期。用户必须选择一个新密码。

添加和查看团队成员 (AWS CLI)

您可以使用 AWS CLI 向项目团队添加团队成员。您还可以查看有关项目中的所有团队成员的信息。

添加团队成员

1. 打开终端或命令窗口。
2. 运行带有 `--project-id`、`-user-arn` 和 `--project-role` 参数的 associate-team-member 命令。您还可通过包含 `--remote-access-allowed` 或 `--no-remote-access-allowed` 参数来指定用户是否具有对项目实例的远程访问权限。例如：

```
aws codestar associate-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/Jane_Doe --project-role Contributor --remote-access-
allowed
```

此命令不返回任何输出。

查看所有团队成员 (AWS CLI)

1. 打开终端或命令窗口。
2. 运行带有 `--project-id` 参数的 list-team-members 命令。例如：

```
aws codestar list-team-members --project-id my-first-projec
```

该命令会返回类似以下内容的输出：

```
{
    "teamMembers":[

 {"projectRole":"Owner","remoteAccessAllowed":true,"userArn":"arn:aws:iam::111111111111:use
Mary_Major"},

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::1111111111
Jane_Doe"},

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::1111111111
John_Doe"},

 {"projectRole":"Viewer","remoteAccessAllowed":false,"userArn":"arn:aws:iam::111111111111:u
John_Stiles"}
 ]
}
```

管理 AWS CodeStar 团队成员的权限

通过更改团队成员的 AWS CodeStar 角色来更改其权限。在 AWS CodeStar 项目中，每个团队成员只能分配给一个角色，而可将多个用户分配给同一角色。您可以使用 AWS CodeStar 控制台或 AWS CLI 管理权限。

> **Important**
>
> 要更改团队成员的角色，您必须具有该项目的 AWS CodeStar 所有者角色或已应用 `AWSCodeStarFullAccess` 策略。
> 更改团队成员的权限不影响此团队成员访问 AWS 以外的任何资源（例如，GitHub 存储库或 Atlassian JIRA 中的问题）。这些访问权限由资源提供者而非 AWS CodeStar 控制。有关更多信息，请参阅资源提供者的文档。

有权访问 AWS CodeStar 项目的任何人都能使用 AWS CodeStar 控制台访问位于 AWS 以外但与此项目有关的资源。
更改项目的团队成员角色不会自动允许或阻止该成员参与该项目的任何 AWS Cloud9 开发环境。要允许或阻止团队成员参与共享环境，请参阅与项目团队成员共享 AWS Cloud9 环境。

您也可以向用户授予远程访问与项目关联的任何 Amazon EC2 Linux 实例的权限。在您授予此权限后，用户必须上传与其在所有团队项目中的 AWS CodeStar 用户配置文件关联的 SSH 公有密钥。要成功连接到 Linux 实例，用户必须已在本地计算机上配置 SSH 并具有私有密钥。

主题

管理团队权限（控制台）

您可以使用 AWS CodeStar 控制台管理团队成员的角色。您还可以管理团队成员是否具有对与项目关联的 Amazon EC2 实例的远程访问权限。

更改团队成员的角色

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您的项目。
3. 在项目的侧导航栏中，选择团队。
4. 在团队成员页面上，选择团队成员，然后选择编辑。
5. 在项目角色中，选择要授予此用户的 AWS CodeStar 角色（所有者、贡献者或查看者）。

   有关 AWS CodeStar 角色及其权限的更多信息，请参阅与 AWS CodeStar 团队协作。

   选择编辑团队成员。

向团队成员授予对 Amazon EC2 实例的远程访问权限

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您的项目。
3. 在项目的侧导航栏中，选择团队。

4. 在团队成员页面上，选择团队成员，然后选择编辑。
5. 选中允许对项目实例进行 SSH 访问，然后选择编辑团队成员。
6. （可选）通知团队成员应为其 AWS CodeStar 用户上传 SSH 公有密钥（如果尚未这样做）。有关更多信息，请参阅将公有密钥添加到您的 AWS CodeStar 用户配置文件。

管理团队权限 (AWS CLI)

您可以使用 AWS CLI 管理分配给团队成员的项目角色。您可以使用相同的 AWS CLI 命令管理该团队成员是否具有对与项目关联的 Amazon EC2 实例的远程访问权限。

管理团队成员的权限

1. 打开终端或命令窗口。
2. 运行带有 `--project-id`、`-user-arn` 和 `--project-role` 参数的 update-team-member 命令。您还可通过包含 `--remote-access-allowed` 或 `--no-remote-access-allowed` 参数来指定用户是否具有对项目实例的远程访问权限。例如，更新名为 John_Doe 的 IAM 用户的项目角色，并将其权限更改为无法远程访问项目 Amazon EC2 实例的查看者：

```
aws codestar update-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/John_Doe --project-role Viewer --no-remote-access-
allowed
```

该命令会返回类似以下内容的输出：

```
{
 "projectRole":"Viewer",
 "remoteAccessAllowed":false,
 "userArn":"arn:aws:iam::111111111111:user/John_Doe"
}
```

从 AWS CodeStar 项目中删除团队成员

在从 AWS CodeStar 项目中删除一个用户后，该用户仍显示在项目存储库的提交历史记录中，但不再有权访问 CodeCommit 存储库或任何其他项目资源（例如，项目管道）。（此规则的例外情况是具有授予对这些资源的访问权限的其他策略的 IAM 用户。）该用户无法访问项目控制面板，并且项目不再显示在用户在 AWS CodeStar 控制面板上看到的项目列表中。您可以使用 AWS CodeStar 控制台或 AWS CLI 从项目团队中删除团队成员。

> **Important**
>
> 虽然从项目中删除团队成员会拒绝对项目 Amazon EC2 实例的远程访问，但不会关闭该用户的任何有效的 SSH 会话。
> 删除团队成员不影响此团队成员访问 AWS 以外的任何资源（例如，GitHub 存储库或 Atlassian JIRA 中的问题）。这些访问权限由资源提供者而非 AWS CodeStar 控制。有关更多信息，请参阅资源提供者的文档。
> 从项目中删除一个团队成员不会自动删除该团队成员的相关 AWS Cloud9 开发环境或阻止该成员受邀参与任何相关 AWS Cloud9 开发环境。要删除开发环境，请参阅从项目中删除 AWS Cloud9 环境。要阻止团队成员参与共享环境，请参阅与项目团队成员共享 AWS Cloud9 环境。

要从项目中删除团队成员，您必须具有该项目的 AWS CodeStar 所有者角色或已将 `AWSCodeStarFullAccess` 策略应用于您的账户。

主题

删除团队成员（控制台）

您可以使用 AWS CodeStar 控制台从项目团队中删除团队成员。

从项目中删除团队成员

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您的项目。
3. 在项目的侧导航栏中，选择团队。
4. 在团队成员页面上，选择团队成员，然后选择移除。

删除团队成员 (AWS CLI)

您可以使用 AWS CLI 从项目团队中删除团队成员。

删除团队成员

1. 打开终端或命令窗口。
2. 运行带 `--project-id` 和 `-user-arn` 的 disassociate-team-member 命令。例如：

```
aws codestar disassociate-team-member --project-id my-first-projec --user-arn
 arn:aws:iam:111111111111:user/John_Doe
```

该命令会返回类似以下内容的输出：

```
{
    "projectId": "my-first-projec",
    "userArn": "arn:aws:iam::111111111111:user/John_Doe"
}
```

使用您的 AWS CodeStar 用户配置文件

您的 AWS CodeStar 用户配置文件与您的 IAM 用户相关联。此配置文件包含在您所属的所有 AWS CodeStar 项目中使用的显示名称和电子邮件地址。您可以上传要与您的配置文件关联的 SSH 公有密钥。此公有密钥是在您连接到 Amazon EC2 实例（与您所属的 AWS CodeStar 项目关联）时所使用的 SSH 公有/私有密钥对的一部分。

> **Note**
>
> 这些主题中的信息仅涵盖您的 AWS CodeStar 用户配置文件。如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），这些资源提供者可能使用自己的用户配置文件，这些配置文件可能具有不同的设置。有关更多信息，请参阅资源提供者的文档。

主题

管理 AWS CodeStar 用户配置文件的显示信息

您可以使用 AWS CodeStar 控制台或 AWS CLI 来更改用户配置文件中的显示名称和电子邮件地址。用户配置文件不特定于项目。它与您的 IAM 用户相关联，应用于 AWS 区域中您属于的 AWS CodeStar 项目。如果您属于多个 AWS 区域中的项目，您将具有单独的用户配置文件。

您只能在 AWS CodeStar 控制台中管理自己的用户配置文件。如果您有 `AWSCodeStarFullAccess` 策略，则可使用 AWS CLI 查看和管理其他配置文件。

> **Note**
>
> 本主题中的信息仅涵盖您的 AWS CodeStar 用户配置文件。如果项目使用了 AWS 以外的资源（例如 GitHub 存储库或 Atlassian JIRA 中的问题），这些资源提供者可能使用自己的用户配置文件，这些配置文件可能具有不同的设置。有关更多信息，请参阅资源提供者的文档。

主题

管理您的用户配置文件（控制台）

通过导航到您作为团队成员的任何项目并更改您的配置文件信息，您可以在 AWS CodeStar 控制台中管理您的用户配置文件。由于用户配置文件是特定于用户而不是特定于项目，您的用户配置文件更改将显示在 AWS 区域中您作为团队成员的每个项目中。

> **Important**
>
> 要使用控制台更改用户的显示信息，您必须以该 IAM 用户的身份进行登录。所有其他用户（甚至是具有项目的 AWS CodeStar 所有者角色或已应用 `AWSCodeStarFullAccess` 策略的用户）均无法更改您的显示信息。

在一个 AWS 区域内的所有项目中更改您的显示信息

1. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。
2. 从导航窗格中选择项目，然后选择您作为团队成员所属于的项目。
3. 在项目的侧导航栏中，选择团队。
4. 在团队成员页面上，选择 IAM 用户，然后选择编辑。
5. 编辑显示名称和/或电子邮件地址，然后选择编辑团队成员。

> **Note**
>
> 显示名称和电子邮件地址都是必需的。有关更多信息，请参阅AWS CodeStar 中的限制。

管理用户配置文件 (AWS CLI)

在 AWS CodeStar 中，可使用 AWS CLI 创建和管理您的用户配置文件。您还可以使用 AWS CLI 查看您的用户配置文件信息，并查看为您在 AWS 区域中的 AWS 账户配置的所有用户配置文件。

确保已为您要在其中创建、管理或查看用户配置文件的区域配置您的 AWS 配置文件。

创建用户配置文件

1. 打开终端或命令窗口。

2. 运行带有 user-arn、display-name 和 email-address 参数的 create-user-profile 命令。例如：

```
aws codestar create-user-profile --user-arn arn:aws:iam:111111111111:user/
John_Stiles --display-name "John Stiles" --email-address "john_stiles@example.com"
```

该命令会返回类似以下内容的输出：

```
{
 "createdTimestamp":1.491439687681E9,"
 displayName":"John Stiles",
 "emailAddress":"john.stiles@example.com",
 "lastModifiedTimestamp":1.491439687681E9,
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

查看您的显示信息

1. 打开终端或命令窗口。
2. 运行带有 user-arn 参数的 describe-user-profile 命令。例如：

```
aws codestar describe-user-profile --user-arn arn:aws:iam:111111111111:user/
Mary_Major
```

该命令会返回类似以下内容的输出：

```
{
 "createdTimestamp":1.490634364532E9,
 "displayName":"Mary Major",
 "emailAddress":"mary.major@example.com",
 "lastModifiedTimestamp":1.491001935261E9,
 "sshPublicKey":"EXAMPLE=",
 "userArn":"arn:aws:iam::111111111111:user/Mary_Major"
}
```

更改您的显示信息

1. 打开终端或命令窗口。

2. 运行带 user-arn 参数和要更改的配置文件参数（例如 display-name 或 email-address）的 update-user-profile 命令。例如，如果显示名称为 Jane Doe 的用户希望将其显示名称更改为 Jane Mary Doe：

```
aws codestar update-user-profile --user-arn arn:aws:iam:111111111111:user/Jane_Doe
 --display-name "Jane Mary Doe"
```

该命令会返回类似以下内容的输出：

```
{
 "createdTimestamp":1.491439687681E9,
 "displayName":"Jane Mary Doe",
 "emailAddress":"jane.doe@example.com",
 "lastModifiedTimestamp":1.491442730598E9,
 "sshPublicKey":"EXAMPLE1",
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

列出您的 AWS 账户中 AWS 区域中的所有用户配置文件

1. 打开终端或命令窗口。
2. 运行 aws codestar list-user-profiles 命令。例如：

```
aws codestar list-user-profiles
```

该命令会返回类似以下内容的输出：

```
{
  "userProfiles":[
 {
  "displayName":"Jane Doe",
  "emailAddress":"jane.doe@example.com",
  "sshPublicKey":"EXAMPLE1",
  "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
 },
 {
  "displayName":"John Doe",
  "emailAddress":"john.doe@example.com",
  "sshPublicKey":"EXAMPLE2",
```

```
  "userArn":"arn:aws:iam::111111111111:user/John_Doe"
 },
 {
  "displayName":"Mary Major",
  "emailAddress":"mary.major@example.com",
  "sshPublicKey":"EXAMPLE=",
  "userArn":"arn:aws:iam::111111111111:user/Mary_Major"
 },
 {
  "displayName":"John Stiles",
  "emailAddress":"john.stiles@example.com",
  "sshPublicKey":"",
  "userArn":"arn:aws:iam::111111111111:user/John_Stiles"
 }
  ]
}
```

将公有密钥添加到您的 AWS CodeStar 用户配置文件

您可以上传公有 SSH 密钥作为您创建和管理的公有-私有密钥对的一部分。您使用此 SSH 公有-私有密钥对访问运行 Linux 的 Amazon EC2 实例。如果项目所有者已向您授予远程访问权限，那么您只能访问那些与项目关联的实例。您可以使用 AWS CodeStar 控制台或 AWS CLI 来管理公有密钥。

> **Important**
>
> AWS CodeStar 项目所有者可以授予项目所有者、贡献者和查看者对项目的 Amazon EC2 实例的 SSH 访问权限，但只有个人（所有者、贡献者或查看者）可以设置 SSH 密钥。要执行此操作，用户必须以个人所有者、贡献者或查看者身份登录。
>
> AWS CodeStar 不管理 AWS Cloud9 环境的 SSH 密钥。

主题

管理您的公有密钥（控制台）

尽管您无法在控制台中生成公有/私有密钥对，但您可以在本地创建一个，然后通过 AWS CodeStar 控制台将其作为用户配置文件的一部分进行添加或管理。

管理您的公有 SSH 密钥

1. 在终端或 Bash 仿真器窗口中，运行 ssh-keygen 命令，以在您的本地计算机上生成 SSH 公有-私有密钥对。您可以采用 Amazon EC2 允许的任何格式生成密钥。有关可接受的格式的信息，请参阅将您自己的公有密钥导入到 Amazon EC2。理想情况下，将生成一个密钥，它是一个采用 OpenSSH 格式的 SSH-2 RSA 密钥，并包含 2048 位。该公有密钥存储在扩展名为 .pub 的文件中。
2. 打开 AWS CodeStar 控制台，地址：https://console.aws.amazon.com/codestar/。

   选择一个您作为团队成员的项目。
3. 在导航窗格中，选择团队。
4. 在团队成员页面上，找到 IAM 用户的名称，然后选择编辑。
5. 在编辑团队成员页面的远程访问下，启用允许对项目实例进行 SSH 访问。
6. 在 SSH 公钥框中，粘贴公钥，然后选择编辑团队成员。

> **Note**
>
> 您可以通过删除此字段中的旧密钥然后粘贴新密钥来更改您的公有密钥。您可以通过删除此字段的内容然后选择编辑团队成员来删除公有密钥。

当您更改或删除公有密钥时，您也在更改用户配置文件。这不是逐个项目的更改。由于您的密钥与配置文件关联，在您已获得远程访问权限的所有项目中，该密钥都将更改（或被删除）。

删除您的公有密钥将在您已获得远程访问权限的所有项目中删除您对运行 Linux 的 Amazon EC2 实例的访问权限。但是，这不会关闭使用该密钥的任何打开的 SSH 会话。请确保关闭所有打开的会话。

管理您的公有密钥 (AWS CLI)

您可以使用 AWS CLI 将 SSH 公有密钥作为用户配置文件的一部分进行管理。

管理您的公有密钥

1. 在终端或 Bash 仿真器窗口中，运行 ssh-keygen 命令，以在您的本地计算机上生成 SSH 公有-私有密钥对。您可以采用 Amazon EC2 允许的任何格式生成密钥。有关可接受的格式的信息，请参阅将您自己的公有密钥导入到 Amazon EC2。理想情况下，将生成一个密钥，它是一个采用 OpenSSH 格式的 SSH-2 RSA 密钥，并包含 2048 位。该公有密钥存储在扩展名为 .pub 的文件中。
2. 要在您的 AWS CodeStar 用户配置文件中添加或更改 SSH 公有密钥，请运行包含 `--ssh-public-key` 参数的 update-user-profile 命令。例如：

```
aws codestar update-user-profile --user-arn arn:aws:iam:111111111111:user/Jane_Doe
 --ssh-key-id EXAMPLE1
```

该命令会返回类似以下内容的输出：

```
{
 "createdTimestamp":1.491439687681E9,
 "displayName":"Jane Doe",
 "emailAddress":"jane.doe@example.com",
 "lastModifiedTimestamp":1.491442730598E9,
 "sshPublicKey":"EXAMPLE1",
 "userArn":"arn:aws:iam::111111111111:user/Jane_Doe"
}
```

使用私有密钥连接到 Amazon EC2 实例

确保您已创建 Amazon EC2 密钥对。将公有密钥添加到 AWS CodeStar 中的用户配置文件。要创建密钥对，请参阅步骤 4：为 AWS CodeStar 项目创建 Amazon EC2 密钥对。要将公有密钥添加到您的用户配置文件，请参阅本主题前面的说明。

使用您的私有密钥连接到 Amazon EC2 Linux 实例

1. 在您的项目在 AWS CodeStar 控制台中处于打开状态的情况下，在导航窗格中选择项目。
2. 在项目资源中，对于类型为 Amazon EC2 且名称以 instance 开头的行，选择 ARN 链接。
3. 在 Amazon EC2 控制台，选择连接。
4. 按照连接到您的实例对话框中的说明操作。

   用户名称使用 `ec2-user`。如果您使用了错误的用户名，则无法连接到实例。

有关更多信息，请参阅适用于 Linux 实例的 Amazon EC2 用户指南中的以下资源：

- 使用 SSH 连接到 Linux 实例
- 使用 PuTTY 从 Windows 连接到 Linux 实例
- 使用 MindTerm 连接到 Linux 实例

AWS CodeStar 中的安全性

AWS 十分重视云安全性。作为 AWS 客户，您将从专为满足大多数安全敏感型企业的要求而打造的数据中心和网络架构中受益。

安全性是AWS和您的共同责任。责任共担模型将其描述为云的安全性和云中的安全性：

- 云的安全性 – AWS 负责保护在 AWS 云中运行 AWS 服务的基础设施。AWS 还向您提供可安全使用的服务。作为 AWS 合规性计划的一部分，第三方审计人员将定期测试和验证安全措施的有效性。要了解适用于 AWS CodeStar 的合规性计划，请参阅合规性计划范围内的亚马逊云科技服务。
- 云中的安全性 - 您的责任由您使用的 AWS 服务决定。您还需要对其它因素负责，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。

此文档将帮助您了解如何在使用 AWS CodeStar 时应用责任共担模型 以下主题说明如何配置 AWS CodeStar 以实现您的安全性和合规性目标。您还会了解如何使用其它 AWS 服务以帮助您监控和保护 AWS CodeStar 资源。

在 AWS CodeStar 中创建自定义策略并使用权限边界时，请仅授予执行任务所需的权限，并将权限范围缩小到目标资源，从而确保最低权限的访问权限。为防止其他项目的成员访问您的项目中的资源，请为组织成员授予每个 AWS CodeStar 项目的单独权限。最佳做法是为每个成员创建一个项目账户，然后为该账户分配基于角色的访问权限。

例如，您可以使用诸如 AWS Control Tower with AWS Organizations 之类的服务为 DevOps 组下的每个开发者角色配置账户。然后，您可以为这些账户分配权限。总体权限适用于账户，但用户对项目外部资源的访问权限有限。

有关使用多账户策略管理 AWS 资源最低权限访问权限的更多信息，请参阅 AWS Control Tower 用户指南中的登录区 AWS 多账户策略。

主题

AWS CodeStar 中的数据保护

AWS 责任共担模式适用于 AWS CodeStar 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 上的博客文章 AWS Shared Responsibility Model and GDPR。

出于数据保护目的，我们建议您保护 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 和用户活动日志记录。
- 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。
- 如果您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（如名称字段）。这包括处理 CodeStar 或其他 AWS 服务 时使用控制台、API、AWS CLI 或 AWS 开发工具包。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志。如果您向外部服务器提供网址，我们强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

AWS CodeStar 中的数据加密

默认情况下，AWS CodeStar 会对其存储的有关项目的信息进行加密。除了项目 ID 之外的所有内容都会进行静态加密，例如，项目名称、描述和用户电子邮件。避免将个人信息放入您的项目 ID 中。默认情况下，AWS CodeStar 还会加密传输中的信息。静态加密或传输中加密均不需要客户操作。

适用于 AWS CodeStar 的身份和访问权限管理

AWS Identity and Access Management (IAM) 是一项 AWS 服务，可以帮助管理员安全地控制对 AWS 资源的访问。IAM 管理员控制可以通过身份验证（登录）和获得授权（具有权限）来使用 AWS CodeStar 资源的人员。IAM 是一项无需额外费用即可使用的 AWS 服务。

主题

受众

使用 AWS Identity and Access Management (IAM) 的方式因您可以在 AWS CodeStar 中执行的操作而异。

服务用户 – 如果使用 AWS CodeStar 服务来完成任务，则您的管理员会为您提供所需的凭证和权限。当您使用更多 AWS CodeStar 功能来完成工作时，您可能需要更多权限。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 中的功能，请参阅 排查 AWS CodeStar 身份和访问的问题。

服务管理员：如果您在公司负责管理 AWS CodeStar 资源，则您可能具有 AWS CodeStar 的完全访问权限。您有责任确定您的服务用户应访问哪些 AWS CodeStar 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。要了解有关您的公司如何将 IAM 与 AWS CodeStar 搭配使用的更多信息，请参阅 AWS CodeStar 如何与 IAM 配合使用。

IAM 管理员：如果您是 IAM 管理员，您可能希望了解有关如何编写策略以管理对 AWS CodeStar 的访问权限的详细信息。要查看您可在 IAM 中使用的 AWS CodeStar 基于身份的策略示例，请参阅 AWS CodeStar 基于身份的策略示例。

使用身份进行身份验证

身份验证是使用身份凭证登录 AWS 的方法。您必须作为 AWS 账户根用户、IAM 用户或通过担任 IAM 角色进行身份验证（登录到 AWS）。

您可以使用通过身份源提供的凭证以联合身份登录到 AWS。AWS IAM Identity Center(IAM Identity Center) 用户、您公司的单点登录身份验证以及您的 Google 或 Facebook 凭证都是联合身份的示例。

当您以联合身份登录时，管理员以前使用 IAM 角色设置了身份联合验证。当您使用联合身份验证访问 AWS 时，您就是在间接代入角色。

根据用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录到 AWS 的更多信息，请参阅《AWS 登录 用户指南》中的如何登录到您的 AWS 账户。

如果您以编程方式访问 AWS，则 AWS 将提供软件开发工具包 (SDK) 和命令行界面 (CLI)，以便使用您的凭证以加密方式签署您的请求。如果您不使用 AWS 工具，则必须自行对请求签名。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，您都可能需要提供其它安全信息。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 AWS 账户 时，最初使用的是一个对账户中所有 AWS 服务 和资源拥有完全访问权限的登录身份。此身份称为 AWS 账户根用户，使用您创建账户时所用的电子邮件地址和密码登录，即可获得该身份。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 用户和组

IAM 用户是 AWS 账户 内对某个人员或应用程序具有特定权限的一个身份。在可能的情况下，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，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。您不能使用群组的身份登录。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，使用群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担任。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

IAM 角色是 AWS 账户 中具有特定权限的身份。它类似于 IAM 用户，但与特定人员不关联。您可以通过切换角色，在 AWS Management Console 中暂时担任 IAM 角色。您可以调用 AWS CLI 或 AWS

API 操作或使用自定义网址以代入角色。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 用户或角色可代入 IAM 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。角色是授予跨账户存取权限的主要方式。但是，对于某些 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座席）。要了解用于跨账户存取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 – 某些 AWS 服务 使用其他 AWS 服务 中的特征。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：当您使用 IAM 用户或角色在 AWS 中执行操作时，您将被视为主体。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 使用主体调用 AWS 服务 的权限，结合请求的 AWS 服务，向下游服务发出请求。只有在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 请求时的政策详情，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担任的 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 - 服务相关角色是与 AWS 服务关联的一种服务角色。服务可以担任代表您执行操作的角色。服务相关角色显示在您的 AWS 账户中，并由该服务拥有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 Amazon 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凭证。这优先于在 EC2 实例中存储访问密钥。要将 AWS 角色分配给 EC2 实例并使其对该实例的所有应用程序可用，您可以创建一个附加到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

您将创建策略并将其附加到 AWS 身份或资源，以控制 AWS 中的访问。策略是 AWS 中的对象；在与身份或资源相关联时，策略定义它们的权限。在主体（用户、根用户或角色会话）发出请求时，AWS 将评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略在 AWS 中存储为 JSON 文档。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内容。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。然后，管理员可以向角色添加 IAM policy，并且用户可以代入角色。

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，假设有一个允许 `iam:GetRole` 操作的策略。具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策略。内联策略直接嵌入单个用户、群组或角色中。托管策略是可以附加到AWS 账户中的多个用户、组和角色的独立策略。托管式策略包括 AWS 托管式策略和客户管理型策略。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。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。主体可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用来自 IAM 的 AWS 托管策略。

访问控制列表（ACL）

访问控制列表 (ACL) 控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3、AWS WAF 和 Amazon VPC 是支持 ACL 的服务示例。要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》 中的访问控制列表 (ACL) 概览。

其他策略类型

AWS 支持额外的、不太常用的策略类型。这些策略类型可以设置更常用的策略类型授予的最大权限。

- 权限边界 - 权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可以为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 `Principal` 字段中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCP) – SCP 是 JSON 策略，指定了组织或组织单位 (OU) 在 AWS Organizations 中的最大权限。AWS Organizations 服务可以分组和集中管理您的企业拥有的多个 AWS 账户 账户。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中实体（包括每个 AWS 账户根用户）的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的 SCP 的工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解 AWS 如何确定在涉及多种策略类型时是否允许请求，请参阅《IAM 用户指南》中的策略评估逻辑。

AWS CodeStar 如何与 IAM 配合使用

在使用 IAM 管理对 AWS CodeStar 的访问之前，您应了解哪些 IAM 功能可与 AWS CodeStar 结合使用。要大致了解 AWS CodeStar 和其他 AWS 服务如何与 IAM 一起使用，请参阅IAM 用户指南中的与 IAM 配合使用的 AWS 服务。

主题

- AWS CodeStar 基于身份的策略
- AWS CodeStar 基于资源的策略
- 基于 AWS CodeStar 标签的授权
- AWS CodeStar IAM 角色
- IAM 用户对 AWS CodeStar 的访问权限
- 联合身份用户对 AWS CodeStar 的访问权限
- 将临时凭证用于 AWS CodeStar
- 服务相关角色
- 服务角色

AWS CodeStar 基于身份的策略

通过 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 会代表您创建多个基于身份的策略，这些策略允许 AWS CodeStar 在 AWS CodeStar 项目范围内创建和管理资源。AWS CodeStar 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内容。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。策略操作通常与相关联的 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。

AWS CodeStar 中的策略操作在操作前使用以下前缀：`codestar:`。例如，要允许指定的 IAM 用户编辑 AWS CodeStar 项目的属性（如其项目描述），您可以使用以下策略声明：

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
```

```
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

策略语句必须包含 Action 或 NotAction 元素。AWS CodeStar 定义了一组自己的操作，以描述您可以使用该服务执行的任务。

要在单个语句中指定多项操作，请使用逗号将它们隔开，如下所示：

```
"Action": [
      "codestar:action1",
      "codestar: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，包括以下操作：

```
"Action": "codestar:List*"
```

要查看 AWS CodeStar 操作列表，请参阅 IAM 用户指南中的 AWS CodeStar 定义的操作。

资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么内容。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。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 Amazon 资源名称 (ARN) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*)指示语句应用于所有资源。

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

AWS CodeStar 项目资源具有以下 ARN：

```
arn:aws:codestar:region:account:project/resource-specifier
```

有关 ARN 格式的更多信息，请参阅 Amazon 资源名称 (ARN) 和 AWS 服务命名空间。

例如，以下内容指定已注册到 AWS 区域 us-east-2 中 AWS 账户 111111111111 的名为 *my-first-projec* 的 AWS CodeStar 项目：

```
arn:aws:codestar:us-east-2:111111111111:project/my-first-projec
```

以下内容指定已注册到 AWS 区域 us-east-2 中 AWS 账户 111111111111 的以名称 my-proj 开头的任何 AWS CodeStar 项目：

```
arn:aws:codestar:us-east-2:111111111111:project/my-proj*
```

某些 AWS CodeStar 操作（例如，列出项目）不能在资源上执行。在这些情况下，您必须使用通配符 (*)。

```
"LisProjects": "*"
```

要查看 AWS CodeStar 资源类型及其 ARN 的列表，请参阅 IAM 用户指南中的 AWS CodeStar 定义的资源。要了解您可以在哪些操作中指定每个资源的 ARN，请参阅 AWS CodeStar 定义的操作。

条件键

AWS CodeStar 不提供任何特定于服务的条件键，但支持使用某些全局条件键。要查看所有 AWS 全局条件键，请参阅《IAM 用户指南》中的 AWS 全局条件上下文键。

示例

要查看 AWS CodeStar 基于身份的策略示例，请参阅 AWS CodeStar 基于身份的策略示例。

AWS CodeStar 基于资源的策略

AWS CodeStar 不支持基于资源的策略。

基于 AWS CodeStar 标签的授权

您可以将标签附加到 AWS CodeStar 项目，或在对 AWS CodeStar 的请求中传递标签。要基于标签控制访问，需要使用 codestar:ResourceTag/*key-name*、aws:RequestTag/*key-name* 或 aws:TagKeys 条件密钥在策略的条件元素中提供标签信息。有关为 AWS CodeStar 资源添加标签的更多信息，请参阅 the section called “使用项目标签”。

要查看基于身份的策略示例，了解如何基于 AWS CodeStar 项目上的标签限制对该项目的访问权限，请参阅基于标签查看 AWS CodeStar 项目。

AWS CodeStar IAM 角色

IAM 角色是您的 AWS 账户中一个具有特定权限的实体。

您可以将 AWS CodeStar 作为 IAM 用户、联合用户、根用户或代入角色使用。所有具有适当权限的用户类型都可以管理其 AWS 资源的项目权限，但 AWS CodeStar 会自动管理 IAM 用户的项目权限。IAM 策略和角色基于项目角色向该用户授予权限和访问权限。您可以使用 IAM 控制台创建其他策略来向 IAM 用户分配 AWS CodeStar 和其他权限。

例如，您可能希望允许用户查看 AWS CodeStar 项目但不能执行更改。在这种情况下，向 AWS CodeStar 项目添加具有查看者角色的 IAM 用户。每个 AWS CodeStar 项目都有一组策略来帮助您控制对项目的访问。此外，您可以控制哪些用户有权访问 AWS CodeStar。

为 IAM 用户和联合身份用户处理 AWS CodeStar 访问权限的方式不同。仅 IAM 用户可添加到团队中。要为 IAM 用户授予对项目的权限，您可以将用户添加到项目团队并为用户分配角色。要为联合身份用户授予对项目的权限，您可以手动将 AWS CodeStar 项目角色的托管策略附加到联合身份用户的角色。

下表总结了可用于每种类型的访问的工具。

权限功能	IAM 用户	联合用户	根用户
用于远程访问 Amazon EC2 和 Elastic Beanstalk 项目的 SSH 密钥管理	✓		
AWS CodeCommitSSH 访问	✓		
AWS CodeStar 管理的 IAM 用户权限	✓		
手动管理的项目权限		✓	✓
用户可作为团队成员添加到项目中	✓		

IAM 用户对 AWS CodeStar 的访问权限

当您向项目添加 IAM 用户并为此用户选择角色时，AWS CodeStar 将自动向此 IAM 用户应用适当的策略。对于 IAM 用户，您不需要在 IAM 中直接附加或管理策略或权限。有关向 AWS CodeStar 项目添加

IAM 用户的信息，请参阅向 AWS CodeStar 项目添加团队成员 。有关从 AWS CodeStar 项目中删除 IAM 用户的信息，请参阅从 AWS CodeStar 项目中删除团队成员 。

将内联策略附加到 IAM 用户

当您向项目添加用户时，AWS CodeStar 将自动为项目附加匹配用户角色的托管策略。您不应将项目的 AWS CodeStar 托管策略手动附加到 IAM 用户。建议您不要附加会更改 AWS CodeStar 项目中的 IAM 用户权限的策略（`AWSCodeStarFullAccess` 除外）。如果您决定创建和附加自己的策略，请参阅 IAM 用户指南中的添加和删除 IAM 身份权限。

联合身份用户对 AWS CodeStar 的访问权限

您可以不创建 IAM 用户或使用根用户，而是使用来自 AWS Directory Service、您的企业用户目录、Web 身份提供商或代入角色的 IAM 用户的用户身份。他们被称为联合身份用户。

授予联合身份用户对您的 AWS CodeStar 项目的访问权限，方法是将 AWS CodeStar 项目级别的策略和权限中所述的托管策略手动附加到用户的 IAM 角色。在 AWS CodeStar 创建您的项目资源和 IAM 角色之后附加所有者、贡献者或查看者策略。

先决条件：

- 您必须已设置身份提供商。例如，您可以设置 SAML 身份提供商并通过该提供商设置 AWS 身份验证。有关设置身份提供商的更多信息，请参阅创建 IAM 身份提供商。有关 SAML 联合身份验证的更多信息，请参阅关于基于 SAML 2.0 的联合身份验证。
- 在通过身份提供商请求访问权限时，您必须已创建联合身份用户要代入的角色。STS 信任策略必须附加到允许联合身份用户代入角色的角色。有关更多信息，请参阅 IAM 用户指南中的联合身份用户和角色。
- 您必须已创建 AWS CodeStar 项目并知道项目 ID。

有关针对身份提供商创建角色的更多信息，请参阅针对第三方身份提供商创建角色（联合身份验证）。

将 AWSCodeStarFullAccess 托管策略附加到联合身份用户的角色

通过附加 `AWSCodeStarFullAccess` 托管策略来授予联合身份用户创建项目的权限。要执行这些步骤，您必须已作为根用户、账户中的 IAM 管理员用户或者具有关联的 `AdministratorAccess` 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

> **Note**
>
> 在创建项目后，不会自动应用您的项目所有者权限。按照将您的项目的 AWS CodeStar 查看者/贡献者/所有者托管策略附加到联合身份用户的角色中所述，使用对您的账户具有管理权限的角色，附加所有者托管策略。

1. 打开 IAM 控制台。在导航窗格中，选择策略。
2. 在搜索字段中输入 AWSCodeStarFullAccess。此时将显示该策略名称，并且策略类型为 AWS 托管。您可以展开该策略以查看策略声明中的权限。
3. 选择策略旁边的圆圈，然后在策略操作下选择附加。
4. 在摘要页面上，选择关联实体选项卡。选择附加。
5. 在附加策略页面上的搜索字段中，筛选联合身份用户的角色。选中角色名称旁边的框，然后选择附加策略。附加的实体 选项卡将显示新附加的实体。

将您的项目的 AWS CodeStar 查看者/贡献者/所有者托管策略附加到联合身份用户的角色

授予联合身份用户对您的项目的访问权限，方法是将适当的所有者、贡献者或查看者托管策略附加到用户的角色。托管策略将授予适当级别的权限。与 IAM 用户不同，您必须为联合身份用户手动附加和分离托管策略。这等效于向 AWS CodeStar 中的团队成员分配项目权限。要执行这些步骤，您必须已作为根用户、账户中的 IAM 管理员用户或者具有关联的 AdministratorAccess 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

先决条件：

- 您必须已创建角色或具有联合身份用户代入的现有角色。
- 您必须知道要授予的权限级别。附加到所有者、贡献者和查看者角色的托管策略提供对您的项目的基于角色的权限。
- 您的 AWS CodeStar 项目必须已创建。在创建项目之前，托管策略在 IAM 中不可用。

1. 打开 IAM 控制台。在导航窗格中，选择策略。
2. 在搜索字段中输入您的项目 ID。此时将显示与您的项目匹配的策略名称，并且策略类型为客户托管。您可以展开该策略以查看策略声明中的权限。
3. 选择其中一个托管策略。选择策略旁边的圆圈，然后在策略操作下选择附加。
4. 在摘要页面上，选择关联实体选项卡。选择附加。

5. 在附加策略页面上的搜索字段中，筛选联合身份用户的角色。选中角色名称旁边的框，然后选择附加策略。附加的实体 选项卡将显示新附加的实体。

从联合身份用户的角色分离 AWS CodeStar 托管策略

在删除您的 AWS CodeStar 项目之前，您必须手动分离附加到联合身份用户的角色的任何托管策略。要执行这些步骤，您必须已作为根用户、账户中的 IAM 管理员用户或者具有关联的 AdministratorAccess 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

1. 打开 IAM 控制台。在导航窗格中，选择策略。
2. 在搜索字段中输入您的项目 ID。
3. 选择策略旁边的圆圈，然后在策略操作下选择附加。
4. 在摘要页面上，选择关联实体选项卡。
5. 在搜索字段中，筛选联合身份用户的角色。选择分离。

将 AWS Cloud9 托管策略附加到联合身份用户的角色

如果使用的是 AWS Cloud9 开发环境，请通过将 AWSCloud9User 托管策略附加到用户角色来授予联合身份用户对该环境的访问权限。与 IAM 用户不同，您必须为联合身份用户手动附加和分离托管策略。要执行这些步骤，您必须已作为根用户、账户中的 IAM 管理员用户或者具有关联的 AdministratorAccess 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

先决条件：

- 您必须已创建角色或具有联合身份用户代入的现有角色。
- 您必须知道要授予的权限级别：
  - AWSCloud9User 托管策略允许用户执行以下操作：
    - 创建自己的 AWS Cloud9 开发环境。
    - 获取有关环境的信息。
    - 更改环境的设置。
  - AWSCloud9Administrator 托管策略允许用户为自己或他人执行以下操作：
    - 创建环境。
    - 获取有关环境的信息。
    - 删除环境。
    - 更改环境的设置。

1. 打开 IAM 控制台。在导航窗格中，选择策略。
2. 在搜索字段中输入策略名称。此时将显示此托管策略，并且策略类型为 AWS 托管。您可以展开该策略以查看策略声明中的权限。
3. 选择其中一个托管策略。选择策略旁边的圆圈，然后在策略操作下选择附加。
4. 在摘要页面上，选择关联实体选项卡。选择附加。
5. 在附加策略页面上的搜索字段中，筛选联合身份用户的角色。选中角色名称旁边的框，然后选择附加策略。附加的实体 选项卡将显示新附加的实体。

从联合身份用户的角色分离 AWS Cloud9 托管策略

如果使用的是 AWS Cloud9 开发环境，则可以通过分离授予访问权限的策略来删除联合身份用户对该环境的访问权限。要执行这些步骤，您必须已作为根用户、账户中的 IAM 管理员用户或者具有关联的 `AdministratorAccess` 托管策略或等效策略的 IAM 用户或联合用户登录到控制台。

1. 打开 IAM 控制台。在导航窗格中，选择策略。
2. 在搜索字段中输入您的项目名称。
3. 选择策略旁边的圆圈，然后在策略操作下选择附加。
4. 在摘要页面上，选择关联实体选项卡。
5. 在搜索字段中，筛选联合身份用户的角色。选择分离。

将临时凭证用于 AWS CodeStar

可以使用临时凭证进行联合身份验证登录，代入 IAM 角色或代入跨账户角色。可以通过调用AWS STS API 操作（如 AssumeRole 或 GetFederationToken）获得临时安全凭证。

AWS CodeStar 支持使用临时凭证，但 AWS CodeStar 团队成员功能不适用于联合访问。AWS CodeStar 团队成员功能仅支持将 IAM 用户添加为团队成员。

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。 管理员可以查看，但不能编辑服务相关角色的权限。

AWS CodeStar 不支持服务相关角色。

服务角色

此功能允许服务代表您代入服务角色。此角色允许服务访问其他服务中的资源以代表您完成操作。服务角色显示在 IAM 账户中，并归该账户所有。这意味着管理员可以更改此角色的权限。但是，这样做可能会中断服务的功能。

AWS CodeStar 支持服务角色。AWS CodeStar 在为您的项目创建和管理资源时，使用服务角色 aws-codestar-service-role。有关更多信息，请参阅 IAM 用户指南中的角色术语和概念。

> **Important**
>
> 您必须以 管理员用户或根账户身份登录才能创建此服务角色。有关更多信息，请参阅IAM 用户指南中的仅限首次访问：您的根用户凭证以及创建您的第一个管理员用户和组。

您首次在 AWS CodeStar 中创建项目时会为您创建此角色。服务角色代表您执行以下操作：

- 创建您在创建项目时选择的资源。
- 在 AWS CodeStar 项目控制面板中显示有关这些资源的信息。

此外，它还在您管理项目资源时代表您执行操作。有关此策略声明的示例，请参阅 AWSCodeStarServiceRole 策略。

此外，AWS CodeStar 还会根据项目类型，创建多个特定于项目的服务角色。会为每个项目类型创建 AWS CloudFormation 角色和工具链角色。

- AWS CloudFormation 角色允许 AWS CodeStar 访问 AWS CloudFormation，以便为您的 AWS CodeStar 项目创建和修改堆栈。
- 工具链角色允许 AWS CodeStar 访问其他 AWS 服务，以创建和修改 AWS CodeStar 项目的资源。

AWS CodeStar 项目级别的策略和权限

在创建项目时，AWS CodeStar 会创建管理您的项目资源所需的 IAM 角色和策略。该策略分为三个类别：

- 面向项目团队成员的 IAM 策略。
- 面向工作线程角色的 IAM 策略。
- 面向运行时执行角色的 IAM 策略。

面向团队成员的 IAM 策略

在创建项目时，AWS CodeStar 将创建三个客户托管策略，分别用于所有者、贡献者和查看者对项目的访问。所有 AWS CodeStar 项目都包含针对这三个访问级别的 IAM 策略。这些访问级别是项目特定的，并由 IAM 托管策略使用标准名称定义，其中 *project-id* 是 AWS CodeStar 项目的 ID（例如，*my-first-projec*）：

- CodeStar_*project-id*_Owner
- CodeStar_*project-id*_Contributor
- CodeStar_*project-id*_Viewer

> **Important**
>
> AWS CodeStar 可能随时更改这些策略。不应手动编辑它们。如果您要添加或更改权限，请将其他策略附加到 IAM 用户。

在将团队成员（IAM 用户）添加到项目并选择其访问级别时，相应的策略将附加到 IAM 用户，并向用户授予一组适当的权限来操作项目资源。大多数情况下，您不需要在 IAM 中直接附加或管理策略或权限。建议不要将 AWS CodeStar 访问级别策略手动附加到 IAM 用户。如果绝对必要，作为 AWS CodeStar 访问级别策略的补充，您可以创建自己的托管策略或内联策略以将您自身级别的权限应用于 IAM 用户。

这些策略的适用范围限定于项目资源和特定操作。在将新资源添加到基础设施堆栈时，AWS CodeStar 会尝试更新团队成员策略以包含对新资源的访问权限（如果它们是受支持的资源类型之一）。

> **Note**
>
> AWS CodeStar 项目中的访问级别策略仅适用于该项目。这可帮助确保用户只能看到其角色所确定的级别有权看到的 AWS CodeStar 项目并与之交互。仅可以为创建 AWS CodeStar 项目的用户应用允许访问所有 AWS CodeStar 资源（不管项目如何）的策略。

AWS CodeStar 访问级别策略各不相同，具体取决于与访问级别所关联的项目相关联的 AWS 资源。与其他 AWS 服务不同，如果在项目资源发生更改时创建和更新项目，则将自定义这些策略。因此，没有规范的所有者、贡献者或查看者托管策略。

AWS CodeStar 所有者角色策略

CodeStar_*project-id*_Owner 客户托管策略允许用户在 AWS CodeStar 项目中不受限制地执行所有操作。这是允许用户添加或删除团队成员的唯一策略。该策略的内容因与项目关联的资源而异。有关示例，请参阅 AWSCodeStar 所有者角色策略。

具有此策略的 IAM 用户可以在项目中执行所有 AWS CodeStar 操作，但与具有 AWSCodeStarFullAccess 策略的 IAM 用户不同，前者无法创建项目。codestar:* 权限的范围限制为特定资源（与项目 ID 关联的 AWS CodeStar 项目）。

AWS CodeStar 贡献者角色策略

CodeStar_*project-id*_Contributor 客户托管策略允许用户参与项目和更改项目控制面板，但不允许用户添加或删除团队成员。该策略的内容因与项目关联的资源而异。有关示例，请参阅 AWS CodeStar 贡献者角色策略。

AWS CodeStar 查看者角色策略

CodeStar_*project-id*_Viewer 客户托管策略允许用户在 AWS CodeStar 中查看项目，但不允许更改其资源，也不允许添加或删除团队成员。该策略的内容因与项目关联的资源而异。有关示例，请参阅 AWS CodeStar 查看者角色策略 。

面向工作线程角色的 IAM 策略

如果您在太平洋夏令时 2018 年 12 月 6 日后创建 AWS CodeStar 项目，则 AWS CodeStar 会创建两个工作线程角色，即 CodeStar-*project-id*-ToolChain 和 CodeStar-*project-id*-CloudFormation。工作线程角色是 AWS CodeStar 创建的项目特定的 IAM 角色，此角色将传递给服务。它授予权限，以便服务能够创建资源并在 AWS CodeStar 项目的上下文中执行操作。工具链工作线程角色与工具链服务（例如 CodeBuild、CodeDeploy 和 CodePipeline）之间建立了信任关系。向项目团队成员（所有者和贡献者）授予了访问权限以将工作线程角色传递到可信下游服务。有关此角色的内联策略声明的示例，请参阅 AWS CodeStar 工具链工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之后）。

CloudFormation 工作线程角色包含 AWS CloudFormation 支持的选定资源的权限，以及在应用程序堆栈中创建 IAM 用户、角色和策略所需的权限。它还与 AWS CloudFormation 建立了信任关系。为了降低权限升级和破坏性操作的风险，AWS CloudFormation 角色策略包含了一个条件，该条件要求针对基础设施堆栈中创建的每个 IAM 实体（用户或角色）设置项目特定的权限边界。有关此角色的内联策略声明的示例，请参阅 AWS CloudFormation 工作线程角色策略。

对于太平洋夏令时 2018 年 12 月 6 日之前创建的 AWS CodeStar 项目，AWS CodeStar 会为工具链资源（例如 CodePipeline、CodeBuild 和 CloudWatch 事件）分别创建单独的工作线程角色，还会为

AWS CloudFormation 创建一个工作线程角色来支持一组有限的资源。其中的每个角色均与相应服务建立了信任关系。向项目团队成员（所有者和贡献者）以及其他一些工作线程角色授予访问权限以将该角色传递到可信下游服务。工作线程角色的权限是在一个内联策略中定义的，该内联策略的适用范围限定于角色可对一组项目资源执行的一组基本操作。这些权限是静态的。它们适用于创建项目时包含在项目中的资源，但不会在项目中添加了新资源时进行更新。有关这些策略声明的示例，请参阅：

- AWS CloudFormation 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）
- AWS CodePipeline 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）
- AWS CodeBuild 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）
- Amazon CloudWatch Events 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）

面向执行角色的 IAM 策略

对于太平洋夏令时 2018 年 12 月 6 日之后创建的项目，AWS CodeStar 会为应用程序堆栈中的示例项目创建一般执行角色。该角色使用权限边界策略将适用范围限定于相应的项目资源。在扩展示例项目时，您将创建额外的 IAM 角色，并且 AWS CloudFormation 角色策略要求使用权限边界限定这些角色的范围，以避免权限升级。有关更多信息，请参阅向项目添加 IAM 角色。

对于太平洋夏令时 2018 年 12 月 6 日之前创建的 Lambda 项目，AWS CodeStar 会创建一个 Lambda 执行角色，该角色附加了一个内联策略，定义了对项目 AWS SAM 堆栈中的资源进行操作所需的权限。在将新资源添加到 SAM 模板时，AWS CodeStar 会尝试更新 Lambda 执行角色策略以包含对新资源的权限（如果它们是受支持的资源类型之一）。

IAM 权限边界

太平洋夏令时 2018 年 12 月 6 日之后，当您创建项目时，AWS CodeStar 会创建一个客户管理型策略，并将该策略作为 IAM 权限边界分配给项目中的 IAM 角色。AWS CodeStar 要求在应用程序堆栈中创建的所有 IAM 实体都有权限边界。权限边界控制角色可具有的最大权限，但不向角色提供任何权限。权限策略定义角色的权限。这意味着，无论向角色添加多少额外权限，使用该角色的任何人都无法执行权限边界中包含的操作之外的任何操作。有关如何评估权限策略和权限边界的信息，请参阅 IAM 用户指南中的策略评估逻辑。

AWS CodeStar 使用项目特定的权限边界来防止权限升级到项目之外的资源。AWS CodeStar 权限边界包含项目资源的 ARN。有关此策略声明的示例，请参阅AWS CodeStar 权限边界策略。

在通过应用程序堆栈 (`template.yml`) 在项目中添加或从中删除受支持的资源时，AWS CodeStar 转换会更新此策略。

将 IAM 权限边界添加到现有项目

如果您具有太平洋夏令时 2018 年 12 月 6 日之前创建的 AWS CodeStar 项目，您应该手动将权限边界添加到项目中的 IAM 角色。作为最佳实践，建议您使用项目特定的边界（仅包含项目中的资源）以避免权限提升到项目之外的资源。按照以下步骤使用随着项目发展而更新的 AWS CodeStar 托管权限边界。

1. 登录 AWS CloudFormation 控制台并找到针对项目中的工具链堆栈的模板。此模板名为 `awscodestar-project-id`。
2. 选择模板，选择操作，然后选择在 Designer 中查看/编辑模板。
3. 找到 `Resources` 部分，并在此部分的顶部包含以下代码段。

```
 PermissionsBoundaryPolicy:
   Description: Creating an IAM managed policy for defining the permissions boundary
for an AWS CodeStar project
   Type: AWS::IAM::ManagedPolicy
   Properties:
     ManagedPolicyName: !Sub 'CodeStar_${ProjectId }_PermissionsBoundary'
     Description: 'IAM policy to define the permissions boundary for IAM entities
created in an AWS CodeStar project'
     PolicyDocument:
       Version: '2012-10-17'
       Statement:
       - Sid: '1'
         Effect: Allow
         Action: ['*']
         Resource:
           - !Sub 'arn:${AWS::Partition}:cloudformation:${AWS::Region}:
${AWS::AccountId}:stack/awscodestar-${ProjectId}-*'
```

   您可能需要额外的 IAM 权限才能从 AWS CloudFormation 控制台更新堆栈。

4. （可选）如果您要创建特定于应用程序的 IAM 角色，请完成此步骤。从 IAM 控制台中，更新附加到项目的 AWS CloudFormation 角色的内联策略以包含以下代码段。您可能需要额外的 IAM 资源来更新策略。

```
      {
           "Action": [
```

```
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::{AccountId}:role/CodeStar-{ProjectId}*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:GetRole",
                "iam:DeleteRole",
                "iam:DeleteUser"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:DeleteRolePolicy",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:DetachRolePolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutRolePermissionsBoundary"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::{AccountId}:policy/
CodeStar_{ProjectId}_PermissionsBoundary"
                }
            },
            "Effect": "Allow"
        }
```

5. 通过您的项目管道推送更改，以便 AWS CodeStar 使用适当的权限来更新权限边界。

有关更多信息，请参阅向项目添加 IAM 角色。

AWS CodeStar 基于身份的策略示例

默认情况下，IAM 用户和角色没有创建或修改 AWS CodeStar 资源的权限。它们还无法使用AWS Management Console、AWS CLI或AWS API 执行任务。管理员必须创建 IAM policy，以便为用户和角色授予权限以对所需的指定资源执行特定的 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。

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。

主题

策略最佳实践

基于身份的策略确定某个人是否可以创建、访问或删除您账户中的 AWS CodeStar 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- AWS 托管式策略及转向最低权限许可入门 – 要开始向用户和工作负载授予权限，请使用 AWS 托管式策略来为许多常见使用场景授予权限。您可以在 AWS 账户 中找到这些策略。建议通过定义特定于您的使用场景的 AWS 客户管理型策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的 AWS 托管式策略或工作职能的 AWS 托管式策略。
- 应用最低权限 – 在使用 IAM policy 设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM policy 中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果通过特定 AWS 服务（例如 AWS CloudFormation）使用服务操作，您还可以使用条件来授予对服务操作的访问权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM policy，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM policy 语言（JSON）和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，有助于制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Access Analyzer 策略验证。
- 需要多重身份验证（MFA）– 如果您所处的场景要求您的 AWS 账户 中有 IAM 用户或根用户，请启用 MFA 来提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实践的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实践。

AWSCodeStarServiceRole 策略

`aws-codestar-service-role` 策略将附加到允许 AWS CodeStar 对其他服务执行操作的服务角色。首次登录 AWS CodeStar 时，需要创建此服务角色。只需创建一次。创建该服务角色后，系统会自动将此策略附加到该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ProjectEventRules",
            "Effect": "Allow",
```

```
            "Action": [
                "events:PutTargets",
                "events:RemoveTargets",
                "events:PutRule",
                "events:DeleteRule",
                "events:DescribeRule"
            ],
            "Resource": [
                "arn:aws:events:*:*:rule/awscodestar-*"
            ]
        },
        {
            "Sid": "ProjectStack",
            "Effect": "Allow",
            "Action": [
                "cloudformation:*Stack*",
                "cloudformation:CreateChangeSet",
                "cloudformation:ExecuteChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:GetTemplate"
            ],
            "Resource": [
                "arn:aws:cloudformation:*:*:stack/awscodestar-*",
                "arn:aws:cloudformation:*:*:stack/awseb-*",
                "arn:aws:cloudformation:*:*:stack/aws-cloud9-*",
                "arn:aws:cloudformation:*:aws:transform/CodeStar*"
            ]
        },
        {
            "Sid": "ProjectStackTemplate",
            "Effect": "Allow",
            "Action": [
                "cloudformation:GetTemplateSummary",
                "cloudformation:DescribeChangeSet"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectQuickstarts",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::awscodestar-*/*"
            ]
        },
        {
            "Sid": "ProjectS3Buckets",
            "Effect": "Allow",
            "Action": [
                "s3:*"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-*",
                "arn:aws:s3:::elasticbeanstalk-*"
            ]
        },
        {
            "Sid": "ProjectServices",
            "Effect": "Allow",
            "Action": [
                "codestar:*",
                "codecommit:*",
                "codepipeline:*",
                "codedeploy:*",
                "codebuild:*",
                "autoscaling:*",
                "cloudwatch:Put*",
                "ec2:*",
                "elasticbeanstalk:*",
                "elasticloadbalancing:*",
                "iam:ListRoles",
                "logs:*",
                "sns:*",
                "cloud9:CreateEnvironmentEC2",
                "cloud9:DeleteEnvironment",
                "cloud9:DescribeEnvironment*",
                "cloud9:ListEnvironments"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectWorkerRoles",
            "Effect": "Allow",
            "Action": [
                "iam:AttachRolePolicy",
                "iam:CreateRole",
```

```
                "iam:DeleteRole",
                "iam:DeleteRolePolicy",
                "iam:DetachRolePolicy",
                "iam:GetRole",
                "iam:PassRole",
                "iam:GetRolePolicy",
                "iam:PutRolePolicy",
                "iam:SetDefaultPolicyVersion",
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:AddRoleToInstanceProfile",
                "iam:CreateInstanceProfile",
                "iam:DeleteInstanceProfile",
                "iam:RemoveRoleFromInstanceProfile"
            ],
            "Resource": [
                "arn:aws:iam::*:role/CodeStarWorker*",
                "arn:aws:iam::*:policy/CodeStarWorker*",
                "arn:aws:iam::*:instance-profile/awscodestar-*"
            ]
        },
        {
            "Sid": "ProjectTeamMembers",
            "Effect": "Allow",
            "Action": [
                "iam:AttachUserPolicy",
                "iam:DetachUserPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iam:PolicyArn": [
                        "arn:aws:iam::*:policy/CodeStar_*"
                    ]
                }
            }
        },
        {
            "Sid": "ProjectRoles",
            "Effect": "Allow",
            "Action": [
                "iam:CreatePolicy",
                "iam:DeletePolicy",
                "iam:CreatePolicyVersion",
```

```
                "iam:DeletePolicyVersion",
                "iam:ListEntitiesForPolicy",
                "iam:ListPolicyVersions",
                "iam:GetPolicy",
                "iam:GetPolicyVersion"
            ],
            "Resource": [
                "arn:aws:iam::*:policy/CodeStar_*"
            ]
        },
        {
            "Sid": "InspectServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:ListAttachedRolePolicies"
            ],
            "Resource": [
                "arn:aws:iam::*:role/aws-codestar-service-role",
                "arn:aws:iam::*:role/service-role/aws-codestar-service-role"
            ]
        },
        {
            "Sid": "IAMLinkRole",
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": "cloud9.amazonaws.com"
                }
            }
        },
        {
            "Sid": "DescribeConfigRuleForARN",
            "Effect": "Allow",
            "Action": [
                "config:DescribeConfigRules"
            ],
            "Resource": [
                "*"
            ]
        },
```

```
        {
            "Sid": "ProjectCodeStarConnections",
            "Effect": "Allow",
            "Action": [
                "codestar-connections:UseConnection",
                "codestar-connections:GetConnection"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ProjectCodeStarConnectionsPassConnections",
            "Effect": "Allow",
            "Action": "codestar-connections:PassConnection",
            "Resource": "*",
            "Condition": {
                "StringEqualsIfExists": {
                    "codestar-connections:PassedToService":
 "codepipeline.amazonaws.com"
                }
            }
        }
    ]
}
```

AWSCodeStarFullAccess 策略

在 设置 AWS CodeStar 说明中，将名为 `AWSCodeStarFullAccess` 的策略附加到您的 IAM 用户。此策略语句允许用户使用与 AWS 账户关联的所有可用 AWS CodeStar 资源在 AWS CodeStar 中执行所有可用操作。这包括创建和删除项目。以下示例是一个代表性的 `AWSCodeStarFullAccess` 策略的片段。实际策略因您启动新 AWS CodeStar 项目时选择的模板而异。

在没有目标堆栈的情况下调用 `cloudformation::DescribeStacks` 时，AWS CloudFormation 需要 `cloudformation::ListStacks` 权限。

权限详细信息

此策略包含以下操作的权限：

- `ec2`：检索有关 EC2 实例的信息以创建 AWS CodeStar 项目。
- `cloud9`：检索有关 AWS Command Line Interface 环境的信息。
- `cloudformation`：检索有关 AWS CodeStar 项目堆栈的信息。
- `codestar`：在 AWS CodeStar 项目中执行操作。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeStarEC2",
      "Effect": "Allow",
      "Action": [
        "codestar:*",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets",
        "cloud9:DescribeEnvironment*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeStarCF",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStack*",
        "cloudformation:ListStacks*",
        "cloudformation:GetTemplateSummary"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awscodestar-*"
      ]
    }
  ]
}
```

您可能不需要向所有用户授予此访问权限。您可以使用 AWS CodeStar 管理的项目角色添加项目级权限。这些角色授予对 AWS CodeStar 项目的特定级别的访问权限，命名如下：

- 所有者
- 贡献者
- 查看者

AWSCodeStar 所有者角色策略

AWS CodeStar 所有者角色策略允许用户在 AWS CodeStar 项目中不受限制地执行所有操作。AWS CodeStar 将 `CodeStar_`*`project-id`*`_Owner` 策略应用于具有所有者访问级别的项目团队成员。

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:*",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Owner"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:*UserProfile",
    ...
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
  ]
}
...
```

AWS CodeStar 贡献者角色策略

AWS CodeStar 贡献者角色策略允许用户参与项目，并更改项目控制面板。AWS CodeStar 会将 CodeStar_*project-id*_Contributor 策略应用于具有贡献者访问级别的项目团队成员。具有贡献者访问权限的用户可向项目添加内容并可更改控制面板，但不能添加或删除团队成员。

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:Describe*",
    "codestar:Get*",
    "codestar:List*",
    "codestar:PutExtendedAccess",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Contributor"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:*UserProfile",
    ...
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
```

```
  ]
}
...
```

AWS CodeStar 查看者角色策略

AWS CodeStar 查看者角色策略允许用户查看 AWS CodeStar 中的项目。AWS CodeStar 将 CodeStar_*project-id*_Viewer 策略应用于具有查看者访问级别的项目团队成员。具有查看者访问权限的用户可以查看 AWS CodeStar 中的项目，但无法更改其资源，也无法添加或删除团队成员。

```
...
{
  "Effect": "Allow",
  "Action": [
    ...
    "codestar:Describe*",
    "codestar:Get*",
    "codestar:List*",
    ...
  ],
  "Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project/project-id",
    "arn:aws:iam::account-id:policy/CodeStar_project-id_Viewer"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:DescribeUserProfile",
    "codestar:ListProjects",
    "codestar:ListUserProfiles",
    "codestar:VerifyServiceRole",
    ...
  ],
  "Resource": [
    "*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "codestar:*UserProfile",
    ...
```

```
  ],
  "Resource": [
    "arn:aws:iam::account-id:user/user-name"
  ]
}
...
```

AWS CodeStar 工具链工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之后）

对于太平洋夏令时 2018 年 12 月 6 日之后创建的 AWS CodeStar 项目，AWS CodeStar 为工作线程角色（此角色用于在其他 AWS 服务中为您的项目创建资源）创建内联策略。此策略的内容取决于您所创建的项目类型。以下是策略示例。有关更多信息，请参阅面向工作线程角色的 IAM 策略。

```
{
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetObjectVersion",
        "s3:GetBucketVersioning",
        "s3:PutObject*",
        "codecommit:CancelUploadArchive",
        "codecommit:GetBranch",
        "codecommit:GetCommit",
        "codecommit:GetUploadArchiveStatus",
        "codecommit:GitPull",
        "codecommit:UploadArchive",
        "codebuild:StartBuild",
        "codebuild:BatchGetBuilds",
        "codebuild:StopBuild",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeChangeSet",
        "cloudformation:CreateChangeSet",
        "cloudformation:DeleteChangeSet",
        "cloudformation:ExecuteChangeSet",
        "codepipeline:StartPipelineExecution",
        "lambda:ListFunctions",
        "lambda:InvokeFunction",
        "sns:Publish"
      ],
```

```
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

AWS CloudFormation 工作线程角色策略

对于太平洋夏令时 2018 年 12 月 6 日之后创建的 AWS CodeStar 项目，AWS CodeStar 为工作线程角色（此角色为 AWS CodeStar 项目创建 AWS CloudFormation 资源）创建内联策略。该策略的内容取决于您的项目所需的资源类型。以下是策略示例。有关更多信息，请参阅面向工作线程角色的 IAM 策略。

```
{
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
```

```
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-region-id-account-id-project-id",
                "arn:aws:s3:::aws-codestar-region-id-account-id-project-id/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "apigateway:DELETE",
                "apigateway:GET",
                "apigateway:PATCH",
                "apigateway:POST",
                "apigateway:PUT",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentConfig",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:DeleteApplication",
                "codedeploy:DeleteDeployment",
                "codedeploy:DeleteDeploymentConfig",
                "codedeploy:DeleteDeploymentGroup",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:RegisterApplicationRevision",
                "codestar:SyncResources",
                "config:DeleteConfigRule",
                "config:DescribeConfigRules",
                "config:ListTagsForResource",
                "config:PutConfigRule",
                "config:TagResource",
                "config:UntagResource",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:DescribeTable",
                "dynamodb:DescribeTimeToLive",
                "dynamodb:ListTagsOfResource",
                "dynamodb:TagResource",
                "dynamodb:UntagResource",
                "dynamodb:UpdateContinuousBackups",
                "dynamodb:UpdateTable",
```

```
                "dynamodb:UpdateTimeToLive",
                "ec2:AssociateIamInstanceProfile",
                "ec2:AttachVolume",
                "ec2:CreateSecurityGroup",
                "ec2:createTags",
                "ec2:DescribeIamInstanceProfileAssociations",
                "ec2:DescribeInstances",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DetachVolume",
                "ec2:DisassociateIamInstanceProfile",
                "ec2:ModifyInstanceAttribute",
                "ec2:ModifyInstanceCreditSpecification",
                "ec2:ModifyInstancePlacement",
                "ec2:MonitorInstances",
                "ec2:ReplaceIamInstanceProfileAssociation",
                "ec2:RunInstances",
                "ec2:StartInstances",
                "ec2:StopInstances",
                "ec2:TerminateInstances",
                "events:DeleteRule",
                "events:DescribeRule",
                "events:ListTagsForResource",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "events:TagResource",
                "events:UntagResource",
                "kinesis:AddTagsToStream",
                "kinesis:CreateStream",
                "kinesis:DecreaseStreamRetentionPeriod",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "kinesis:IncreaseStreamRetentionPeriod",
                "kinesis:RemoveTagsFromStream",
                "kinesis:StartStreamEncryption",
                "kinesis:StopStreamEncryption",
                "kinesis:UpdateShardCount",
                "lambda:CreateAlias",
                "lambda:CreateFunction",
                "lambda:DeleteAlias",
                "lambda:DeleteFunction",
                "lambda:DeleteFunctionConcurrency",
                "lambda:GetFunction",
```

```
                "lambda:GetFunctionConfiguration",
                "lambda:ListTags",
                "lambda:ListVersionsByFunction",
                "lambda:PublishVersion",
                "lambda:PutFunctionConcurrency",
                "lambda:TagResource",
                "lambda:UntagResource",
                "lambda:UpdateAlias",
                "lambda:UpdateFunctionCode",
                "lambda:UpdateFunctionConfiguration",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "s3:DeleteBucketWebsite",
                "s3:PutAccelerateConfiguration",
                "s3:PutAnalyticsConfiguration",
                "s3:PutBucketAcl",
                "s3:PutBucketCORS",
                "s3:PutBucketLogging",
                "s3:PutBucketNotification",
                "s3:PutBucketPublicAccessBlock",
                "s3:PutBucketVersioning",
                "s3:PutBucketWebsite",
                "s3:PutEncryptionConfiguration",
                "s3:PutInventoryConfiguration",
                "s3:PutLifecycleConfiguration",
                "s3:PutMetricsConfiguration",
                "s3:PutReplicationConfiguration",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:GetTopicAttributes",
                "sns:ListSubscriptionsByTopic",
                "sns:ListTopics",
                "sns:SetSubscriptionAttributes",
                "sns:Subscribe",
                "sns:Unsubscribe",
                "sqs:CreateQueue",
                "sqs:DeleteQueue",
                "sqs:GetQueueAttributes",
                "sqs:GetQueueUrl",
                "sqs:ListQueueTags",
                "sqs:TagQueue",
                "sqs:UntagQueue"
            ],
            "Resource": "*",
```

```
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:AddPermission",
                "lambda:RemovePermission"
            ],
            "Resource": [
                "arn:aws:lambda:region-id:account-id:function:awscodestar-*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStar-project-id*"
            ],
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "codedeploy.amazonaws.com"
                }
            },
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodeDeploy"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:region-id:aws:transform/Serverless-2016-10-31",
                "arn:aws:cloudformation:region-id:aws:transform/CodeStar"
            ],
            "Effect": "Allow"
```

```
        },
        {
            "Action": [
                "iam:CreateServiceLinkedRole",
                "iam:GetRole",
                "iam:DeleteRole",
                "iam:DeleteUser"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "iam:PermissionsBoundary": "arn:aws:iam::account-id:policy/
CodeStar_project-id_PermissionsBoundary"
                }
            },
            "Action": [
                "iam:AttachRolePolicy",
                "iam:AttachUserPolicy",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:DeleteRolePolicy",
                "iam:DeleteUserPolicy",
                "iam:DetachUserPolicy",
                "iam:DetachRolePolicy",
                "iam:PutUserPermissionsBoundary",
                "iam:PutRolePermissionsBoundary"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:CreateKey",
                "kms:CreateAlias",
                "kms:DeleteAlias",
                "kms:DisableKey",
                "kms:EnableKey",
                "kms:UpdateAlias",
                "kms:TagResource",
                "kms:UntagResource"
            ],
```

```
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Condition": {
                "StringEquals": {
                    "ssm:ResourceTag/awscodestar:projectArn":
 "arn:aws:codestar:project-id:account-id:project/project-id"
                }
            },
            "Action": [
                "ssm:GetParameter*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

AWS CloudFormation 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）

如果您的 AWS CodeStar 项目是在太平洋夏令时 2018 年 12 月 6 日之前创建的，那么 AWS CodeStar 会为 AWS CloudFormation 工作线程角色创建内联策略。下面的策略声明是一个示例。

```
{
    "Statement": [
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codestar:SyncResources",
                "lambda:CreateFunction",
```

```
                "lambda:DeleteFunction",
                "lambda:AddPermission",
                "lambda:UpdateFunction",
                "lambda:UpdateFunctionCode",
                "lambda:GetFunction",
                "lambda:GetFunctionConfiguration",
                "lambda:UpdateFunctionConfiguration",
                "lambda:RemovePermission",
                "lambda:listTags",
                "lambda:TagResource",
                "lambda:UntagResource",
                "apigateway:*",
                "dynamodb:CreateTable",
                "dynamodb:DeleteTable",
                "dynamodb:DescribeTable",
                "kinesis:CreateStream",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:ListTopics",
                "sns:GetTopicAttributes",
                "sns:SetTopicAttributes",
                "s3:CreateBucket",
                "s3:DeleteBucket",
                "config:DescribeConfigRules",
                "config:PutConfigRule",
                "config:DeleteConfigRule",
                "ec2:*",
                "autoscaling:*",
                "elasticloadbalancing:*",
                "elasticbeanstalk:*"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda"
            ],
            "Effect": "Allow"
```

```
        },
        {
            "Action": [
                "cloudformation:CreateChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:us-east-1:aws:transform/Serverless-2016-10-31",
                "arn:aws:cloudformation:us-east-1:aws:transform/CodeStar"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodePipeline 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）

如果您的 AWS CodeStar 项目是在太平洋夏令时 2018 年 12 月 6 日之前创建的，那么 AWS CodeStar 会为 CodePipeline 工作线程角色创建内联策略。下面的策略声明是一个示例。

```
{
    "Statement": [
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion",
                "s3:GetBucketVersioning",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codecommit:CancelUploadArchive",
                "codecommit:GetBranch",
                "codecommit:GetCommit",
                "codecommit:GetUploadArchiveStatus",
                "codecommit:UploadArchive"
            ],
            "Resource": [
```

```
                "arn:aws:codecommit:us-east-1:account-id:project-id"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codebuild:StartBuild",
                "codebuild:BatchGetBuilds",
                "codebuild:StopBuild"
            ],
            "Resource": [
                "arn:aws:codebuild:us-east-1:account-id:project/project-id"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "cloudformation:DescribeStacks",
                "cloudformation:DescribeChangeSet",
                "cloudformation:CreateChangeSet",
                "cloudformation:DeleteChangeSet",
                "cloudformation:ExecuteChangeSet"
            ],
            "Resource": [
                "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-
id-lambda/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudFormation"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodeBuild 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）

如果您的 AWS CodeStar 项目是在太平洋夏令时 2018 年 12 月 6 日之前创建的，那么 AWS CodeStar 会为 CodeBuild 工作人员角色创建内联策略。下面的策略声明是一个示例。

```
{
    "Statement": [
        {
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe/*",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app",
                "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app/*"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "codecommit:GitPull"
            ],
            "Resource": [
                "arn:aws:codecommit:us-east-1:account-id:project-id"
            ],
            "Effect": "Allow"
        },
        {
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Encrypt",
```

```
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:account-id:alias/aws/s3"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Amazon CloudWatch Events 工作线程角色策略（太平洋夏令时 2018 年 12 月 6 日之前）

如果您的 AWS CodeStar 项目是在太平洋夏令时 2018 年 12 月 6 日之前创建的，那么 AWS CodeStar 会为 CloudWatch Events 工作线程角色创建内联策略。下面的策略声明是一个示例。

```
{
    "Statement": [
        {
            "Action": [
                "codepipeline:StartPipelineExecution"
            ],
            "Resource": [
                "arn:aws:codepipeline:us-east-1:account-id:project-id-Pipeline"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodeStar 权限边界策略

如果您在太平洋夏令时 2018 年 12 月 6 日之后创建 AWS CodeStar 项目，AWS CodeStar 会为您的项目创建权限边界策略。此策略可防止权限升级到项目之外的资源。它是一个动态策略，随项目的发展而更新。此策略的内容取决于您所创建的项目类型。以下是策略示例。有关更多信息，请参阅IAM 权限边界。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "1",
```

```
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::*/AWSLogs/*/Config/*"
      ]
    },
    {
      "Sid": "2",
      "Effect": "Allow",
      "Action": [
        "*"
      ],
      "Resource": [
        "arn:aws:codestar:us-east-1:account-id:project/project-id",
        "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-id-
lambda/eefbbf20-c1d9-11e8-8a3a-500c28b4e461",
        "arn:aws:cloudformation:us-east-1:account-id:stack/awscodestar-project-
id/4b80b3f0-c1d9-11e8-8517-500c28b236fd",
        "arn:aws:codebuild:us-east-1:account-id:project/project-id",
        "arn:aws:codecommit:us-east-1:account-id:project-id",
        "arn:aws:codepipeline:us-east-1:account-id:project-id-Pipeline",
        "arn:aws:execute-api:us-east-1:account-id:7rlst5mrgi",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudFormation",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CloudWatchEventRule",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodeBuild",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-CodePipeline",
        "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda",
        "arn:aws:lambda:us-east-1:account-id:function:awscodestar-project-id-lambda-
GetHelloWorld-KFKTXYNH9573",
        "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-app",
        "arn:aws:s3:::aws-codestar-us-east-1-account-id-project-id-pipe"
      ]
    },
    {
      "Sid": "3",
      "Effect": "Allow",
      "Action": [
        "apigateway:GET",
        "config:Describe*",
        "config:Get*",
        "config:List*",
        "config:Put*",
```

```
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogGroups",
        "logs:PutLogEvents"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

列出项目的资源

在此示例中，您希望为 AWS 账户中的指定 IAM 用户授予访问权限，以列出 AWS CodeStar 项目的资源。

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:ListResources",
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

使用 AWS CodeStar 控制台

虽然访问 AWS CodeStar 控制台不需要特定权限，但是，除非您具有 AWSCodeStarFullAccess 策略或 AWS CodeStar 项目级别角色之一（所有者、贡献者或查看者），否则，您将无法执行任何有用的操作。有关 AWSCodeStarFullAccess 的更多信息，请参阅 AWSCodeStarFullAccess 策略。有关项目级别策略的更多信息，请参阅面向团队成员的 IAM 策略。

对于只需要调用 AWS CLI 或 AWS API 的用户，无需为其提供最低控制台权限。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策略。此策略包括在控制台上完成此操作或者以编程方式使用 AWS CLI 或 AWS API 所需的权限。

```
{
    "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": "*"
        }
    ]
}
```

更新 AWS CodeStar 项目

在此示例中，您希望为 AWS 账户中的指定 IAM 用户授予访问权限，以编辑 AWS CodeStar 项目属性，例如其项目描述。

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

向项目添加团队成员

在此示例中，您希望授予指定 IAM 用户权限，使其具有向项目 ID 为 *my-first-projec* 的 AWS CodeStar 项目添加团队成员的能力，但明确拒绝该用户删除团队成员的能力：

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:AssociateTeamMember",
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    },
    {
      "Effect" : "Deny",
      "Action" : [
        "codestar:DisassociateTeamMember",
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
      ]

  ]
}
```

列出与 AWS 账户关联的用户配置文件

在此示例中，您允许附加了此策略的 IAM 用户列出与 AWS 账户关联的所有 AWS CodeStar 用户配置文件：

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:ListUserProfiles",
              ],
      "Resource" : "*"
    }
  ]
}
```

基于标签查看 AWS CodeStar 项目

您可以在基于身份的策略中使用条件，以便基于标签控制对 AWS CodeStar 项目的访问。此示例说明了如何创建允许查看项目的策略。但是，仅当项目标签 Owner 的值为该用户的用户名时，才会授予此权限。此策略还授予在控制台上完成此操作的必要权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListProjectsInConsole",
            "Effect": "Allow",
            "Action": "codestar:ListProjects",
            "Resource": "*"
        },
        {
            "Sid": "ViewProjectIfOwner",
            "Effect": "Allow",
            "Action": "codestar:GetProject,
            "Resource": "arn:aws:codestar:*:*:project/*",
            "Condition": {
                "StringEquals": {"codestar:ResourceTag/Owner": "${aws:username}"}
            }
        }
```

```
    ]
}
```

您可以将该策略附加到您账户中的 IAM 用户。如果名为 `richard-roe` 的用户尝试查看 AWS CodeStar 项目，则必须将项目标记为 `Owner=richard-roe` 或 `owner=richard-roe`。否则，将拒绝其访问。条件标签键 `Owner` 匹配 `Owner` 和 `owner`，因为条件键名称不区分大小写。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。

对 AWS 托管式策略的 AWS CodeStar 更新

查看有关自此服务开始跟踪这些更改起，AWS CodeStar 的 AWS 托管式策略更新的详细信息。有关此页面更改的自动提示，请订阅 AWS CodeStar 文档历史记录页面上的 RSS 源。

更改	说明	日期
AWSCodeStarFullAccess 策略：更新 AWSCodeStarFullAccess 策略	更新了 AWS CodeStar 访问角色策略。该策略的结果是一样的，但是 cloudformation 除了已经需要的 DescribeStacks 之外还需要 ListStacks。	2023 年 3 月 24 日
AWSCodeStarServiceRole 策略：更新 AWSCodeStarServiceRole 策略	AWS CodeStar 服务角色的策略已更新，更正了策略声明中的冗余操作。 服务角色策略允许 AWS CodeStar 服务代表您执行操作。	2021 年 9 月 23 日
AWS CodeStar 已开启跟踪更改	AWS CodeStar 为其 AWS 托管式策略开启了跟踪更改。	2021 年 9 月 23 日

排查 AWS CodeStar 身份和访问的问题

使用以下信息可帮助您诊断和修复在使用 AWS CodeStar 和 IAM 时可能遇到的常见问题。

主题

- 我无权在 AWS CodeStar 中执行操作
- 我无权执行 iam:PassRole
- 我想要允许我的 AWS 账户之外的用户访问我的 AWS CodeStar 资源

我无权在 AWS CodeStar 中执行操作

如果 AWS Management Console 告诉您，您无权执行某个操作，请联系您的管理员寻求帮助。您的管理员是提供登录凭证的人。

当 `mateojackson` IAM 用户尝试使用控制台查看有关 *`widget`* 的详细信息，但不具有 `codestar:`*`GetWidget`* 权限时，会发生以下示例错误。

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

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `codestar:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

我无权执行 iam:PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS CodeStar。

有些 AWS 服务 允许将现有角色传递到该服务，而不是创建新服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 AWS CodeStar 中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

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

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系 AWS 管理员。您的管理员是提供登录凭证的人。

我想要允许我的 AWS 账户之外的用户访问我的 AWS CodeStar 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，可以代入角色。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 是否支持这些功能，请参阅 AWS CodeStar 如何与 IAM 配合使用。
- 要了解如何为您拥有的 AWS 账户 中的资源提供访问权限，请参阅《IAM 用户指南》中的为您拥有的另一个 AWS 账户 中的 IAM 用户提供访问权限。
- 要了解如何为第三方 AWS 账户 提供您的资源的访问权限，请参阅《IAM 用户指南》中的为第三方拥有的 AWS 账户 提供访问权限。
- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的为经过外部身份验证的用户（身份联合验证）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。

使用 AWS CloudTrail 记录 AWS CodeStar API 调用

AWS CodeStar 与 AWS CloudTrail 集成，后者是在 AWS 中记录用户、角色或 AWS CodeStar 服务所执行操作的服务。CloudTrail 将 AWS CodeStar 的所有 API 调用作为事件捕获。捕获的调用包含来自 AWS CodeStar 控制台的调用和对 AWS CodeStar 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 S3 存储桶（包括 AWS CodeStar 的事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的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 CodeStar 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。

要了解有关 CloudTrail 的更多信息，请参阅《AWS CloudTrail 用户指南》。

CloudTrail 中的 AWS CodeStar 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 AWS CodeStar 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 的事件），请创建跟踪。默认情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪在 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 S3 存储桶。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措施。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 操作，AWS CodeStar API 参考中介绍了这些操作。例如，对 `DescribeProject`、`UpdateProject` 和 `AssociateTeamMember` 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 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 AWS CodeStar 日志文件条目

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，该条目阐述了正在 AWS CodeStar 中调用的 `CreateProject` 操作：

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROAJLIN2OF3UBEXAMPLE:role-name",
    "arn": "arn:aws:sts::account-ID:assumed-role/role-name/role-session-name",
    "accountId": "account-ID",
    "accessKeyId": "ASIAJ44LFQS5XEXAMPLE",
    "sessionContext": {
      "attributes": {
```

```
        "mfaAuthenticated": "false",
        "creationDate": "2017-06-04T23:56:57Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROAJLIN2OF3UBEXAMPLE",
        "arn": "arn:aws:iam::account-ID:role/service-role/role-name",
        "accountId": "account-ID",
        "userName": "role-name"
      }
    },
    "invokedBy": "codestar.amazonaws.com"
  },
  "eventTime": "2017-06-04T23:56:57Z",
  "eventSource": "codestar.amazonaws.com",
  "eventName": "CreateProject",
  "awsRegion": "region-ID",
  "sourceIPAddress": "codestar.amazonaws.com",
  "userAgent": "codestar.amazonaws.com",
  "requestParameters": {
    "clientRequestToken": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "id": "project-ID",
    "stackId": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "description": "AWS CodeStar created project",
    "name": "project-name",
    "projectTemplateId": "arn:aws:codestar:region-ID::project-template/project-
template-name"
  },
  "responseElements": {
    "projectTemplateId": "arn:aws:codestar:region-ID::project-template/project-
template-name",
    "arn": "arn:aws:codestar:us-east-1:account-ID:project/project-ID",
    "clientRequestToken": "arn:aws:cloudformation:region-ID:account-ID:stack/stack-
name/additional-ID",
    "id": "project-ID"
  },
  "requestID": "7d7556d0-4981-11e7-a3bc-dd5daEXAMPLE",
  "eventID": "6b0d6e28-7a1e-4a73-981b-c8fdbEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "account-ID"
}
```

AWS CodeStar 的合规性验证

AWS CodeStar 不在任何 AWS 合规性计划的范围内。

有关特定合规性计划范围内的AWS服务的列表，请参阅合规性计划范围内的AWS服务。有关一般信息，请参阅AWS合规性计划。

您可以使用 AWS Artifact 下载第三方审计报告。有关更多信息，请参阅下载 AWS Artifact 中的报告。

AWS CodeStar 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。

AWS CodeStar 中的基础设施安全性

作为一项托管式服务，AWS CodeStar 受 AWS 全球网络安全保护。有关 AWS 安全服务以及 AWS 如何保护基础架构的信息，请参阅 AWS 云安全。要按照基础设施安全最佳实践设计您的 AWS 环境，请参阅《安全性支柱 AWS Well-Architected Framework》中的 基础设施保护。

您可以使用 AWS 发布的 API 调用通过网络访问 CodeStar。客户端必须支持以下内容：

- 传输层安全性协议 (TLS) 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 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。

默认情况下， AWS CodeStar 不会隔离服务流量。使用 AWS CodeStar 创建的项目面向公共互联网开放，除非您手动修改通过 Amazon EC2、API Gateway 或 Elastic Beanstalk 的访问设置。这是有意设计的。您可以根据需要修改 Amazon EC2、API Gateway 或 Elastic Beanstalk 中的访问设置，包括阻止所有互联网访问。

默认情况下，AWS CodeStar 不支持 VPC 端点（AWS PrivateLink），但您可以直接在项目资源上配置该支持。

AWS CodeStar 中的限制

下表介绍了 AWS CodeStar 中的限制。AWS CodeStar 依赖于项目资源的其他 AWS 服务。这些服务限制中有一些可以更改。有关可以更改的限制的信息，请参阅 AWS 服务限制。

项目数	一个 AWS 账户中的最大项目数为 333。实际限制因其他服务依赖项的级别而异（例如您的 AWS 账户允许的 CodePipeline 中的最大管道数）。
一个 IAM 用户可以属于的 AWS CodeStar 项目数	每个单独的 IAM 用户最多 10 个。
项目 ID	一个 AWS 账户中的项目 ID 必须唯一。项目 ID 的长度必须至少为 2 个字符，且不能超过 15 个字符。允许的字符包括： 字母 a 至 z（含这两个字母）。 数字 0 至 9（含这两个数字）。 特殊字符 -（减号）。 不允许使用任何其他字符，如大写字母、空格、.（句点）、@ 符号、_（下划线）。
项目名称	项目名称的长度不能超过 100 个字符，并且不能以空格开头或结尾。
产品描述	任意字符组合，长度在 0 到 1,024 个字符之间。产品描述是可选的。
AWS CodeStar 项目中的团队成员	100
用户配置文件中的显示名称	任意字符组合，长度在 1 到 100 个字符之间。显示名称必须至少包含一个字符。该字符不能为空格。显示名称不能以空格开头或结尾。

用户配置文件中的电子邮件地址	电子邮件地址必须包含一个 @ 并以有效的域扩展名结尾。
对 AWS CodeStar 的联合访问权限、根账户访问权限或临时访问权限	AWS CodeStar 支持联合身份用户和使用临时访问凭证。不建议将 AWS CodeStar 用于根账户。
IAM 角色	附加到 IAM 角色的任何托管策略中的字符最多为 5,120 个。

故障排除 AWS CodeStar

以下信息可帮助您处理 AWS CodeStar 中的常见问题。

主题

项目创建失败：未创建项目

问题：在尝试创建项目时，您看到一条指出创建失败的消息。

可能的修复措施：导致失败的最常见原因有：

- 带此 ID 的项目已存在于您的 AWS 账户中，但可能位于另一个 AWS 区域中。
- 您用于登录 AWS Management Console 的 IAM 用户不具有创建项目所需的权限。
- AWS CodeStar 服务角色缺少一个或多个所需权限。
- 您已达到项目的一个或多个资源的最大限制（例如，有关 IAM 中的客户管理型策略、Amazon S3 存储桶或 CodePipeline 中的管道的限制）。

在创建项目之前，请验证您是否已将 AWSCodeStarFullAccess 策略应用于您的 IAM 用户。有关更多信息，请参阅 AWSCodeStarFullAccess 策略。

在创建项目时，请确保 ID 是唯一的，并且满足 AWS CodeStar 要求。请务必已选中 AWS CodeStar would like permission to administer AWS resources on your behalf 复选框。

要排查其他问题，请打开 AWS CloudFormation 控制台，为您尝试创建的项目选择堆栈，然后选择事件选项卡。一个项目可能有多个堆栈。堆栈名称以 awscodestar- 开头，后跟项目 ID。堆栈可能位于 Deleted 筛选视图下。查看堆栈事件中的任何失败消息并纠正作为这些失败的原因列出的问题。

项目创建：我在创建项目的过程中尝试编辑 Amazon EC2 配置时，出现一个错误

问题：在项目创建过程中编辑 Amazon EC2 配置选项时，您看到一条错误消息或灰显选项，并且无法继续创建项目。

可能的修复措施：导致错误消息的最常见原因有：

- AWS CodeStar 项目模板中的 VPC（默认 VPC 或编辑 Amazon EC2 配置时使用的 VPC）具有专用实例租赁，并且专用实例不支持该实例类型。请选择另一个实例类型或另一个 Amazon VPC。
- 您的 AWS 账户没有 Amazon VPC。您可能已删除默认 VPC，并且未创建任何其他 VPC。通过 https://console.aws.amazon.com/vpc/ 打开 Amazon VPC 控制台，选择您的 VPC，并确保至少配置了一个 VPC。如果没有 VPC，请创建一个 VPC。有关更多信息，请参阅 Amazon Virtual Private Cloud 概览 中的 Amazon VPC 入门指南。
- Amazon VPC 没有任何子网。请选择另一个 VPC，或者为该 VPC 创建子网。有关更多信息，请参阅 VPC 和子网基础知识。

项目删除：已删除 AWS CodeStar 项目，但资源仍存在

问题：已删除 AWS CodeStar 项目，但为该项目创建的资源仍存在。默认情况下，在删除项目时，AWS CodeStar 将删除项目资源。即使用户选中删除资源复选框，某些资源（例如 Amazon S3 存储桶）也将保留，因为存储桶可能包含数据。

可能的修复措施：打开 AWS CloudFormation 控制台，然后找到用于创建项目的一个或多个 AWS CloudFormation 堆栈。堆栈名称以 awscodestar- 开头，后跟项目 ID。这些堆栈可能位于 Deleted 筛选视图下。查看与堆栈关联的事件以发现为项目创建的资源。在您创建了 AWS CodeStar 项目的 AWS 区域中为每项资源打开控制台，然后手动删除这些资源。

可能保留的项目资源包括：

- Amazon S3 中的一个或多个项目存储桶。与其他项目资源不同，Amazon S3 中的项目存储桶在选中删除关联的 AWS 资源以及 AWS CodeStar 项目复选框后不会被删除。

  通过以下网址打开 Simple Storage Service（Amazon S3）控制台：https://console.aws.amazon.com/s3/。
- CodeCommit 中项目的源存储库。

  打开 CodeCommit 控制台 (https://console.aws.amazon.com/codecommit)。
- CodePipeline 中项目的管道。

  在 https://console.aws.amazon.com/codepipeline/ 打开 CodePipeline 控制台。
- CodeDeploy 中的应用程序和关联的部署组。

  从 https://console.aws.amazon.com/codedeploy/ 打开 CodeDeploy 控制台。
- AWS Elastic Beanstalk 中的应用程序和关联的环境。

  从 https://console.aws.amazon.com/elasticbeanstalk/ 打开 Elastic Beanstalk 控制台。
- AWS Lambda 中的函数。

  通过 https://console.aws.amazon.com/lambda/ 打开 AWS Lambda 控制台。
- API Gateway 中的一个或多个 API。

  打开 API Gateway 控制台，网址为：https://console.aws.amazon.com/apigateway/。
- IAM 中的一个或多个 IAM 策略或角色。

  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/。
- Amazon EC2 中的一个实例。

  通过以下网址打开 Amazon EC2 控制台：https://console.aws.amazon.com/ec2/。
- AWS Cloud9 中的一个或多个开发环境。

  要查看、访问和管理开发环境，请通过以下网址打开 AWS Cloud9 控制台：https://console.aws.amazon.com/cloud9/。

如果项目使用了 AWS 以外的资源（例如，GitHub 存储库或 Atlassian JIRA 中的问题），即使选中删除关联的 AWS 资源以及 CodeStar 项目复选框，系统也不会删除这些资源。

团队管理失败：无法将 IAM 用户添加到 AWS CodeStar 项目中的团队

问题：在尝试将用户添加到项目时，您看到一条错误消息，指出添加失败。

可能的修复措施：导致该错误的最常见原因是，用户已达到可在 IAM 中向用户应用的托管策略数限制。如果您在其中尝试添加用户的 AWS CodeStar 项目中没有所有者角色，或者 IAM 用户不存在或已被删除，则您也可能会收到此错误。

请确保您以作为该 AWS CodeStar 项目中的所有者的用户身份进行登录。有关更多信息，请参阅向 AWS CodeStar 项目添加团队成员 。

要排查其他问题，请打开 IAM 控制台，选择您尝试添加的用户，然后检查已向该 IAM 用户应用的托管策略数。

有关更多信息，请参阅 IAM 实体和对象的限制。有关可更改的限制，请参阅 AWS 服务限制。

访问失败：联合身份用户无法访问 AWS CodeStar 项目

问题：联合身份用户无法在 AWS CodeStar 控制台中查看项目。

可能的修复措施：如果您以联合身份用户登录，请确保您已将适当的托管策略附加到您登录时代入的角色。有关更多信息，请参阅将您的项目的 AWS CodeStar 查看者/贡献者/所有者托管策略附加到联合身份用户的角色。

通过手动附加策略，将联合身份用户添加到 AWS Cloud9 环境。请参阅将 AWS Cloud9 托管策略附加到联合身份用户的角色。

访问失败：联合身份用户无法访问或创建 AWS Cloud9 环境

问题：联合身份用户无法在 AWS Cloud9 控制台中查看或创建AWS Cloud9 环境。

可能的修复措施：如果您以联合身份用户登录，请确保您已将适当的托管策略附加到联合身份用户的角色。

通过手动将策略附加到联合身份用户的角色，向 AWS Cloud9 环境添加联合身份用户。请参阅将 AWS Cloud9 托管策略附加到联合身份用户的角色。

访问失败：联合身份用户可以创建 AWS CodeStar 项目，但无法查看项目资源

问题：联合身份用户能够创建项目，但无法查看项目资源，如项目管道。

可能的修复措施：如果您附加了 **AWSCodeStarFullAccess** 托管策略，您将有权在 AWS CodeStar 中创建项目。但是，要访问所有项目资源，您必须附加所有者托管策略。

在 AWS CodeStar 创建项目资源后，对所有项目资源的项目权限都可在所有者、贡献者和查看者托管策略中提供。要访问所有资源，您必须手动将所有者策略附加到您的角色。请参阅步骤 3：配置用户的 IAM 权限。

服务角色问题：无法创建服务角色

问题：在 AWS CodeStar 中尝试创建项目时，您看到一条提示您创建服务角色的消息。在选择该选项以创建服务角色时，您看到一条错误。

可能的修复措施：导致该错误的最常见原因是，您用来登录 AWS 的账户的权限不足，无法创建服务角色。要创建 AWS CodeStar 服务角色 (`aws-codestar-service-role`)，您必须以管理用户身份登录或使用根账户登录。从控制台注销，然后使用已应用 `AdministratorAccess` 托管策略的 IAM 用户登录。

服务角色问题：服务角色无效或缺失

问题：打开 AWS CodeStar 控制台时，您看到一条消息，指出 AWS CodeStar 服务角色缺失或无效。

可能的修复措施：导致该错误的最常见原因是，管理用户已编辑或删除该服务角色 (`aws-codestar-service-role`)。如果已删除该服务角色，系统将提示您创建它。您必须以管理用户身份登录或使用根账户登录才能创建该角色。如果该角色已经过编辑，则它将不再有效。以管理用户身份登录 IAM 控制台，在角色列表中找到该服务角色，然后删除它。切换到 AWS CodeStar 控制台并按照屏幕上的说明创建该服务角色。

项目角色问题：AWS CodeStar 项目中的实例的 AWS Elastic Beanstalk 运行状况检查失败

问题：如果您创建的 AWS CodeStar 项目包含 2017 年 9 月 22 日之前的 Elastic Beanstalk，则 Elastic Beanstalk 运行状况检查可能会失败。如果您自创建项目以来尚未更改 Elastic Beanstalk 配置，则运行

状况检查失败并报告灰显状态。尽管运行状况检查失败，但您的应用程序仍正常运行。如果您自创建项目以来更改了 Elastic Beanstalk 配置，则运行状况检查失败并且您的应用程序可能无法正常运行。

修复措施：一个或多个 IAM 角色缺少必需的 IAM 策略语句。将缺少的策略添加到 AWS 账户中的受影响角色。

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/。

   （如果您无法执行此操作，请咨询您的 AWS 账户管理员以寻求帮助。）
2. 在导航窗格中，选择角色。
3. 在角色列表中，选择 CodeStarWorker-***Project-ID***-EB，其中 *Project-ID* 是受影响项目的 ID。（如果您无法轻松地在列表中找到角色，请在搜索框中输入角色的部分名称或全称。）
4. 在权限选项卡上，选择附加策略。
5. 在策略列表中，选择 AWSElasticBeanstalkEnhancedHealth 和 AWSElasticBeanstalkService。（如果您无法轻松地在列表中找到策略，请在搜索框中输入策略的部分名称或全称。）
6. 选择附加策略。
7. 对于其名称遵循 CodeStarWorker-***Project-ID***-EB 模式的每个受影响角色，重复步骤 3 到 6。

项目角色问题：项目角色无效或缺失

问题：尝试向项目添加用户时，您看到一条错误消息，指出由于项目角色的策略缺失或无效，该添加失败。

可能的修复措施：导致该错误的最常见原因是，已在 IAM 中编辑或删除一个或多个项目策略。项目策略对于 AWS CodeStar 项目是唯一的，且无法重新创建。无法使用该项目。在 AWS CodeStar 中创建一个项目，然后将数据迁移到该新项目。从不可用项目的存储库克隆项目代码，然后将该代码推送至新项目的存储库。将团队 wiki 信息从旧项目复制到新项目。向新项目添加用户。在您确定已迁移所有数据和设置后，删除不可用的项目。

项目扩展：无法连接到 JIRA

问题：当您使用 Atlassian JIRA 扩展尝试将 AWS CodeStar 项目连接到 JIRA 实例时，将显示以下消息：“该 URL 不是有效 JIRA URL。验证 URL 是否正确。”

可能的修复措施：

- 请确保 JIRA URL 正确，然后再次尝试连接。
- 可能无法从公共 Internet 访问您的自承载 JIRA 实例。请联系您的网络管理员以确保可从公共 Internet 访问您的 JIRA 实例，然后再次尝试连接。

GitHub：无法访问存储库的提交历史记录、问题或代码

问题：在某一项目（在 GitHub 中存储其代码）的控制面板中，提交历史记录和 GitHub 问题磁贴显示连接错误，或在这些磁贴中选择在 GitHub 中打开或创建问题时显示错误。

可能的原因：

- AWS CodeStar 项目可能无法再访问 GitHub 存储库。
- 存储库可能已在 GitHub 中删除或重命名。

AWS CloudFormation：由于缺少权限，堆栈创建已回滚

在将资源添加到 `template.yml` 文件后，查看 AWS CloudFormation 堆栈更新是否显示任何错误消息。如果未满足特定条件（例如，缺少所需的资源权限时），堆栈更新将失败。

> **Note**
>
> 截至 2019 年 5 月 2 日，我们已为所有的现有项目更新 AWS CloudFormation 工作线程角色策略。此更新减小了授予您的项目管道的访问权限的范围，以提高项目的安全性。

要排查问题，请在项目管道的 AWS CodeStar 控制面板视图中查看失败状态。

然后，选择您的管道的部署阶段中的 CloudFormation 链接以在 AWS CloudFormation 控制台中对失败进行问题排查。要查看堆栈创建详细信息，请展开项目的事件列表，然后查看任何失败消息。该消息指出缺少哪些权限。改正 AWS CloudFormation 工作线程角色策略，然后再次执行管道。

AWS CloudFormation 无权在 Lambda 执行角色上执行 iam:PassRole

如果您具有在太平洋夏令时 2018 年 12 月 6 日之前创建的项目，而该项目会创建 Lambda 函数，则您可能会看到类似如下的 AWS CloudFormation 错误：

```
User: arn:aws:sts::id:assumed-role/CodeStarWorker-project-id-CloudFormation/
AWSCloudFormation is not authorized to perform: iam:PassRole on resource:
 arn:aws:iam::id:role/CodeStarWorker-project-id-Lambda (Service: AWSLambdaInternal;
 Status Code: 403; Error Code: AccessDeniedException; Request ID: id)
```

出现此错误的原因是您的 AWS CloudFormation 工作线程角色无权传递用于预配置新 Lambda 函数的角色。

要修复此错误，您需要使用以下代码段更新您的 AWS CloudFormation 工作线程角色策略。

```
{
        "Action":[ "iam:PassRole" ],
        "Resource": [
           "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda",
           ],

        "Effect": "Allow"
}
```

在更新该策略之后，请再次执行您的管道。

或者，您也可以按照 将 IAM 权限边界添加到现有项目 中所述，通过将权限边界添加到您的项目中，为您的 Lambda 函数使用自定义角色

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。

AWS CodeStar 用户指南发布说明

下表描述了 AWS CodeStar 用户指南的每次发布中所做的重要更改。如需对此文档更新的通知，您可以订阅 RSS 源。

变更	说明	日期
访问策略更新	更新了 AWS CodeStar 访问角色策略。该策略的结果是一样的，但是 cloudformation 除了已经需要的 DescribeStacks 之外还需要 ListStacks。要参考更新后的策略，请参阅 AWSCodeStarFullAccess 策略。	2023 年 3 月 24 日
服务角色策略更新	更新了 AWS CodeStar 服务角色策略。要参考更新后的策略，请参阅 AWSCodeStarServiceRole 策略。	2021 年 9 月 23 日
将连接资源用于具有 GitHub 源存储库的项目	当您使用控制台通过 GitHub 存储库在 AWS CodeStar 中创建项目时，系统将使用连接资源来管理您的 GitHub 操作。连接使用 GitHub 应用程序，而之前的 GitHub 授权使用 OAuth。有关说明如何创建使用到 GitHub 的连接的项目的教程，请参阅教程：使用 GitHub 源代码库创建项目。本教程还向您展示了如何为项目源代码库创建、查看和合并拉取请求。	2021 年 4 月 27 日

AWS CodeStar 支持在美国西部（北加利福尼亚）区域使用 AWS Cloud9	AWS CodeStar 现在支持在美国西部（北加利福尼亚）区域使用 AWS Cloud9。有关更多信息，请参阅设置 Cloud9。	2021 年 2 月 16 日
更新文档以反映新的控制台体验	2020 年 8 月 12 日，AWS CodeStar 服务在 AWS 控制台中改为全新的用户体验。更新了用户指南，以适应全新控制台体验。	2020 年 8 月 12 日
可使用 AWS CodeStar CLI 创建 AWS CodeStar 项目	可使用 CLI 命令创建 AWS CodeStar 项目。AWS CodeStar 使用源代码和您提供的工具链模板创建您的项目和基础设施。请参阅在 AWS CodeStar 中创建项目 (AWS CLI)。	2018 年 10 月 24 日
所有 AWS CodeStar 项目模板现在包含关于基础设施更新的 AWS CloudFormation 文件	AWS CodeStar 与 AWS CloudFormation 配合使用以允许您使用代码在云中创建支持服务和服务器或无服务器平台。AWS CloudFormation 文件现在可用于所有 AWS CodeStar 项目模板类型（使用 Lambda、EC2 或 Elastic Beanstalk 计算平台的模板）。此文件存储在源存储库中的 `template.yml` 中。您可以查看和修改此文件以向您的项目中添加资源。请参阅项目模板。	2018 年 8 月 3 日

现在可通过 RSS 获得 AWS CodeStar 用户指南更新通知	HTML 版本的 AWS CodeStar 用户指南现在支持更新的 RSS 源，此类更新记载在“文档更新发布说明”页面上。RSS 源包括 2018 年 6 月 30 日及之后所做的更新。此前宣布的更新仍在“文档更新发布说明”页面中提供。使用顶部菜单面板中的 RSS 按钮，订阅此源。	2018 年 6 月 30 日

下表描述了 2018 年 6 月 30 日之前 AWS CodeStar 用户指南的每次发布中所做的重要更改。

更改	描述	更改日期
AWS CodeStar 用户指南现在通过 GitHub 提供	本指南现在通过 GitHub 提供。您也可以使用 GitHub 提交反馈和更改本指南内容的请求。有关更多信息，请在指南的导航栏中选择在 GitHub 上编辑图标，或者参阅 GitHub 网站上的 awsdocs/aws-codestar-user-guide 存储库。	2018 年 2 月 22 日
AWS CodeStar 现已在亚太地区（首尔）推出	AWS CodeStar 现已在亚太（首尔）区域提供。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 AWS CodeStar。	2018 年 2 月 14 日
AWS CodeStar 现已在亚太地区（东京）和加拿大（中部）推出	AWS CodeStar 现已在亚太地区（东京）和加拿大（中部）推出。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 AWS CodeStar。	2017 年 12 月 20 日
AWS CodeStar 现在支持 AWS Cloud9	AWS CodeStar 现在支持使用 AWS Cloud9（基于 Web 浏览器的联机 IDE）处理项目代码。有关更多信息，请参阅将 AWS Cloud9 与 AWS CodeStar 结合使用。 有关受支持的 AWS 区域的列表，请参阅 Amazon Web Services 一般参考 中的 AWS Cloud9。	2017 年 11 月 30 日
AWS CodeStar 现在支持 GitHub	AWS CodeStar 现在支持在 GitHub 中存储项目代码。有关更多信息，请参阅创建项目。	2017 年 10 月 12 日

更改	描述	更改日期
AWS CodeStar 现已在美国西部（北加利福尼亚）和欧洲地区（伦敦）推出	AWS CodeStar 现已在美国西部（北加利福尼亚）和欧洲地区（伦敦）推出。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 AWS CodeStar。	2017 年 17 月 8 日
AWS CodeStar 现已在亚太地区（悉尼）、亚太地区（新加坡）和欧洲地区（法兰克福）推出	AWS CodeStar 现已在亚太地区（悉尼）、亚太地区（新加坡）和欧洲地区（法兰克福）推出。有关更多信息，请参阅 Amazon Web Services 一般参考 中的 AWS CodeStar。	2017 年 25 月 7 日
AWS CloudTrail 现在支持 AWS CodeStar	AWS CodeStar 现在与 CloudTrail 集成在一起，后者是一种服务，可在 AWS 账户中捕获由 AWS CodeStar 或代表它发出的 API 调用，并将日志文件提交到您指定的 Amazon S3 存储桶。有关更多信息，请参阅使用 AWS CloudTrail 记录 AWS CodeStar API 调用。	2017 年 6 月 14 日
初始版本	这是 AWS CodeStar 用户指南 的首个版本。	2017 年 4 月 19 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。