使用者指南

AWS CodeStar

AWS CodeStar: 使用者指南

Table of Contents

Amazon Web Services (AWS) 將於 2024 年 7 月 31 日停止支援建立和檢視 AWS CodeStar 專案。2024 年 7 月 31 日之後，您將無法再存取 AWS CodeStar 主控台或建立新專案。不過， 建立 AWS 的資源 AWS CodeStar，包括您的來源儲存庫、管道和組建，將不受此變更影響，且將繼續運作。 AWS CodeStar 連線和 AWS CodeStar 通知不會受此中止的影響。

如果您想要追蹤工作、開發程式碼，以及建置、測試和部署應用程式，Amazon CodeCatalyst 會提供簡化的入門程序和其他功能來管理您的軟體專案。進一步了解 Amazon CodeCatalyst 的功能和定價。

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。

什麼是 AWS CodeStar？

AWS CodeStar 是以雲端為基礎的服務，可用於建立、管理和使用軟體開發專案 AWS。您可以使用 AWS CodeStar 專案， AWS 在 上快速開發、建置和部署應用程式。 AWS CodeStar 專案會為您的專案開發工具鏈建立和整合 AWS 服務。根據您選擇的 AWS CodeStar 專案範本，該工具鏈可能包含來源控制、建置、部署、虛擬伺服器或無伺服器資源等。 AWS CodeStar 也會管理專案使用者（稱為團隊成員）所需的許可。透過將使用者新增為 AWS CodeStar 專案的團隊成員，專案擁有者可以快速且簡單地授予每個團隊成員對專案及其資源的適當角色存取權。

主題

我可以使用什麼 AWS CodeStar？

您可以使用 AWS CodeStar 來協助您在雲端中設定應用程式開發，並從單一集中式儀表板管理您的開發。具體而言，您可以：

- 使用適用於 Web 應用程式、Web 服務等的範本 AWS ，在 上在幾分鐘內啟動新的軟體專案： AWS CodeStar 包含各種專案類型和程式設計語言的專案範本。由於 AWS CodeStar 負責設定，因此所有專案資源都會設定為一起運作。
- 管理您團隊的專案存取權： AWS CodeStar 提供一個集中式主控台，其中可讓您指派專案團隊成員所需的角色以存取工具與資源。這些許可會自動套用至專案中使用的所有 AWS 服務，因此您不需要建立或管理複雜的 IAM 政策。
- 在一個位置視覺化、操作和協作您的專案： AWS CodeStar 包含專案儀表板，提供專案、其工具鏈和重要事件的整體檢視。您可以監控最新的專案活動，像是最新程式碼遞交、程式碼變更狀態追蹤、建置結果和部署，所有的操作都透過相同網頁執行。您可以從單一儀表板監控專案之進行中狀況，並深入問題進行調查。
- 快速快速逐一查看所有所需的工具： AWS CodeStar 包含您專案的整合式開發工具鏈。團隊成員推送程式碼，變更會自動部署。整合問題追蹤可讓團隊成員追蹤後續應採取的行動。您和您的團隊可以更快速有效地在所有程式碼交付階段共同作業。

如何開始使用 AWS CodeStar？

若要開始使用 AWS CodeStar：

1. AWS CodeStar 請依照中的步驟準備使用 設定 AWS CodeStar。
2. 依照入門 AWS CodeStar教學課程中的步驟，試驗 AWS CodeStar 。
3. 依照中的步驟與其他開發人員共用您的專案將團隊成員新增至 AWS CodeStar 專案 。
4. 依照中的步驟整合您最愛的 IDE搭配 使用 IDE AWS CodeStar。

設定 AWS CodeStar

您必須先完成下列步驟 AWS CodeStar，才能開始使用 。

主題

步驟 1：建立 帳戶

註冊 AWS 帳戶

如果您沒有 AWS 帳戶，請完成下列步驟來建立一個。

註冊 AWS 帳戶

1. 開啟 https://portal.aws.amazon.com/billing/signup。
2. 請遵循線上指示進行。

   部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

   當您註冊 時 AWS 帳戶，AWS 帳戶根使用者會建立 。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。作為安全最佳實務，請將管理存取權指派給使用者，並且僅使用根使用者來執行需要根使用者存取權的任務。

AWS 會在註冊程序完成後傳送確認電子郵件給您。您可以隨時登錄 https://aws.amazon.com/ 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

建立具有管理存取權的使用者

註冊 之後 AWS 帳戶，請保護您的 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立管理使用者，以免將根使用者用於日常任務。

保護您的 AWS 帳戶根使用者

1. 選擇根使用者並輸入 AWS 帳戶 您的電子郵件地址，以帳戶擁有者AWS Management Console身分登入 。在下一頁中，輸入您的密碼。

   如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的以根使用者身分登入。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

   如需說明，請參閱《IAM 使用者指南》中的為您的 AWS 帳戶 根使用者 （主控台） 啟用虛擬 MFA 裝置。

建立具有管理存取權的使用者

1. 啟用 IAM Identity Center。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的啟用 AWS IAM Identity Center。

2. 在 IAM Identity Center 中，將管理存取權授予使用者。

   如需使用 IAM Identity Center 目錄 做為身分來源的教學課程，請參閱AWS IAM Identity Center《 使用者指南》中的使用預設值設定使用者存取 IAM Identity Center 目錄。

以具有管理存取權的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

  如需使用 IAM Identity Center 使用者登入的說明，請參閱AWS 登入 《 使用者指南》中的登入 AWS 存取入口網站。

指派存取權給其他使用者

1. 在 IAM Identity Center 中，建立一個許可集來遵循套用最低權限的最佳實務。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的建立許可集。

2. 將使用者指派至群組，然後對該群組指派單一登入存取權。

   如需指示，請參閱《AWS IAM Identity Center 使用者指南》中的新增群組。

步驟 2：建立 AWS CodeStar 服務角色

建立 服務角色，用來授予代表您管理 AWS 資源和 IAM 許可的 AWS CodeStar 許可。您只需建立服務角色一次。

> **Important**
>
> 您必須以 管理員使用者 (或根帳戶) 身分登入，才能建立服務角色。如需詳細資訊，請參閱建立您的第一個 IAM 使用者和群組。

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 選擇 Start project (開始專案)。

   如果您沒有看到 Start project (開始專案)，反而被引導到專案清單頁面，表示已建立服務角色。
3. 在建立服務角色頁面上，選擇Yes, create role (是的，建立角色)。
4. 離開精靈。您稍後返回至此。

步驟 3：設定使用者的 IAM 許可

除了管理使用者之外，您還可以使用 AWS CodeStar 做為 IAM 使用者、聯合身分使用者、根使用者或擔任的角色。如需 IAM 使用者與聯合身分使用者 AWS CodeStar 可以執行哪些操作的詳細資訊，請參閱 AWS CodeStar IAM 角色。

如果您尚未設定任何 IAM 使用者，請參閱 IAM 使用者。

若要提供存取權，請新增權限至您的使用者、群組或角色：

- 中的使用者和群組 AWS IAM Identity Center：

  建立權限合集。請按照 AWS IAM Identity Center 使用者指南 中的 建立權限合集 說明進行操作。
- 透過身分提供者在 IAM 中管理的使用者：

  建立聯合身分的角色。請按照 IAM 使用者指南 的 為第三方身分提供者 (聯合) 建立角色 中的指示進行操作。
- IAM 使用者：

- 建立您的使用者可擔任的角色。請按照 IAM 使用者指南 的 為 IAM 使用者建立角色 中的指示進行操作。
- (不建議) 將政策直接附加至使用者，或將使用者新增至使用者群組。請遵循 IAM 使用者指南的新增許可到使用者 (主控台) 中的指示。

步驟 4：建立專案的 AWS CodeStar Amazon EC2 金鑰對

許多 AWS CodeStar 專案會使用 AWS CodeDeploy 或 AWS Elastic Beanstalk 將程式碼部署至 Amazon EC2 執行個體。若要存取與您的專案相關聯的 Amazon EC2 執行個體，請為您的 IAM 使用者建立 Amazon EC2 金鑰對。您的 IAM 使用者必須具有建立和管理 Amazon EC2 金鑰的許可（例如，採取 `ec2:CreateKeyPair`和 `ec2:ImportKeyPair`動作的許可）。如需詳細資訊，請參閱 Amazon EC2 金鑰對。

步驟 5：開啟 AWS CodeStar 主控台

登入 AWS Management Console，然後開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。

後續步驟

恭喜，您完成設定！若要開始使用 AWS CodeStar，請參閱 入門 AWS CodeStar。

入門 AWS CodeStar

在本教學課程中，您會使用 來 AWS CodeStar 建立 Web 應用程式。在來源儲存庫中，此專案包含範本程式碼、持續部署工具鏈和專案儀表板，其中可以讓您檢視和監控您的專案。

遵循以下步驟：

- 在 中建立專案 AWS CodeStar。
- 探索專案。
- 遞交程式碼變更。
- 查看您自動部署的程式碼變更。
- 新增其他使用者來處理您的專案。
- 清理不再需要的專案資源。

> **Note**
>
> 如果尚未完成，首先完成 設定 AWS CodeStar 中的步驟，包括步驟 2：建立 AWS CodeStar 服務角色。您必須使用 IAM 中管理使用者的帳戶登入。若要建立專案，您必須 AWS Management Console 使用具有 **`AWSCodeStarFullAccess`**政策的 IAM 使用者登入 。

主題

- 教學課程：使用 GitHub 來源儲存庫建立專案

步驟 1：建立 AWS CodeStar 專案

在此步驟中，您建立適用於 Web 應用程式的 JavaScript (Node.js) 軟體開發專案。您可以使用 AWS CodeStar 專案範本來建立專案。

> **Note**
>
> 本教學中使用的 AWS CodeStar 專案範本使用以下選項：
>
> - 應用程式類別：Web 應用程式
> - 程式設計語言：Node.js
> - AWS 服務：Amazon EC2
>
> 如果您選擇其他選項，您的體驗可能不會符合此教學課程中的記錄。

在 中建立專案 AWS CodeStar

1. 登入 AWS Management Console，然後開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。

   請確定您已登入要建立專案及其資源的 AWS 區域。例如，若要在美國東部（俄亥俄）建立專案，請確定您已選取該 AWS 區域。如需 AWS CodeStar 可用 AWS 區域的相關資訊，請參閱《AWS 一般參考》中的區域和端點。
2. 在頁面上AWS CodeStar，選擇建立專案。
3. 在選擇專案範本頁面上，從專案範本清單中選擇 AWS CodeStar 專案類型。您可使用篩選條件搜尋列，以縮減選項。例如，對於以 Node.js 撰寫的 Web 應用程式專案部署到 Amazon EC2 執行個體，請選取 Web 應用程式、Node.js 和 Amazon EC2 核取方塊。然後從符合這組選項的範本中選擇。

   如需詳細資訊，請參閱AWS CodeStar 專案範本。
4. 選擇 Next (下一步)。
5. 在專案名稱文字輸入欄位中，輸入專案的名稱，例如#######。在專案 ID 中，專案的 ID 衍生自此專案名稱，但限制為 15 個字元。

例如，名為 *My First Project* 之專案的預設 ID 為 *my-first-projec*。此專案 ID 是與專案相關聯的所有資源名稱的基礎。 AWS CodeStar 會使用此專案 ID 做為程式碼儲存庫 URL 的一部分，以及 IAM 中相關安全存取角色和政策的名稱。專案建立之後，就無法再變更其 ID。若要在建立專案之前編輯專案 ID，請在專案 ID 中輸入您要使用的 ID。

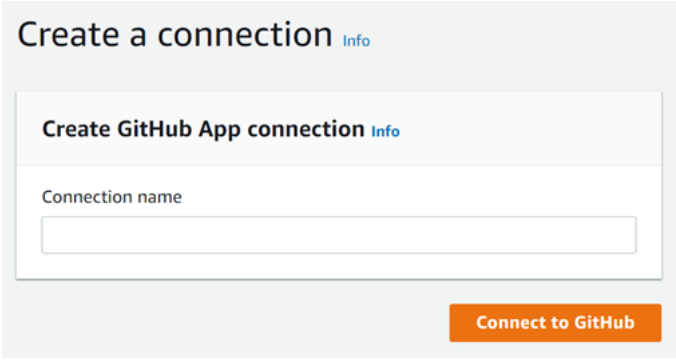
如需專案名稱和專案 ID 限制的資訊，請參閱 中的限制 AWS CodeStar。

> **Note**
>
> 您 AWS 區域中 AWS 的帳戶的專案 IDs 必須是唯一的。

6. 選擇儲存庫提供者、AWS CodeCommit 或 GitHub (GitHub)。
7. 如果您選擇 AWS CodeCommit，請針對儲存庫名稱接受預設 AWS CodeCommit 儲存庫名稱，或輸入不同的名稱。然後跳到步驟 9。
8. 如果您選擇 GitHub，則需要選擇或建立連線資源。如果您有現有的連線，請在搜尋欄位中選擇。否則，請立即建立新的連線。選擇連線到 GitHub。

   隨即顯示建立連線頁面。

   > **Note**
   >
   > 若要建立連線，您必須擁有 GitHub 帳戶。如果您要為組織建立連線，您必須是組織擁有者。



   a. 在建立 GitHub 應用程式連線下，在連線名稱輸入文字欄位中輸入連線的名稱。選擇連線到 GitHub。

      連線至 GitHub 頁面會顯示並顯示 GitHub 應用程式欄位。

b. 在 GitHub Apps (GitHub 應用程式) 底下，選擇應用程式安裝，或選擇 Install a new app (安裝新應用程式) 以建立安裝。

> **Note**
>
> 您可以為您連至特定供應商的所有連線安裝一個應用程式。如果您已安裝 AWS Connector for GitHub 應用程式，請選擇它並略過此步驟。

c. 在安裝 GitHub AWS 連接器頁面上，選擇您要安裝應用程式的帳戶。

> **Note**
>
> 如果您先前已安裝應用程式，可以選擇 Configure (設定)，繼續前往應用程式安裝的修改頁面，或者您可以使用上一步按鈕返回主控台。

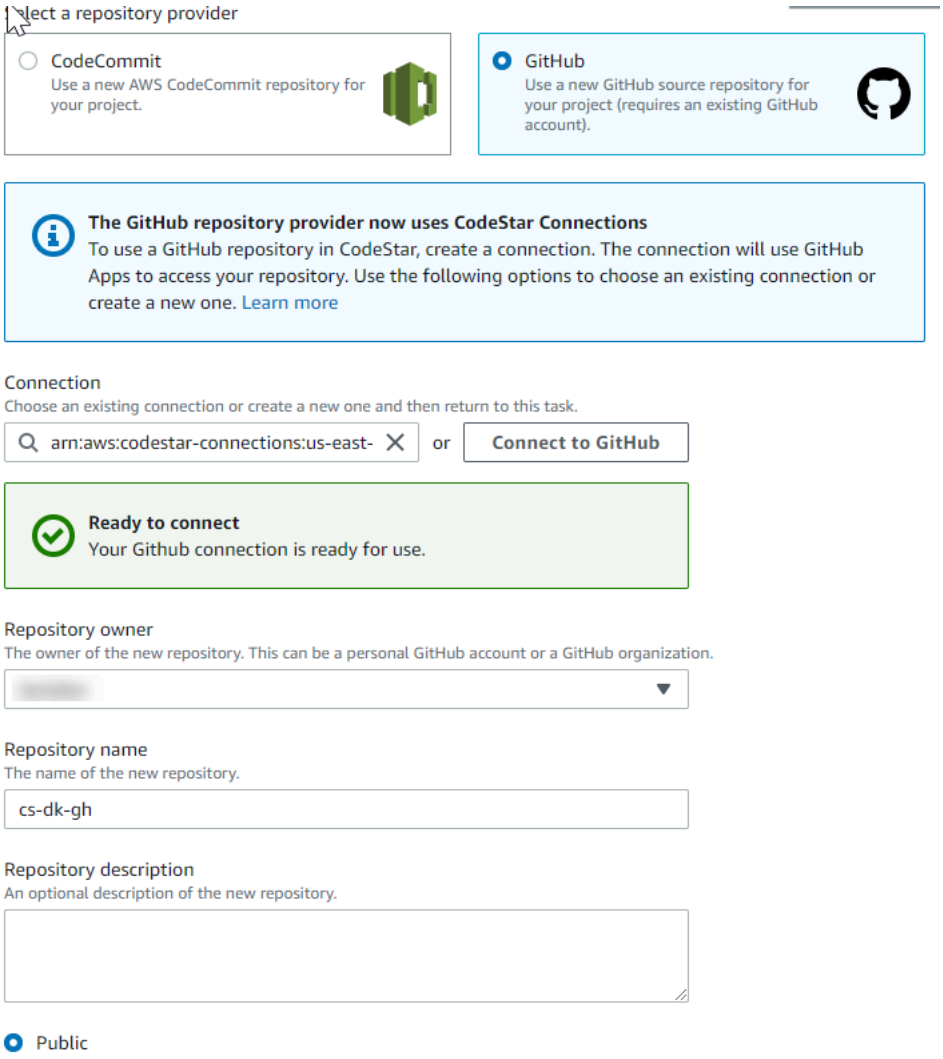
d. 如果顯示確認密碼以繼續頁面，請輸入您的 GitHub 密碼，然後選擇登入。

e. 在安裝 GitHub AWS 連接器頁面上，保留預設值，然後選擇安裝。

f. 在連線至 GitHub 頁面上，新安裝的安裝 ID 會顯示在 GitHub Apps 文字輸入欄位中。

建立連線後，在 CodeStar 建立專案頁面中，即會顯示準備連線訊息。

> **Note**
>
> 您可以在開發人員工具主控台的設定下檢視您的連線。如需詳細資訊，請參閱連線入門。



g. 針對儲存庫擁有者，選擇 GitHub 組織或您的個人 GitHub 帳戶。

h. 在 Repository name (儲存庫名稱) 中，請接受預設 GitHub 儲存庫名稱，或輸入另一個名稱。

i. 選擇公有或私有。

> **Note**
>
> 若要使用 AWS Cloud9 做為您的開發環境，您必須選擇公有。

j. (選用) 在 Repository description (儲存庫描述) 中，輸入此 GitHub 儲存庫的描述。

> **Note**
>
> 如果您選擇 Alexa Skill 專案範本，則需要連接 Amazon 開發人員帳戶。如需使用 Alexa Skill 專案的詳細資訊，請參閱 教學課程：在 CodeStar 中 AWS 建立 Alexa 技能專案。

9. 如果您的專案已部署到 Amazon EC2 執行個體，且您想要進行變更，請在 Amazon EC2 組態中設定 Amazon EC2 執行個體。例如，您可為專案選擇可用的執行個體類型。

> **Note**
>
> 不同的 Amazon EC2 執行個體類型提供不同層級的運算能力，並且可能有不同的相關成本。如需詳細資訊，請參閱 Amazon EC2 執行個體類型和 Amazon EC2 定價。
> 如果您有多個虛擬私有雲端 (VPC) 或在 Amazon Virtual Private Cloud 中建立的多個子網路，您也可以選擇要使用的 VPC 和子網路。不過，如果您選擇專用執行個體上不支援的 Amazon EC2 執行個體類型，則無法選擇執行個體租用設定為專用的 VPC。
> 如需詳細資訊，請參閱什麼是 Amazon VPC？和專用執行個體基本概念。

   在金鑰對中，選擇您在 中建立的 Amazon EC2 金鑰對步驟 4：建立專案的 AWS CodeStar Amazon EC2 金鑰對。選取我確認我有權存取私有金鑰檔案。
10. 選擇 Next (下一步)。
11. 檢閱資源和組態詳細資訊。
12. 選擇 Next (下一步) 或 Create project (建立專案)。(顯示的選項視您的專案範本而定。)

    建立專案可能需要幾分鐘的時間，包括 儲存庫。
13. 專案擁有儲存庫之後，您可以使用儲存庫頁面來設定其存取權。使用後續步驟中的連結來設定 IDE、設定問題追蹤，或將團隊成員新增至您的專案。

步驟 2：為您的 AWS CodeStar 使用者設定檔新增顯示資訊

當您建立專案，您會被加入到專案團隊做為擁有者。如果這是您第一次使用 AWS CodeStar，系統會要求您提供：

- 對其他使用者顯示的您的顯示名稱。
- 對其他使用者顯示的電子郵件地址。

此資訊用於您的 AWS CodeStar 使用者設定檔。使用者設定檔並非專案專屬，但僅限於 AWS 區域。您必須在屬於專案的每個 AWS 區域中建立使用者設定檔。每個設定檔可以包含不同的資訊，依您的喜好。

輸入使用者名稱和電子郵件地址，然後選擇下一步。

> **Note**
>
> 此使用者名稱和電子郵件地址用於您的 AWS CodeStar 使用者設定檔。如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），這些資源提供者可能有自己的使用者設定檔，具有不同的使用者名稱和電子郵件地址。如需詳細資訊，請參閱資源提供者的文件。

步驟 3：檢視您的專案

您的 AWS CodeStar 專案頁面可讓您和團隊檢視專案資源的狀態，包括專案的最新遞交、持續交付管道的狀態，以及執行個體的效能。若要查看這些資源的詳細資訊，請從導覽列中選擇對應的頁面。

在您的新專案中，導覽列包含下列頁面：

- 概觀頁面包含專案活動、專案資源和專案README內容的相關資訊。
- IDE 頁面可讓您將專案連線至整合式開發環境 (IDE)，以修改、測試和推送原始程式碼變更。它包含為 GitHub 和 AWS CodeCommit 儲存庫設定 IDEs 的說明，以及 AWS Cloud9 環境的相關資訊。
- 儲存庫頁面會顯示您的儲存庫詳細資訊，包括名稱、提供者、上次修改的時間，以及複製 URLs。您也可以查看有關最新遞交的資訊，以及檢視和建立提取請求。
- 管道頁面會顯示管道的 CI/CD 資訊。您可以檢視管道詳細資訊，例如名稱、最新動作和狀態。您可以查看管道的歷史記錄並發佈變更。您也可以檢視管道個別步驟的狀態。
- 監控頁面會根據專案的組態顯示 Amazon EC2 或 AWS Lambda 指標。例如，它會顯示管道中由 AWS Elastic Beanstalk 或 CodeDeploy 資源部署到 的任何 Amazon EC2 執行個體的 CPU 使用率。在使用 的專案中 AWS Lambda，它會顯示 Lambda 函數的調用和錯誤指標。按小時顯示此資訊。如果您在本教學課程中使用建議的 AWS CodeStar 專案範本，您應該會在應用程式首次部署到這些執行個體時看到活動明顯激增。您可以重新整理監控以查看您的執行個體運作狀態的變化，其可協助您找出問題或更多資源的需要。
- 問題頁面用於將您的 AWS CodeStar 專案與 Atlassian JIRA 專案整合。設定此圖磚可讓您和您的專案團隊從專案儀表板追蹤 JIRA 問題。

主控台左側的導覽窗格可讓您在專案、團隊和設定頁面之間導覽。

步驟 4：遞交變更

首先，查看專案中包含的範例應用程式。從您的專案導覽中的任何位置選擇檢視應用程式，以查看應用程式的外觀。您的範例 Web 應用程式會顯示在新的視窗或瀏覽器索引標籤中。這是 AWS CodeStar 建置和部署的專案範例。

如果您想要查看程式碼，請在導覽列中選擇儲存庫。選擇儲存庫名稱下的連結，專案的儲存庫會在新索引標籤或視窗中開啟。閱讀儲存庫的 readme 檔案內容 (`README.md`)，並瀏覽這些檔案的內容。

在此步驟中，您變更程式碼，然後推送變更至儲存庫。您可以數種不同方式的其中一種來執行：

- 如果專案的程式碼存放在 CodeCommit 或 GitHub 儲存庫中，您可以使用 直接從 Web 瀏覽器 AWS Cloud9 使用程式碼，而無需安裝任何工具。如需詳細資訊，請參閱建立專案 AWS Cloud9 的環境。
- 如果專案的程式碼存放在 CodeCommit 儲存庫中，且您已安裝 Visual Studio 或 Eclipse，您可以使用 AWS Toolkit for Visual Studio 或 AWS Toolkit for Eclipse 更輕鬆地連線至程式碼。如需詳細資訊，請參閱搭配 使用 IDE AWS CodeStar。如果您沒有 Visual Studio 或 Eclipse，請安裝 Git 用戶端，並依照此步驟稍後的說明操作。
- 如果專案的程式碼是存放在 GitHub 儲存庫，您可以使用 IDE 的工具連接到 GitHub。
  - 針對 Visual Studio，您可以使用 GitHub Extension for Visual Studio 等工具。如需詳細資訊，請參閱 GitHub Extension for Visual Studio 網站上的概觀頁面，以及 GitHub 網站上的 GitHub for Visual Studio 入門。
  - 對於 Eclipse，您可以使用如 EGit for Eclipse 之類的工具。如需詳細資訊，請參閱 EGit 網站上的 EGit 文件。
  - 關於其他 IDE，請參閱 IDE 的文件。
- 關於其他類型的程式碼儲存庫的詳細資訊，請參閱儲存庫提供者的文件。

以下指示說明如何對範例做次要變更。

設定您的電腦以確認變更 (IAM 使用者)

> **Note**
>
> 在此程序中，我們假設專案的程式碼存放在 CodeCommit 儲存庫中。關於其他類型的程式碼儲存庫的詳細資訊，請參閱儲存庫提供者的文件，然後請直接跳到下一個程序：複製專案儲存庫並做變更。

如果程式碼存放在 CodeCommit 中，且您已使用 CodeCommit 或您已使用 AWS CodeStar 主控台為專案建立 AWS Cloud9 開發環境，則不需要更多組態。跳到下一個程序：複製專案儲存庫並做變更。

1. 安裝 Git 於您的本機電腦。
2. 登入 AWS Management Console ，並在 https://console.aws.amazon.com/iam/ ：// 開啟 IAM 主控台。

   以 IAM 使用者身分登入，該使用者將使用 Git 登入資料連線至 CodeCommit 中的 AWS CodeStar 專案儲存庫。
3. 在 IAM 主控台的導覽窗格中，選擇使用者，然後從使用者清單中選擇您的 IAM 使用者。
4. 在使用者詳細資訊頁面上，選擇安全登入資料索引標籤，然後在 CodeCommit 的 HTTPS Git 登入資料中選擇產生。

   > **Note**
   >
   > 您無法為 Git 登入資料選擇自己的登入憑證。如需詳細資訊，請參閱使用 Git 登入資料和 HTTPS 搭配 CodeCommit。
5. 複製 IAM 為您產生的登入憑證。您可以選擇顯示，然後複製此資訊並貼至本機電腦的安全檔案，或者您可以選擇下載登入資料下載此資訊為 .CSV 檔案。您需要此資訊才能連線至 CodeCommit。

   儲存您的登入資料之後，選擇 Close (關閉)。

   > **Important**
   >
   > 這是您儲存登入資料的唯一機會。如果您不儲存這些名稱，您可以從 IAM 主控台複製使用者名稱，但無法查詢密碼。您必須重設密碼，然後儲存密碼。

設定您的電腦以確認變更 (聯合身分使用者)

您可以使用主控台來上傳檔案到您的儲存庫，或者使用 Git 從您的本機電腦連線。如果您使用的是聯合身分存取權，請依照以下步驟以使用 Git 來連線，並從您的本機電腦複製儲存庫。

> **Note**
>
> 在此程序中，我們假設專案的程式碼存放在 CodeCommit 儲存庫中。關於其他類型的程式碼儲存庫的詳細資訊，請參閱儲存庫提供者的文件，然後請直接跳到下一個程序：複製專案儲存庫並做變更。

1. 安裝 Git 於您的本機電腦。
2. 安裝 AWS CLI。
3. 針對聯合身分使用者設定臨時安全性登入資料。如需詳細資訊，請參閱暫時存取 CodeCommit 儲存庫。臨時登入資料包含：

   - AWS 存取金鑰
   - AWS 私密金鑰
   - 工作階段字符

   如需臨時登入資料的詳細資訊，請參閱 GetFederationToken 的許可。
4. 使用 AWS CLI 登入資料協助程式連線至您的儲存庫。如需詳細資訊，請參閱使用 AWS CLI 登入資料協助程式在 Linux、macOS 或 Unix 上設定 HTTPS 連線至 CodeCommit 儲存庫的步驟，或使用 CLI 登入資料協助程式在 Windows AWS 上設定 HTTPS 連線至 CodeCommit 儲存庫的步驟
5. 下列範例顯示如何連線至 CodeCommit 儲存庫，並將遞交推送至該儲存庫。

範例：複製專案儲存庫並做變更

> **Note**
>
> 此程序說明如何複製專案的程式碼儲存庫到您的電腦、變更專案的 `index.html` 檔案，然後推送您的變更到遠端儲存庫。在此程序中，我們假設專案的程式碼存放在 CodeCommit 儲存庫中，而且您使用來自命令列的 Git 用戶端。如需其他類型的程式碼儲存庫或工具的詳細資訊，請參閱供應商的文件，以了解如何複製儲存庫、變更檔案，然後推送程式碼。

1. 如果您使用 AWS CodeStar 主控台為專案建立 AWS Cloud9 開發環境，請開啟開發環境，然後跳至此程序的步驟 3。若要開啟開發環境的詳細資訊，請參閱開啟專案 AWS Cloud9 的環境。

在 AWS CodeStar 主控台中開啟您的專案時，在導覽列上，選擇儲存庫。在複製 URL 中，選擇您已為 CodeCommit 設定的連線類型的通訊協定，然後複製連結。例如，如果您遵循先前程序中的步驟來設定 CodeCommit 的 Git 登入資料，請選擇 HTTPS。

2. 在您的本機電腦上開啟終端機或命令列視窗，將目錄變更到臨時目錄。執行 git clone 命令以複製儲存庫到您的電腦。貼上您複製的連結。例如，對於使用 HTTPS 的 CodeCommit：

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

第一次連線時，系統會提示您輸入儲存庫的登入憑證。針對 CodeCommit，輸入您在上一個程序中下載的 Git 登入資料登入資料。

3. 導覽到電腦上的複製目錄，並瀏覽內容。

4. 開啟 `index.html` 檔案 (在公有資料夾中)，然後對該檔案進行變更。例如，在 <H2> 標籤後新增段落，例如：

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

儲存檔案。

5. 在終端機或命令提示字元處新增您變更的檔案，然後遞交及推送您的變更：

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

6. 在儲存庫頁面上，檢視進行中的變更。您應該會看到儲存庫的遞交歷史記錄更新您的遞交，包括遞交訊息。在管道頁面中，您可以看到管道取得對儲存庫的變更，並開始建置和部署。部署 Web 應用程式之後，您可以選擇檢視應用程式以檢視變更。

> **Note**
>
> 如果任何管道階段顯示失敗，請參閱以下的故障診斷說明：
>
> - 如需來源階段，請參閱AWS CodeCommit 《使用者指南》中的故障診斷 AWS CodeCommit。
> - 如需建置階段，請參閱AWS CodeBuild 《使用者指南》中的故障診斷 AWS CodeBuild。

- 如需部署階段，請參閱AWS CloudFormation 《使用者指南》中的故障診斷 AWS CloudFormation。
- 有關其他問題，請參閱故障診斷 AWS CodeStar。

步驟 5：新增更多團隊成員

每個 AWS CodeStar 專案都已設定三個 AWS CodeStar 角色。每個角色提供自己的專案及其資源的存取層級：

- 擁有者：可新增和移除團隊成員、變更專案儀表板，以及刪除專案。
- 貢獻者：如果程式碼存放在 CodeCommit 中，但無法新增或移除團隊成員或刪除專案，則可以變更專案儀表板並貢獻程式碼。這是您應該為 AWS CodeStar 專案中的大多數團隊成員選擇的角色。
- 檢視器：如果程式碼存放在 CodeCommit 中，且專案狀態為 ，但無法從專案儀表板移動、新增或移除圖磚，則可以檢視專案儀表板、專案程式碼。

> **Important**
>
> 如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），則這些資源的存取是由資源提供者控制，而不是由資源提供者控制 AWS CodeStar。如需詳細資訊，請參閱資源提供者的文件。
>
> 有權存取 AWS CodeStar 專案的任何人，都可以使用 AWS CodeStar 主控台來存取 外部 AWS 但與專案相關的資源。
>
> AWS CodeStar 不允許專案團隊成員參與專案的任何相關 AWS Cloud9 開發環境。若要允許團隊成員參與共享的環境，請參閱與專案團隊成員共用 AWS Cloud9 環境。

如需有關團隊與專案角色的詳細資訊，請參閱使用 AWS CodeStar 團隊。

將團隊成員新增至 AWS CodeStar 專案 （主控台）

1. 開啟 AWS CodeStar 主控台，網址為 http：//https://console.aws.amazon.com/codestar/。
2. 從導覽窗格中選擇專案，然後選擇您的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。

4. 在 Team members (團隊成員) 頁面上，選擇 Add team member (新增團隊成員)。
5. 在 Choose user (選擇使用者) 中，執行下列其中一項操作：

   - 如果您要新增的人員已有 IAM 使用者，請從清單中選擇 IAM 使用者。

     > **Note**
     >
     > 已新增至另一個 AWS CodeStar 專案的使用者會出現在現有 AWS CodeStar 使用者清單中。

     在專案角色中，為此使用者選擇 AWS CodeStar 角色（擁有者、貢獻者或檢視器）。這屬於 AWS CodeStar 專案層級角色，唯有專案的擁有者能進行變更。套用至 IAM 使用者時，該角色會提供存取 AWS CodeStar 專案資源所需的所有許可。它會套用為存放在 IAM 中的 CodeCommit 中的程式碼建立和管理 Git 登入資料，或為 IAM 中的使用者上傳 Amazon EC2 SSH 金鑰所需的政策。

     > **Important**
     >
     > 除非您以該使用者身分登入主控台，否則您無法提供或變更 IAM 使用者的顯示名稱或電子郵件資訊。如需詳細資訊，請參閱管理 AWS CodeStar 使用者設定檔的顯示資訊 。

     選擇新增團隊成員。

   - 如果您要新增至專案的人員沒有 IAM 使用者，請選擇建立新的 IAM 使用者。系統會將您重新導向至 IAM 主控台，您可以在其中建立新的 IAM 使用者，如需詳細資訊，請參閱《IAM 使用者指南》中的建立 IAM 使用者。建立 IAM 使用者後，返回 AWS CodeStar 主控台，重新整理使用者清單，然後從下拉式清單中選擇您建立的 IAM 使用者。輸入您要套用至此新使用者的 AWS CodeStar顯示名稱、電子郵件地址和專案角色，然後選擇新增團隊成員。

     > **Note**
     >
     > 為了方便管理，您應該將專案的 Owner (擁有者) 角色指派給至少一個使用者。

6. 傳送下列資訊給新的團隊成員：

   - AWS CodeStar 專案的連線資訊。

- 如果原始程式碼存放在 CodeCommit 中，則使用 Git 登入資料設定從其本機電腦存取 CodeCommit 儲存庫的指示。 CodeCommit
- 有關使用者如何管理其顯示名稱、電子郵件地址和公有 Amazon EC2 SSH 金鑰的資訊，如 中所述使用 AWS CodeStar 使用者設定檔 。
- 一次性密碼和連線資訊，如果使用者是初次使用 AWS ，而且您為該使用者建立了 IAM 使用者。此密碼會在使用者首次登入後過期，因此使用者必須選擇新的密碼。

步驟 6：清除

恭喜您！您已完成教學課程。如果您不想繼續使用此專案及其資源，您應該將其刪除，以避免 AWS 您的帳戶可能繼續產生費用。

在 中刪除專案 AWS CodeStar

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 在導覽窗格中選擇專案。
3. 選取您要刪除的專案，然後選擇刪除。

   或者，開啟專案，然後從主控台左側的導覽窗格中選擇設定。在專案詳細資訊頁面上，選擇 Delete project (刪除專案)。
4. 在刪除確認頁面中，輸入刪除。如果您想要刪除專案資源，請保持選取刪除資源。選擇 刪除 。

   刪除專案可能需要幾分鐘的時間。刪除後，專案不會再出現在 AWS CodeStar 主控台的專案清單中。

> **Important**
>
> 如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），即使您選取核取方塊，也不會刪除這些資源。
> 如果任何 AWS CodeStar 受管政策已手動連接至非 IAM 使用者的角色，則無法刪除您的專案。在專案受管政策是連接至聯合身分使用者角色的情況下，您必須先分離該政策，才能刪除專案。如需詳細資訊，請參閱???。

步驟 7：讓您的專案準備好進行生產環境

在建立專案後，您可以隨時建立、測試和部署程式碼。檢閱下列有關維護生產環境中的專案考量：

- 定期套用修補程式，並針對應用程式所使用的依存項目，檢閱其安全最佳實務。如需詳細資訊，請參閱AWS CodeStar 資源的安全最佳實務。
- 定期監控您的專案程式設計語言所建議的環境。

後續步驟

以下是一些其他資源，可協助您了解 AWS CodeStar：

- 教學課程：在 AWS CodeStar中建立和管理無伺服器專案 使用在 中使用邏輯建立和部署 Web 服務的專案， AWS Lambda 並且可由 Amazon API Gateway 中的 API 呼叫。
- AWS CodeStar 專案範本 描述您可以建立的其他類型專案。
- 使用 AWS CodeStar 團隊 提供有關讓其他人協助您處理專案的相關資訊。

教學課程：在 AWS CodeStar中建立和管理無伺服器專案

在本教學課程中，您會使用 AWS CodeStar 來建立專案，該專案使用無 AWS 伺服器應用程式模型 (AWS SAM) 來建立和管理託管於 的 Web 服務 AWS 的資源 AWS Lambda。

AWS CodeStar 使用依賴 AWS SAM AWS CloudFormation來提供建立和管理支援 AWS 資源的簡化方式，包括 Amazon API Gateway APIs、 AWS Lambda 函數和 Amazon DynamoDB 資料表。（此專案不會使用任何 Amazon DynamoDB 資料表。）

如需詳細資訊，請參閱 GitHub 上的AWS 無伺服器應用程式模型 (AWS SAM)。

必要條件： 完成設定 AWS CodeStar中的步驟。

> **Note**
>
> AWS 您的帳戶可能需要支付與本教學課程相關的費用，包括 使用之 AWS 服務的費用 AWS CodeStar。如需詳細資訊，請參閱 AWS CodeStar 定價。

主題

概觀

您在此教學課程中：

1. 使用 AWS CodeStar 建立使用 AWS SAM 建置和部署 Python 型 Web 服務的專案。此 Web 服務託管在 中， AWS Lambda 可透過 Amazon API Gateway 存取。
2. 探索專案的主要資源，包括：
   - 存放專案原始碼的儲存 AWS CodeCommit 庫。這原始碼包括 Web 服務的邏輯，和定義相關的 AWS 資源。
   - 自動化建置原始程式碼的 AWS CodePipeline 管道。此管道使用 AWS SAM 來建立和部署函數至 AWS Lambda、在 Amazon API Gateway 中建立相關的 API，以及將 API 連接至函數。
   - 部署至 的 函數 AWS Lambda。
   - 在 Amazon API Gateway 中建立的 API。
3. 測試 Web 服務，確認 已如預期 AWS CodeStar 建置並部署 Web 服務。
4. 設定您的本機工作站以使用專案的原始碼。
5. 利用您的本機工作站變更專案的原始碼。當您新增函數到專案時，然後推送您的變更至原始碼時，AWS CodeStar 會重建和重新部署 Web 服務。
6. 再次測試 Web 服務，以確認 如預期 AWS CodeStar 重建和重新部署。
7. 使用本機工作站編寫單元測試，將一些手動測試取代為自動化測試。當您推送單元測試時， 會 AWS CodeStar 重建並重新部署 Web 服務，並執行單元測試。
8. 檢視單元測試的結果。
9. 清理專案。此步驟可協助您避免 AWS 對您的帳戶收取與本教學課程相關的費用。

步驟 1：建立專案

在此步驟中，您會使用 AWS CodeStar 主控台來建立專案。

1. 登入 AWS Management Console 並開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。

   > **Note**
   >
   > 您必須 AWS Management Console 使用與您在 中建立或識別的 IAM 使用者相關聯的登入資料來登入 設定 AWS CodeStar。這個使用者必須連接 **AWSCodeStarFullAccess** 受管政策。

2. 選擇您要建立專案及其資源 AWS 的區域。

   如需 AWS CodeStar 可用 AWS 區域的相關資訊，請參閱《 AWS 一般參考》中的區域和端點。

3. 選擇建立專案。
4. 在 Choose a project template (選擇專案範本) 頁面：

   - 針對應用程式類型，選取 Web 服務。
   - 針對程式設計語言，選取 Python。
   - 針對AWS 服務，選取 AWS Lambda。

5. 選擇包含您的選取項目的方塊。選擇 Next (下一步)。
6. 在 Project name (專案名稱) 中，輸入專案的名稱 (如 **My SAM Project**)。如果您使用不同於範例的名稱，請務必在本教學課程中加以使用。

   針對專案 ID， 會為此專案 AWS CodeStar 選擇相關的識別符 （ 例如 my-sam-project)。如果您看到不同的專案 ID，請在此教學課程中都使用此名稱。

   保留所選的 AWS CodeCommit，不要變更儲存庫名稱值。

7. 選擇 Next (下一步)。
8. 檢閱您的設定，然後選擇建立專案。

   如果這是您第一次 AWS CodeStar 在此 AWS 區域中使用 ，請在顯示名稱和電子郵件中輸入 AWS CodeStar 您要用於 IAM 使用者的顯示名稱和電子郵件地址。選擇 Next (下一步)。

9. 等待 AWS CodeStar 建立專案。這可能需要幾分鐘的時間。在重新整理時看到專案佈建橫幅之前，請勿繼續。

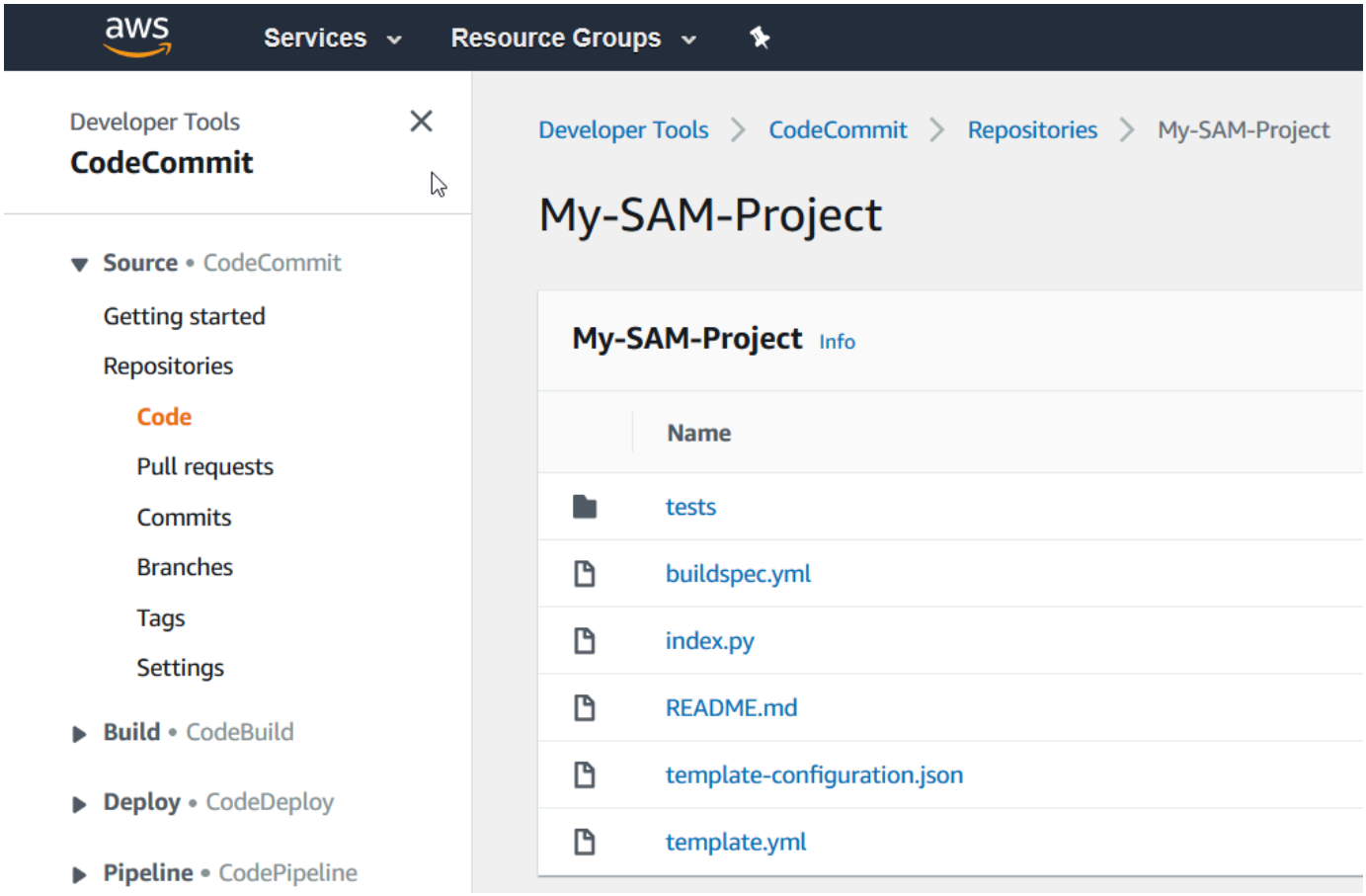
步驟 2：探索專案資源

在此步驟中，您將探索專案的四個 AWS 資源，以了解專案的運作方式：

- 存放專案原始碼的 AWS CodeCommit 儲存庫。 AWS CodeStar 會提供儲存庫名稱 my-sam-project，其中 my-sam-project 是專案的名稱。
- 使用 CodeBuild 和 SAM AWS 在 API Gateway 中自動化建置和部署 Web 服務的 Lambda 函數和 API 的 AWS CodePipeline 管道。 AWS CodeStar gives 管道名稱為 my-sam-project--Pipeline，其中 my-sam-project 是專案的 ID。
- 包含 web service. AWS CodeStar gives 邏輯的 Lambda 函數名稱為 awscodestar-my-sam-project-lambda-HelloWorld-***RANDOM_ID***，其中：
  - my-sam-project 是專案的 ID。
  - HelloWorld 是儲存 AWS CodeCommit 庫中 `template.yaml` 檔案指定的函數 ID。您稍後會探索這個檔案。
  - *RANDOM_ID* 是 SAM AWS 指派給函數的隨機 ID，以協助確保唯一性。
- API Gateway 中的 API 可讓您更輕鬆地呼叫 Lambda 函數。 AWS CodeStar gives API 的名稱為 awscodestar-my-sam-project--lambda，其中 my-sam-project 是專案的 ID。

探索 CodeCommit 中的原始程式碼儲存庫

1. 在 AWS CodeStar 主控台中開啟您的專案時，在導覽列上，選擇儲存庫。
2. 在儲存庫詳細資訊中選擇 CodeCommit 儲存庫 (**`My-SAM-Project`**) 的連結。
3. 在 CodeCommit 主控台的程式碼頁面上，會顯示專案的原始程式碼檔案：
   - `buildspec.yml`CodePipeline 指示 CodeBuild 在建置階段使用 ，以使用 AWS SAM 封裝 Web 服務。
   - `index.py`，其中包含 Lambda 函數的邏輯。此函數只會以 ISO 格式輸出字串 `Hello World` 和時間戳記。
   - `README.md`，其中包含有關儲存庫的一般資訊。
   - `template-configuration.json`，其中包含的專案 ARN，內含標記具有專案 ID 之資源所用的佔位符
   - `template.yml`，SAM AWS 會使用 來封裝 Web 服務，並在 API Gateway 中建立 API。



若要檢視檔案的內容，從清單中選擇檔案。

如需使用 CodeCommit 主控台的詳細資訊，請參閱 AWS CodeCommit 使用者指南。

在 CodePipeline 中探索管道

1. 若要檢視管道的相關資訊，在主控台中 AWS CodeStar 開啟您的專案時，在導覽列上選擇管道，您會看到管道包含：
   - 從 CodeCommit 取得原始程式碼的來源階段。
   - 使用 CodeBuild 建置原始程式碼的建置階段。
   - 部署階段，用於使用 AWS SAM 部署建置的原始程式碼 AWS 和資源。
2. 若要檢視管道的詳細資訊，請在管道詳細資訊中，選擇管道以在 CodePipeline 主控台中開啟管道。

如需有關使用 CodePipeline 主控台的資訊，請參閱 AWS CodePipeline 使用者指南。

在概觀頁面上探索專案活動 AWS 和服務資源

1. 在 AWS CodeStar 主控台中開啟專案，然後從導覽列中選擇概觀。
2. 檢閱專案活動和專案資源清單。

在 Lambda 中探索函數

1. 在 AWS CodeStar 主控台中開啟您的專案時，在側邊導覽列上，選擇概觀。
2. 在專案資源的 ARN 欄中，選擇 Lambda 函數的連結。

   函數的程式碼會顯示在 Lambda 主控台中。

如需有關使用 Lambda 主控台的資訊，請參閱 AWS Lambda 開發人員指南。

在 API Gateway 中探索 API

1. 在 AWS CodeStar 主控台中開啟您的專案時，在側邊導覽列上，選擇概觀。
2. 在專案資源的 ARN 欄中，選擇 Amazon API Gateway API 的連結。

   API 的資源會顯示在 API Gateway 主控台中。

如需有關使用 API Gateway 主控台的資訊，請參閱 API Gateway 開發人員指南。

步驟 3：測試 Web 服務

在此步驟中，您會測試 AWS CodeStar 剛建置和部署的 Web 服務。

1. 在專案仍從上一個步驟開啟的情況下，在導覽列上，選擇管道。
2. 在繼續之前，請確定來源、建置和部署階段顯示成功。這可能需要幾分鐘的時間。

> **Note**
>
> 如果任何階段均顯示失敗，請參閱以下的故障診斷說明：
>
> - 如需來源階段，請參閱AWS CodeCommit 《 使用者指南》中的故障診斷 AWS CodeCommit。

- 如需建置階段，請參閱AWS CodeBuild 《 使用者指南》中的故障診斷 AWS CodeBuild。
- 如需部署階段，請參閱AWS CloudFormation 《 使用者指南》中的故障診斷 AWS CloudFormation。
- 有關其他問題，請參閱故障診斷 AWS CodeStar。

3. 選擇檢視應用程式。

在 Web 瀏覽器中開啟的新標籤上面，Web 服務會顯示以下回應輸出：

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

步驟 4：設定您的本機工作站以編輯專案程式碼

在此步驟中，您設定本機工作站以在 AWS CodeStar 專案中編輯原始碼。您的本機工作站可以是執行 macOS、Windows 或 Linux 的實體或虛擬電腦。

1. 在專案仍在之前的步驟中開啟的情況下：
   - 在導覽列中，選擇 IDE，然後展開存取您的專案程式碼。
   - 選擇命令列界面下方的檢視指示。

     如果您已安裝 Visual Studio 或 Eclipse，請改為選擇 Visual Studio 或 Eclipse 下方的檢視指示，依照指示操作，然後跳至 步驟 5：新增邏輯到 Web 服務。
2. 遵循指示完成以下任務：
   a. 在您的本機工作站設定 Git。
   b. 使用 IAM 主控台為您的 IAM 使用者產生 Git 登入資料。
   c. 將專案的 CodeCommit 儲存庫複製到本機工作站。
3. 在左側導覽中，選擇專案以返回您的專案概觀。

步驟 5：新增邏輯到 Web 服務

在此步驟中，您使用本機工作站新增邏輯到 Web 服務。具體而言，您可以新增 Lambda 函數，然後將其連接到 API Gateway 中的 API。

1. 在本機工作站上，移至包含複製原始碼儲存庫的目錄。
2. 在該目錄中，建立名為 `hello.py` 的檔案。新增下列程式碼，然後儲存檔案：

```
import json

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

   前述程式碼會輸出字串 `Hello` 以及發起人傳送到函數的字串。

3. 在相同目錄中，開啟 `template.yml` 檔案。將下列程式碼新增至檔案結尾，然後儲存檔案：

```
  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 使用此程式碼在 Lambda 中建立函數，在 API Gateway 中新增 API 的新方法和路徑，然後將此方法和路徑連接到新函數。

> **Note**
>
> 前述程式碼的縮排是很重要的。如果您不將程式碼如實完全顯示，專案可能無法正確建置。

4. 執行 git add . 以將您的檔案變更加入到模擬儲存庫之暫存區域。不要忘記期間 (.)，它會新增所有已變更的檔案。

> **Note**
>
> 如果您使用 Visual Studio 或 Eclipse，而不是命令列，使用 Git 的說明可能會不同。請參閱 Visual Studio 或 Eclipse 文件。

5. 執行 git commit -m "Added hello.py and updated template.yaml." 以遞交複製儲存庫中的暫存檔案
6. 執行 git push 以將您的遞交推送到遠端儲存庫。

> **Note**
>
> 系統可能會提示您輸入稍早為您產生的登入憑證。為了避免您每次與遠端儲存庫互動時系統都出現提示，請考慮安裝和設定 Git 登入資料管理工具。例如，在 macOS 或 Linux 上，您可以在終端機執行 git config credential.helper 'cache --timeout 900'，提示的間隔不短於 15 分鐘。或者，您可以執行 git config credential.helper 'store --file ~/.git-credentials'，系統不會提示您再輸入一次。Git 將您的登入資料以明文存放在主目錄中的純文字檔案。如需詳細資訊，請參閱 Git 網站上的 Git Tools - Credential Storage。

在 AWS CodeStar 偵測到推送後，它會指示 CodePipeline 使用 CodeBuild 和 AWS SAM 來重建和重新部署 Web 服務。您可以在管道頁面上觀看部署進度。

AWS SAM 為新函數命名為 awscodestar-my-sam-project-lambda-Hello-*RANDOM_ID*，其中：

- my-sam-project 是專案的 ID。
- Hello 是函數 ID，如 `template.yaml` 檔案之指定。
- *RANDOM_ID* 是 SAM AWS 為唯一性指派給函數的隨機 ID。

步驟 6：測試增強的 Web 服務

在此步驟中，您會根據在上一個步驟中新增的邏輯，測試 AWS CodeStar 建置和部署的增強型 Web 服務。

1. 在 AWS CodeStar 主控台中仍開啟您的專案時，在導覽列上選擇管道。
2. 請確定管道已再次執行，而且在繼續之前，來源、建置和部署階段會顯示成功。這可能需要幾分鐘的時間。

   > **Note**
   >
   > 如果任何階段均顯示失敗，請參閱以下的故障診斷說明：
   >
   > - 如需來源階段，請參閱AWS CodeCommit 《 使用者指南》中的故障診斷 AWS CodeCommit。
   > - 如需建置階段，請參閱AWS CodeBuild 《 使用者指南》中的故障診斷 AWS CodeBuild。
   > - 如需部署階段，請參閱AWS CloudFormation 《 使用者指南》中的故障診斷 AWS CloudFormation。
   > - 有關其他問題，請參閱故障診斷 AWS CodeStar。

3. 選擇檢視應用程式。

   在 Web 瀏覽器中開啟的新標籤上面，Web 服務會顯示以下回應輸出：

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

4. 在標籤的地址方塊，新增路徑 **/hello/** 和您的名字至 URL 的尾端 (例如，https://***API_ID***.execute-api.***REGION_ID***.amazonaws.com/Prod/hello/***YOUR_FIRST_NAME***)，然後按 Enter。

如果您的名字是 Mary，Web 服務會顯示下列回應輸出：

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

步驟 7：新增單元測試到 Web 服務

在此步驟中，您會使用本機工作站來新增在 Web 服務 AWS CodeStar 上執行的測試。此測試會取代您稍早所做的手動測試。

1. 在本機工作站上，移至包含複製原始碼儲存庫的目錄。
2. 在該目錄中，建立名為 `hello_test.py` 的檔案。新增下列程式碼，然後儲存檔案。

```
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
```

此測試會檢查 Lambda 函數的輸出是否為預期的格式。若是，則測試成功。否則，測試失敗。

3. 在相同目錄中，開啟 `buildspec.yml` 檔案。將檔案的內容取代為下列程式碼，然後儲存檔案。

```
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
```

此建置規格會指示 CodeBuild 在其建置環境中安裝 Python 測試架構 pytest。CodeBuild 使用 pytest 來執行單元測試。建置規格的其他部分同前。

4. 使用 Git 將這些變更推送到遠端儲存庫。

```
git add .

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

git push
```

步驟 8：檢視單元測試結果

在此步驟中，您查看單元測試是否成功或失敗。

1. 在 AWS CodeStar 主控台中仍開啟您的專案時，在導覽列上，選擇管道。
2. 在繼續之前，請確定管道已再次執行。這可能需要幾分鐘的時間。

   如果單元測試成功，則組建階段會顯示成功。
3. 若要檢視單位測試結果詳細資訊，請在建置階段中，選擇 CodeBuild 連結。
4. 在 CodeBuild 主控台的建置專案：my-sam-project 頁面上，在建置歷史記錄中，選擇資料表建置執行欄中的連結。
5. 在 my-sam-project:***BUILD_ID*** 頁面，在 Build logs (建置日誌) 中選擇 View entire log (查看整個日誌) 連結。
6. 在 Amazon CloudWatch Logs 主控台中，查看日誌輸出是否有類似下列的測試結果。在以下測試結果中，測試已通過：

```
...
============================= 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 ===========================
...
```

   如果測試失敗，日誌輸出中應有詳細資訊來協助您排除障礙。

步驟 9：清除

在此步驟中，您清除專案，以避免此專案持續產生費用。

如果您想要繼續使用此專案，您可以略過此步驟，但 AWS 您的帳戶可能會繼續收費。

1. 在 AWS CodeStar 主控台中仍開啟您的專案時，在導覽列上選擇設定。
2. 在專案詳細資訊中，選擇刪除專案。
3. 輸入 **delete**，保持選取刪除資源方塊，然後選擇刪除。

> **Important**
>
> 如果您清除此方塊，則會從 刪除專案記錄 AWS CodeStar，但會保留專案的許多 AWS 資源。 AWS 您的帳戶可能會繼續收費。

如果仍有為此專案 AWS CodeStar 建立的 Amazon S3 儲存貯體，請依照下列步驟將其刪除。：

1. 開啟 Amazon S3 主控台，網址為 https://console.aws.amazon.com/s3/：//。
2. 在儲存貯體清單中，選擇 aws-codestar-***REGION_ID*-*ACCOUNT_ID***-my-sam-project--pipe 旁的圖示，其中：
   - *REGION_ID* 是您剛刪除之專案的 AWS 區域 ID。
   - *ACCOUNT_ID* 是 AWS 您的帳戶 ID。
   - my-sam-project 是您剛刪除的專案 ID。
3. 選擇清空儲存貯體。輸入儲存貯體的名稱，然後選擇確認。
4. 選擇 Delete Bucket (刪除儲存貯體)。輸入儲存貯體的名稱，然後選擇確認。

後續步驟

現在您已完成這個教學課程，我們建議您檢閱下列資源：

- 本入門 AWS CodeStar教學課程使用 專案來建立和部署在 Amazon EC2 執行個體上執行的 Node.js 型 Web 應用程式。
- AWS CodeStar 專案範本 描述您可以建立的其他類型專案。
- 使用 AWS CodeStar 團隊 說明其他人如何協助您運作您的專案。

教學課程： AWS CodeStar 使用 在 中建立專案 AWS CLI

本教學課程說明如何使用 AWS CLI 建立具有範例原始程式碼和範例工具鏈範本的 AWS CodeStar 專案。 AWS CodeStar 佈建 AWS CloudFormation 工具鏈範本中指定的 AWS 基礎設施和 IAM 資源。專案會管理您的工具鏈資源來建置並部署您的原始程式碼。

AWS CodeStar 使用 AWS CloudFormation 來建置和部署您的範本程式碼。此範例程式碼會建立託管在 中的 Web 服務， AWS Lambda 並且可以透過 Amazon API Gateway 存取。

先決條件：

- 完成「設定 AWS CodeStar」中的步驟。
- 您必須已建立 Amazon S3 儲存貯體。在此教學課程中，您會將範例原始程式碼和工具鏈範本上傳至此位置。

> **Note**
>
> AWS 您的帳戶可能需要支付與本教學課程相關的費用，包括 所使用的 AWS 服務 AWS CodeStar。如需詳細資訊，請參閱 AWS CodeStar 定價。

主題

步驟 1：下載並檢閱範例原始程式碼

此教學課程提供一個 zip 檔案可供下載。其中包含 Lambda 運算平台上的 Node.js 範例應用程式的範例原始程式碼。原始程式碼進入您的儲存庫時，將出現其資料夾和檔案，如下所示：

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

您的範例原始程式碼將呈現下列專案元素：

- `tests/`：為此專案的 CodeBuild 專案設定的單位測試。此資料夾包含在範本程式碼中，但並非建立專案所必須之元素。
- `app.js`：您專案的應用程式原始程式碼。

- `buildspec.yml`：CodeBuild 資源建置階段的建置指示。具有 CodeBuild 資源的工具鏈範本需要此檔案。
- `package.json`：您應用程式原始程式碼的相依性資訊。
- `README.md`：所有 AWS CodeStar 專案均具備的專案 readme 檔案。此檔案包含在範本程式碼中，但並非建立專案所必須之元素。
- `template.yml`：所有 AWS CodeStar 專案中包含的基礎設施範本檔案或 SAM 範本檔案。這與您稍後將於本教學課程上傳的工具鏈 template.yml 不同。此檔案包含在範本程式碼中，但並非建立專案所必須之元素。

步驟 2：下載範例工具鏈範本

本教學課程提供的範例工具鏈範本會建立儲存庫 (CodeCommit)、管道 (CodePipeline) 和建置容器 (CodeBuild)，並使用 AWS CloudFormation 將原始程式碼部署至 Lambda 平台。除了這些資源之外，您還可以使用 IAM 角色來限制執行時間環境的許可範圍、CodePipeline 用來存放部署成品的 Amazon S3 儲存貯體，以及當您將程式碼推送至儲存庫時用來觸發管道部署的 CloudWatch Events 規則。為了符合 AWS IAM 最佳實務，請縮減此範例定義的工具鏈角色政策的範圍。

下載並解壓縮 YAML 格式的範例 AWS CloudFormation 範本。

稍後在此教學課程執行 create-project 命令時，此範本會在 AWS CloudFormation中建立下列自訂工具鏈資源。如需此教學課程中建立的資源詳細資訊，請參閱 AWS CloudFormation 使用者指南中的下列主題：

- AWS::CodeCommit::Repository AWS CloudFormation 資源會建立 CodeCommit 儲存庫。
- AWS::CodeBuild::Project AWS CloudFormation 資源會建立 CodeBuild 組建專案。
- AWS::CodeDeploy::Application AWS CloudFormation 資源會建立 CodeDeploy 應用程式。
- AWS::CodePipeline::Pipeline AWS CloudFormation 資源會建立 CodePipeline 管道。
- AWS::S3::Bucket AWS CloudFormation 資源會建立管道的成品儲存貯體。
- AWS::S3::BucketPolicy AWS CloudFormation 資源會建立管道成品儲存貯體的成品儲存貯體政策。
- AWS::IAM::Role AWS CloudFormation 資源會建立 CodeBuild IAM 工作者角色，提供管理 CodeBuild 組建專案的 AWS CodeStar 許可。
- AWS::IAM::Role AWS CloudFormation 資源會建立 CodePipeline IAM 工作者角色，提供建立管道的 AWS CodeStar 許可。
- AWS::IAM::Role AWS CloudFormation 資源會建立 IAM AWS CloudFormation 工作者角色，提供建立資源堆疊的 AWS CodeStar 許可。

- AWS::IAM::Role AWS CloudFormation 資源會建立 IAM AWS CloudFormation 工作者角色，提供建立資源堆疊的 AWS CodeStar 許可。
- AWS::IAM::Role AWS CloudFormation 資源會建立 IAM AWS CloudFormation 工作者角色，提供建立資源堆疊的 AWS CodeStar 許可。
- AWS::Events::Rule AWS CloudFormation 資源會建立 CloudWatch Events 規則，以監控您的儲存庫是否有推送事件。
- AWS::IAM::Role AWS CloudFormation 資源會建立 CloudWatch Events IAM 角色。

步驟 3：在 中測試您的工具鏈範本 AWS CloudFormation

上傳工具鏈範本前，您可在 AWS CloudFormation 測試工具鏈範本並針對錯誤進行疑難排解。

1. 將更新後的範本儲存至本機電腦，然後開啟 AWS CloudFormation 主控台。選擇 Create Stack (建立堆疊)。您應該會在清單中看到新資源。
2. 檢視堆疊中的堆疊建立錯誤。
3. 測試完成後，請刪除堆疊。

> **Note**
>
> 請務必刪除堆疊和在其中建立的所有資源 AWS CloudFormation。否則，建立專案時，可能會出現資源名稱已使用的錯誤。

步驟 4：上傳您的原始程式碼和工具鏈範本

若要建立 AWS CodeStar 專案，您必須先將原始碼封裝為 .zip 檔案，並將其放在 Amazon S3 中。使用這些內容 AWS CodeStar 初始化您的儲存庫。在 AWS CLI執行命令以建立專案時，請於輸入檔案指定此位置。

您還必須上傳`toolchain.yml`檔案並將其放在 Amazon S3 中。當您執行 命令在 中建立專案時，您可以在輸入檔案中指定此位置 AWS CLI

上傳您的原始程式碼和工具鏈範本

1. 下列範例檔案結構顯示來源檔案和工具鏈範本準備就緒可進行壓縮和上傳。範本程式碼包含 `template.yml` 檔案。請記住，此檔案與 `toolchain.yml` 檔案不同。

```
ls
src toolchain.yml

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

2. 建立原始程式碼檔案的 .zip 檔案。

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

3. 使用 cp命令，並將檔案包含為參數。

   下列命令會將 .zip 檔案 和 上傳至 `toolchain.yml` Amazon S3。

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

設定 Amazon S3 儲存貯體以共用原始程式碼

- 由於您正在 Amazon S3 中存放原始程式碼和工具鏈，因此您可以使用 Amazon S3 儲存貯體政策和物件 ACLs，以確保其他 IAM 使用者或 AWS 帳戶可以從您的範例建立專案。 AWS CodeStar 會確保建立自訂專案的任何使用者都可以存取他們要使用的工具鏈和來源。

  欲讓所有人都能使用您的範例，請執行下列命令：

```
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
```

步驟 5：在 中建立專案 AWS CodeStar

使用這些步驟來建立您的專案。

> **Important**
>
> 請確定您在其中設定偏好的 AWS 區域 AWS CLI。您的專案是在 中設定的 AWS 區域中建立 AWS CLI。

1. 執行 create-project 命令並納入 `--generate-cli-skeleton` 參數：

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

即會在輸出中顯示 JSON 格式化資料。將資料複製到本機電腦或執行個體上 AWS CLI 已安裝 的位置中的檔案 （ 例如 *input.json*)。如下所示修改複製的資料，並儲存您的結果。此輸入檔案的專案名稱設定為 `MyProject`，儲存貯體名稱則設定為 `myBucket`。

- 請確認您已提供 `roleArn` 參數。對於自訂範本，例如在本教學中的範例範本，您必須提供角色。此角色必須具有建立步驟 2：下載範例工具鏈範本中指定之所有資源的許可。
- 請確認您在 `stackParameters` 底下提供 `ProjectId` 參數。針對此教學課程提供的範例範本必須具備此參數。

```
{
    "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. 切換到包含您剛儲存之檔案的目錄，然後再次執行 create-project 命令。納入 `--cli-input-json` 參數。

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

3. 若執行成功，則會在輸出中顯示與下列內容相似的資料：

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

- 輸出包含新專案的資訊：
  - `id` 值代表專案 ID。
  - `arn` 值代表專案的 ARN。

4. 使用 describe-project 命令來檢查專案建立的狀態。納入 `--id` 參數。

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

類似下列內容的資料會顯示在輸出中：

```
{
    "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"
    }
}
```

- 輸出包含新專案的資訊：
  - `id` 值代表專屬的專案 ID。
  - `state` 值代表專案建立的狀態 (如 `CreateInProgress` 或 `CreateComplete`)。

建立專案期間，您可透過命令列或慣用的 IDE 來新增團隊成員或針對專案儲存庫進行設定存取。

教學課程：在 CodeStar 中 AWS 建立 Alexa 技能專案

AWS CodeStar 是 上的雲端型開發服務 AWS ，可提供快速開發、建置和部署應用程式所需的工具 AWS。使用 AWS CodeStar，您可以在幾分鐘內設定整個持續交付工具鏈，讓您更快地開始發佈程式碼。在 AWS CodeStar 上的 Alexa 技能專案範本可讓您從 AWS 您的帳戶建立簡單的 Hello World Alexa 技能，只需按幾下滑鼠。本範本也會建立基本的部署管道，讓您開始使用技能開發的持續整合 (CI) 工作流程。

從 建立 Alexa 技能的主要優點 AWS CodeStar 是，您可以開始使用 中的技能開發， AWS 並將 Amazon 開發人員帳戶連接到專案，以直接從中將技能部署到開發階段 AWS。部署 (CI) 管道的使用也會準備就緒，其中的儲存庫具備專案所需的所有原始碼。您可透過偏好的 IDE 來設定此儲存庫，運用熟悉的工具來建立技能。

先決條件

- 前往 https://developer.amazon.com 建立 Amazon 開發人員帳戶，可免費註冊。此帳戶會擁有您的 Alexa 技能。
- 如果您沒有 AWS 帳戶，請使用下列程序來建立帳戶。

  註冊 AWS

  1. 開啟 https://aws.amazon.com/，然後選擇建立 AWS 帳戶。

     > **Note**
     >
     > 如果您先前 AWS Management Console 使用 AWS 帳戶根使用者 登入資料登入 ，請選擇登入不同的帳戶。如果您先前使用 IAM 登入資料登入主控台，請選擇使用 AWS 帳戶根使用者 登入資料登入。然後選擇建立新 AWS 帳戶。

  2. 請遵循線上指示進行。

> **Important**
>
> 建立 Alexa 技能專案後，請將所有編輯功能限制在僅能於專案儲存庫內進行。建議您不要直接使用其他 Alexa Skills Kit 工具 (如 ASK CLI 或 ASK 開發人員主控台) 來編輯此技能，這些工具並未與專案儲存庫整合。使用這些工具會造成實際技能與儲存庫程式碼不同步。

步驟 1：建立專案並連結您的 Amazon 開發人員帳戶

此教學課程將使用 Node.js 在 AWS Lambda上執行，藉此建立技能。其他語言大部分步驟都相同，只是技能名稱會有差異。有關您所選的特定專案範本詳細資訊，請參閱專案儲存庫內的 README.md 檔案。

1. 登入 AWS Management Console，然後開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。
2. 選擇您要建立專案及其資源 AWS 的區域。Alexa 技能執行期可在下列 AWS 區域使用：
   - 亞太區域 (東京)
   - 歐洲 (愛爾蘭)
   - 美國東部 (維吉尼亞北部)
   - 美國西部 (奧勒岡)
3. 選擇建立專案。
4. 在 Choose a project template (選擇專案範本) 頁面：
   a. 針對應用程式類型，選擇 Alexa Skill。
   b. 針對程式設計語言，選擇 Node.js。
5. 選擇包含您的選取項目的方塊。
6. 在 Project name (專案名稱) 中，輸入專案的名稱 (如 **`My Alexa Skill`**)。如果您使用不同的名稱，請務必在本教學課程中使用它。 會為此專案 AWS CodeStar 選擇專案 ID 的相關識別符 （ 例如 my-alexa-skill)。如果您看到不同的專案 ID，請在此教學課程中都使用此名稱。
7. 請選擇 AWS CodeCommit (AWS CodeCommit) 做為此教學課程的儲存庫，請不要變更 Repository name (儲存庫名稱) 的值。
8. 選擇 Connect Amazon developer account (連接 Amazon 開發人員帳戶) 來連結至您的 Amazon 開發人員帳戶以託管技能。如果您沒有 Amazon 開發人員帳戶，請建立 帳戶，並先從 Amazon 開發人員完成註冊。
9. 使用您的 Amazon 開發人員登入資料登入。選擇允許，然後選擇確認以完成連線。

10若有許多廠商 ID 都與您的 Amazon 開發人員帳戶相關聯，請選擇欲用於此專案的 ID。請確認您使用的帳戶已指派管理員或開發人員角色。

11選擇 Next (下一步)。

12.（選用）如果這是您第一次 AWS CodeStar 在此 AWS 區域中使用 ，請輸入 AWS CodeStar 您要用於 IAM 使用者的顯示名稱和電子郵件地址。選擇 Next (下一步)。

13等待 AWS CodeStar 建立專案。這可能需要幾分鐘的時間。除非您看到專案佈建橫幅，否則請勿繼續。

步驟 2：在 Alexa 模擬器內測試您的技能

在第一個步驟中， 為您 AWS CodeStar 建立技能，並將其部署到 Alexa 技能開發階段。接下來，您要在 Alexa 模擬器內測試該技能。

1. 在 AWS CodeStar 主控台的專案中，選擇檢視應用程式。Alexa 模擬器將開啟新的分頁。
2. 使用您在步驟 1 連接至專案的 Amazon 開發人員帳戶登入資料來登入。
3. 在 Test (測試) 底下，選擇 Development (開發) 來啟動測試。
4. 輸入 `ask hello node hello`。技能預設的呼叫名稱為 `hello node`。
5. 您的技能應回應 `Hello World!`。

技能在 Alexa 模擬器啟用時，您亦可在支援 Alexa 的裝置 (須已向您的 Amazon 開發人員帳戶註冊) 上叫用此技能。欲在裝置上測試您的技能，請說 Alexa, ask hello node to say hello。

如需 Alexa 模擬器的詳細資訊，請參閱在開發人員主控台內測試您的技能相關文章。

步驟 3：探索您的專案資源

在建立專案的過程中， AWS CodeStar 也會代表您建立 AWS 資源。這些資源包括使用 CodeCommit 的專案儲存庫、使用 CodePipeline 的部署管道和 AWS Lambda 函數。您可以從導覽列存取這些資源。例如，選擇儲存庫會顯示 CodeCommit 儲存庫的詳細資訊。您可以在管道頁面中檢視管道部署狀態。您可以在導覽列中選擇概觀，以檢視在專案中建立的完整 AWS 資源清單。此清單包含每個資源的連結。

步驟 4：修改技能回應

您將在此步驟中小幅修改您的技能回應，以理解反覆運算的週期。

1. 在導覽列中，選擇儲存庫。選擇儲存庫名稱下的連結，專案的儲存庫會在新索引標籤或視窗中開啟。此儲存庫包含建置規格 (buildspec.yml)、 AWS CloudFormation 應用程式堆疊 (template.yml)、readme 檔案及技能套件格式 (專案結構) 內的技能原始碼。
2. 前往 lambda > custom (自訂) > index.js (若使用 Node.js) 的檔案。此檔案包含您使用 ASK SDK 的請求處理程式碼。
3. 選擇編輯。
4. 將第 24 列的字串 `Hello World!` 取代為字串 `Hello. How are you?`。
5. 向下捲動到檔案結尾。輸入作者名稱、電子郵件地址，以及選用的遞交訊息。
6. 選擇 Commit changes (遞交變更) 來確認儲存庫的變更。
7. 返回 中的專案 AWS CodeStar ，並檢查管道頁面。您現在應看到管道正在部署。
8. 管道部署完成後，請於 Alexa 模擬器內再次測試您的技能。您的技能現應回應 `Hello. How are you?`。

步驟 5：將您的本機工作站設定為連接至專案儲存庫

您稍早直接從 CodeCommit 主控台對原始碼進行了小幅變更。在此步驟中，您將設定專案儲存庫以搭配本機工作站，如此即可從命令列或您偏好的 IDE 編輯並管理程式碼。下列步驟會說明如何設定命令列工具。

1. 如有必要 AWS CodeStar，導覽至 中的專案儀表板。
2. 在導覽列中，選擇 IDE。
3. 在存取您的專案程式碼中，檢視命令列界面下方的說明。
4. 遵循指示完成以下任務：
   a. 從網站 (如 Git Downloads) 將 Git 安裝到您的本機工作站。
   b. 安裝 AWS CLI。如需詳細資訊，請參閱安裝 AWS 命令列界面。
   c. 使用 IAM 使用者存取金鑰和私密金鑰設定 AWS CLI。如需詳細資訊，請參閱設定 AWS CLI。
   d. 將專案的 CodeCommit 儲存庫複製到本機工作站。如需詳細資訊，請參閱連線至 CodeCommit 儲存庫。

後續步驟

此教學課程讓您了解基本技能的入門。欲繼續您的技能開發之旅，請參閱下列資源。

- 在 Alexa Developers YouTube 頻道觀看 How Alexa Skills Work 與其他影片，理解技能的基本概念。
- 檢閱技能套件格式、技能資訊清單結構描述及互動模型結構描述等文件，理解技能的各種元件。
- 檢閱 Alexa Skills Kit 及 ASK SDK 等文件，將您的想法轉化為技能。

教學課程：使用 GitHub 來源儲存庫建立專案

使用 AWS CodeStar，您可以設定儲存庫，以建立、檢閱和合併與專案團隊的提取請求。

在本教學課程中，您會在 GitHub 儲存庫中建立具有範例 Web 應用程式原始程式碼的專案、部署變更的管道，以及應用程式託管於雲端的 EC2 執行個體。建立專案之後，本教學課程會示範如何建立和合併 GitHub 提取請求，以變更 Web 應用程式首頁。

主題

步驟 1：建立專案並建立 GitHub 儲存庫

在此步驟中，使用 主控台來建立專案，並建立與新 GitHub 儲存庫的連線。若要存取您的 GitHub 儲存庫，您可以建立連線資源， AWS CodeStar 使用 來管理 GitHub 的授權。建立專案時，會為您佈建其額外資源。

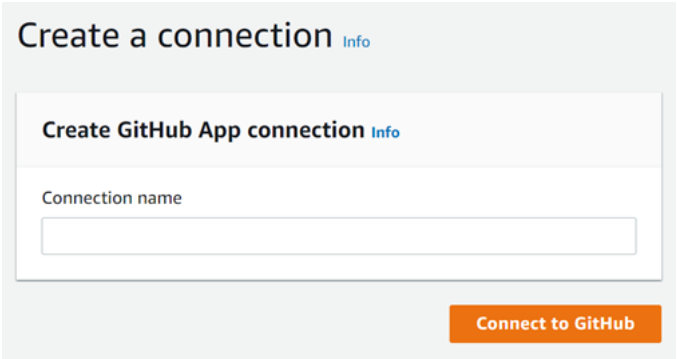
1. 登入 AWS Management Console，然後開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。
2. 選擇您要建立專案及其資源 AWS 的區域。
3. 在頁面上AWS CodeStar，選擇建立專案。
4. 在選擇專案範本頁面上，選取 Web 應用程式、Node.js 和 Amazon EC2 核取方塊。然後從符合這組選項的範本中選擇。

   如需詳細資訊，請參閱AWS CodeStar 專案範本。
5. 選擇 Next (下一步)。
6. 在 Project name (專案名稱) 中，輸入專案的名稱 (如 **`MyTeamProject`**)。如果您使用不同名稱，請在此教學課程中都使用此名稱。

7. 在專案儲存庫下，選擇 GitHub。
8. 如果您選擇 GitHub，則需要選擇或建立連線資源。如果您有現有的連線，請在搜尋欄位中選擇。否則，您將在此處建立新的連線。選擇連線到 GitHub。

   隨即顯示建立連線頁面。

   > **Note**
   >
   > 若要建立連線，您必須擁有 GitHub 帳戶。如果您要為組織建立連線，您必須是組織擁有者。



   a. 在建立 GitHub 應用程式連線下，在連線名稱中，輸入連線的名稱。選擇連線到 GitHub。

      連線至 GitHub 頁面會顯示 ，並顯示 GitHub 應用程式欄位。

   b. 在 GitHub Apps (GitHub 應用程式) 底下，選擇應用程式安裝，或選擇 Install a new app (安裝新應用程式) 以建立安裝。

      > **Note**
      >
      > 您可以為您連至特定供應商的所有連線安裝一個應用程式。如果您已安裝 AWS Connector for GitHub 應用程式，請選擇它並略過此步驟。

   c. 在安裝 GitHub AWS 連接器頁面上，選擇您要安裝應用程式的帳戶。

      > **Note**
      >
      > 如果您先前已安裝應用程式，可以選擇 Configure (設定)，繼續前往應用程式安裝的修改頁面，或者您可以使用上一步按鈕返回主控台。

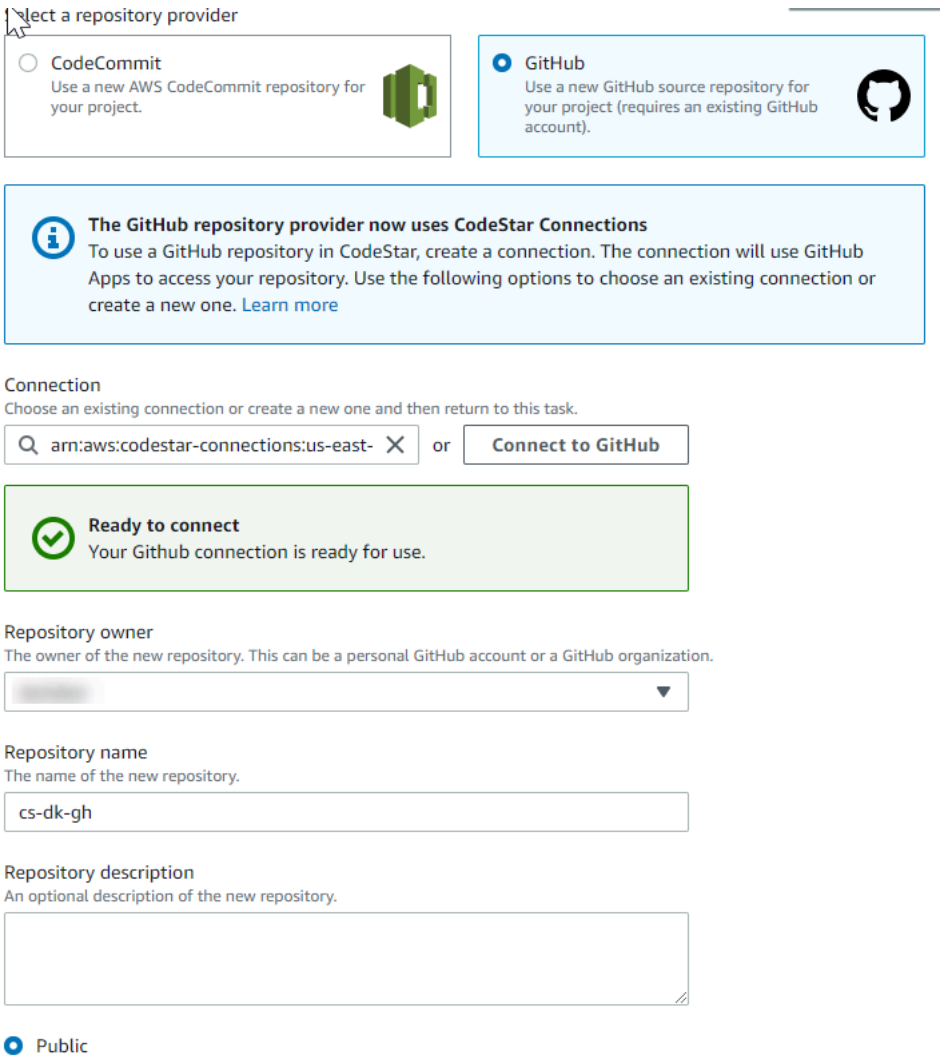
d. 如果顯示確認密碼以繼續頁面，請輸入您的 GitHub 密碼，然後選擇登入。

e. 在安裝 GitHub AWS 連接器頁面上，保留預設值，然後選擇安裝。

f. 在連線至 GitHub 頁面上，新安裝的安裝 ID 會顯示在 GitHub 應用程式中。

成功建立連線後，在 CodeStar 建立專案頁面中，即會顯示「準備連線」訊息。

> **Note**
>
> 您可以在開發人員工具主控台的設定下檢視連線。如需詳細資訊，請參閱連線入門。



g. 針對儲存庫擁有者，選擇 GitHub 組織或您的個人 GitHub 帳戶。

h. 在 Repository name (儲存庫名稱) 中，請接受預設 GitHub 儲存庫名稱，或輸入另一個名稱。

i. 選擇公有或私有。

> **Note**
>
> 如果您想要使用 AWS Cloud9 做為開發環境，您必須選擇公有儲存庫。

j. (選用) 在 Repository description (儲存庫描述) 中，輸入此 GitHub 儲存庫的描述。

9. 如果您的專案已部署到 Amazon EC2 執行個體，且您想要進行變更，請在 Amazon EC2 組態中設定 Amazon EC2 執行個體。例如，您可為專案選擇可用的執行個體類型。

   在金鑰對中，選擇您在 中建立的 Amazon EC2 金鑰對步驟 4：建立專案的 AWS CodeStar Amazon EC2 金鑰對。選取我確認我有權存取私有金鑰檔案。

10. 選擇 Next (下一步)。
11. 檢閱資源和組態詳細資訊。
12. 選擇 Next (下一步) 或 Create project (建立專案)。(顯示的選項視您的專案範本而定。)

    建立專案時，請等待幾分鐘。

13. 建立專案之後，請選擇檢視應用程式以檢視您的 Web 應用程式。

步驟 2：檢視您的原始程式碼

在此步驟中，您可以檢視原始程式碼和可用於原始儲存庫的工具。

1. 在專案的導覽列中，選擇儲存庫。

   若要在 GitHub 中檢視遞交清單，請選擇檢視遞交。這會在 GitHub 中開啟您的遞交歷史記錄。

   若要檢視問題，請選擇專案的問題索引標籤。若要在 GitHub 中建立新問題，請選擇建立 GitHub 問題。這會在 GitHub 中開啟您的儲存庫問題表單。

2. 在儲存庫索引標籤下，選擇儲存庫名稱下的連結，專案的儲存庫會在新索引標籤或視窗中開啟。此儲存庫包含專案的原始程式碼。

步驟 3：建立 GitHub 提取請求

在此步驟中，您會對原始程式碼進行次要變更，並建立提取請求。

1. 在 GitHub 中，在您的儲存庫中建立新的功能分支。選擇主要分支下拉式清單欄位，然後在名為 的欄位中輸入新的分支feature-branch。選擇建立新分支。分支已建立並為您簽出。
2. 在 GitHub 中，變更feature-branch分支。開啟公有資料夾並開啟 index.html 檔案。
3. 在 AWS CodeStar 主控台的提取請求下，若要在 GitHub 中建立提取請求，請選擇建立提取請求。這會在 GitHub 中開啟您的儲存庫提取請求表單。在 GitHub 中，選擇鉛筆圖示以編輯檔案。

   在 之後Congratulations!，新增字串，Well done, <name>!並以<name>您的名稱取代 。選擇 Commit changes (遞交變更)。變更會遞交至您的功能分支。
4. 在 AWS CodeStar 主控台中，選擇您的專案。選擇儲存庫索引標籤。在提取請求下，選擇建立提取請求。

   表單會在 GitHub 中開啟。將主要分支保留在基本分支中。針對比較，選擇您的功能分支。檢視差異。
5. 在 GitHub 中，選擇建立提取請求。建立名為 Update index.html 的提取請求。
6. 在 AWS CodeStar 主控台中，檢視新的提取請求。選擇合併變更以將變更遞交至儲存庫，並將提取請求與儲存庫的主分支合併。
7. 返回 中的專案 AWS CodeStar ，並檢查管道頁面。您現在應看到管道正在部署。
8. 建立專案之後，請選擇檢視應用程式以檢視您的 Web 應用程式。

AWS CodeStar 專案範本

AWS CodeStar 專案範本可讓您從範例應用程式開始，並使用為支援開發專案而建立 AWS 的資源進行部署。當您選擇 AWS CodeStar 專案範本時，會為您佈建應用程式類型、程式設計語言和運算平台。在使用 Web 應用程式、Web 服務、Alexa 技能和靜態網頁建立專案後，您可以將範例應用程式取代為您自己的。

AWS CodeStar 建立專案後，您可以修改支援交付應用程式 AWS 的資源。 與 AWS CodeStar 搭配使用， AWS CloudFormation 可讓您使用程式碼在雲端建立支援服務和伺服器/無伺服器平台。 AWS CloudFormation 可讓您在文字檔案中建立整個基礎設施的模型。

主題

AWS CodeStar 專案檔案和資源

AWS CodeStar 專案是原始程式碼與建立資源的組合，以部署程式碼。可協助您建置、發佈和部署程式碼的資源集合，稱為工具鏈資源。在建立專案時， AWS CloudFormation 範本使用連續整合/連續部署 (CI/CD) 管道佈建您的工具鏈資源。

您可以使用 AWS CodeStar 以兩種方式建立專案，具體取決於您在 AWS 資源建立方面的經驗水準：

- 當您使用主控台建立專案時， 會 AWS CodeStar 建立您的工具鏈資源，包括您的儲存庫，並將範例應用程式程式碼和專案檔案填入您的儲存庫。根據一組預先設定專案選項，使用主控台以快速設定範例專案。
- 當您使用 CLI 建立專案時，您會提供 AWS CloudFormation 範本來建立工具鏈資源和應用程式原始碼。使用 CLI AWS CodeStar 允許 從您的範本建立專案，然後使用您的範本程式碼填入您的儲存庫。

AWS CodeStar 專案提供單一管理點。您可以使用建立專案精靈，在主控台設定範例專案。然後，您可以將它用做協作平台來管理團隊的權限和資源。如需詳細資訊，請參閱什麼是 AWS CodeStar？。當使用主控台來建立專案時，會提供您的原始碼做為範本程式碼，並且為您建立 CI/CD 工具鏈資源

當您在 主控台中建立專案時， 會 AWS CodeStar 佈建下列資源：

- GitHub 或 CodeCommit 中的程式碼儲存庫。
- 在專案儲存庫中，`README.md` 檔案提供檔案和目錄的詳細資訊。
- 在專案儲存庫中，`template.yml` 檔案存放您的應用程式執行階段堆疊的定義。您可以使用此檔案來新增或修改非工具鏈資源的專案資源，例如用於通知、資料庫支援、監控和追蹤 AWS 的資源。
- AWS 與您的管道建立關聯的 服務和資源，例如 Amazon S3 成品儲存貯體、Amazon CloudWatch Events 和相關服務角色。
- 具備完整原始碼和公有 HTTP 端點的作用中範例應用程式。
- 根據 AWS CodeStar 專案範本類型的 AWS 運算資源：
  - Lambda 函數。
  - Amazon EC2 執行個體。
  - AWS Elastic Beanstalk 環境。
- 從 2018 年 12 月 6 日開始，PDT：
  - 許可邊界，是一種專用 IAM 政策，用於控制對專案資源的存取權。許可邊界預設會連接到範例專案中的角色。如需詳細資訊，請參閱工作者角色的 IAM 許可邊界。
  - 使用 AWS CloudFormation 建立專案資源的 IAM 角色 AWS CloudFormation ，其中包含所有 AWS CloudFormation 支援資源的許可，包括 IAM 角色。
  - 工具鏈 IAM 角色。
  - 您可以在應用程式堆疊中修改 Lambda 的執行角色。
- 2018 年 12 月 6 日之前的 PDT：
  - IAM AWS CloudFormation 角色，用於建立支援有限資源集的專案 AWS CloudFormation 資源。
  - 用於建立 CodePipeline 資源的 IAM 角色。
  - 用於建立 CodeBuild 資源的 IAM 角色。
  - 用於建立 CodeDeploy 資源的 IAM 角色，如果適用於您的專案類型。
  - 用於建立 Amazon EC2 Web 應用程式的 IAM 角色，如果適用於您的專案類型。
  - 用於建立 CloudWatch Events 資源的 IAM 角色。
  - Lambda 的執行角色，會動態修改以包含部分資源集。

專案包含顯示狀態的詳細資訊頁面，並包含團隊管理的連結、IDEs 或儲存庫的設定指示連結，以及儲存庫中原始碼變更的遞交歷史記錄。您也可以選擇工具，用於連接到外部問題追蹤工具，例如 Jira 。

開始使用：選擇專案範本

當您在主控台中選擇 AWS CodeStar 專案時，您會從一組預先設定的選項中選擇範本程式碼和資源，以快速開始使用。這些選項稱為專案範本。每個 AWS CodeStar 專案範本都包含程式設計語言、應用程式類型和運算平台。您選擇的組合決定專案範本。

選擇範本運算平台

每個範本會設定以下其中一種運算平台類型：

- 當您選擇 AWS Elastic Beanstalk 專案時，您會部署到雲端 Amazon Elastic Compute Cloud 執行個體上的環境 AWS Elastic Beanstalk 。
- 當您選擇 Amazon EC2 專案時， 會 AWS CodeStar 建立 Linux EC2 執行個體，以在雲端中託管您的應用程式。您的專案團隊成員可以存取執行個體，而您的團隊會使用您提供給 SSH 的金鑰對，以加入您的 Amazon EC2 執行個體。 AWS CodeStar 也有受管 SSH，使用團隊成員許可來管理金鑰對連線。
- 當您選擇時 AWS Lambda， 會 AWS CodeStar 建立透過 Amazon API Gateway 存取的無伺服器環境，無需維護任何執行個體或伺服器。

選擇範本應用程式類型

每個範本會設定以下其中一種應用程式類型：

- Web 服務

  Web 服務用於在背景執行任務，例如呼叫 API。在 AWS CodeStar 建立您的範例 Web 服務專案後，您可以選擇端點 URL 來查看 Hello World 輸出，但此應用程式類型的主要用途不是做為使用者介面 (UI)。此類別中的 AWS CodeStar 專案範本支援在 Ruby、Java、ASP.NET：//2、PHP、Node.js 等開發。

- Web 應用程式

  Web 應用程式具有 UI。在 AWS CodeStar 建立您的範例 Web 應用程式專案之後，您可以選擇端點 URL 以查看互動式 Web 應用程式。此類別中的 AWS CodeStar 專案範本支援在 Ruby、Java、ASP.NET：//2、PHP、Node.js 等開發。

- 靜態網頁

如果您想要 HTML 網站適用的專案，請選擇此範本。此類別的 AWS CodeStar 專案範本支援以 HTML5 開發。

- Alexa 技能

  若您的 Alexa 技能專案須搭配 AWS Lambda 函數，請選擇此範本。當您建立技能專案時，AWS CodeStar 會傳回一個 Amazon Resource Name (ARN)，您可以將其用作服務端點。如需詳細資訊，請參閱將自訂技能託管為 AWS Lambda 函數。

  > **Note**
  >
  > Alexa 技能的 Lambda 函數僅在美國東部 (維吉尼亞北部)、美國西部 (奧勒岡)、歐洲 (愛爾蘭) 和亞太區域 (東京) 區域受到支援。

- Config 規則

  如果您想要 AWS Config 規則的專案，可讓您在帳戶中跨 AWS 資源自動化規則，請選擇此範本。該函數會傳回可當成您的規則的服務端點使用的 ARN。

選擇範本程式設計語言

當您選擇專案範本時，可選擇像 Ruby、Java、ASP.NET、PHP、Node.js 及更多的程式設計語言。

如何變更您的 AWS CodeStar 專案

您可以更新您的專案，藉由修改：

- 用於您的應用程式的範本程式碼和程式設計語言資源。
- 基礎設施的組成資源，也是您的應用程式 (作業系統、支援應用程式和服務、部署參數和雲端運算平台) 存放和部署之處。您可以在 `template.yml` 檔案中修改應用程式資源。這是建立您的應用程式執行階段環境所需的 AWS CloudFormation 檔案。

> **Note**
>
> 如果您使用 Alexa Skills AWS CodeStar 專案，則無法變更 AWS CodeStar 來源儲存庫 (CodeCommit 或 GitHub) 以外的技能。若您在 Alexa 開發人員入口網站編輯技能，變更可能不會出現在來源儲存庫，造成這兩個版本不同步。

變更應用程式原始碼和推送變更

若要修改範例原始碼、指令碼和其他應用程式來源檔案，請以下列方式編輯您的來源儲存庫的檔案：

- 在 CodeCommit 或 GitHub 中使用編輯模式。
- 在 IDE 中開啟專案，例如 AWS Cloud9。
- 在本機複製儲存庫，然後認可和推送您的變更。如需相關資訊，請參閱 步驟 4：遞交變更。

使用 Template.yml 檔案變更應用程式資源

使用 AWS CloudFormation 來建模和部署應用程式的執行期資源，而不是手動修改基礎設施資源。

您可以透過編輯您的專案儲存庫中的 `template.yml` 檔案，在執行階段堆疊中修改或新增應用程式資源 (例如，Lambda 函數)。您可以新增可當成 AWS CloudFormation 資源使用的任何資源。

若要變更 AWS Lambda 函數的程式碼或設定，請參閱 新增資源到專案。

修改專案儲存庫中的 `template.yml` 檔案，以新增屬於應用程式 AWS CloudFormation 資源的資源類型。當您將應用程式資源新增至 `template.yml` 檔案的 `Resources`區段時， AWS CloudFormation 請為您 AWS CodeStar 建立資源。如需 AWS CloudFormation 資源及其所需屬性的清單，請參閱AWS 資源類型參考。如需詳細資訊，請參閱步驟 1：編輯 IAM 中的 CloudFormation 工作者角色所提供的此範例。

AWS CodeStar 可讓您透過設定和建模應用程式的執行期環境來實作最佳實務。

如何管理變更應用程式資源的許可

當您使用 AWS CloudFormation 來新增執行時間應用程式資源，例如 Lambda 函數時， AWS CloudFormation 工作者角色可以使用它已經擁有的許可。對於某些執行時間應用程式資源，您必須先手動調整 AWS CloudFormation 工作者角色的許可，再編輯 `template.yml` 檔案。

如需變更 AWS CloudFormation 工作者角色許可的範例，請參閱 步驟 5：在資源許可新增內嵌政策。

AWS CodeStar 最佳實務

AWS CodeStar 已與許多 產品和服務整合。下列各節說明 AWS CodeStar 和這些相關產品和服務的最佳實務。

主題

AWS CodeStar 資源的安全最佳實務

您應該定期套用修補程式，並針對應用程式所使用的依存項目，檢閱其安全最佳實務。若要在生產環境中更新範本程式碼及維護專案，則可善用這些安全最佳實務：

- 追蹤架構的後續安全公告和更新。
- 在開始部署專案前，請務必遵循專為架構所開發的最佳實務。
- 定期查看架構的依存項目，並視需要進行更新。
- 每個 AWS CodeStar 範本都包含程式設計語言的組態指示。請參閱 `README.md` 檔案，其位於專案的來源儲存庫中。
- 作為隔離專案資源的最佳實務，請使用 中介紹的多帳戶策略來管理 AWS 資源的最低權限存取中的安全性 AWS CodeStar。

設定依存項目版本的最佳實務

AWS CodeStar 專案中的範例原始程式碼使用來源儲存庫中 `package.json` 檔案所列的相依性。根據最佳實務，您應一律將依存項目設為指向特定版本。這就是所謂的鎖定版本。不建議您將版本設定為 `latest`，因為該版本的變更可能會導致應用程式損壞，且不會另行通知。

監控和記錄 AWS CodeStar 資源的最佳實務

您可以使用 中的記錄功能 AWS 來判斷使用者在帳戶中採取的動作，以及所使用的資源。日誌檔顯示：

- 動作的時間和日期。
- 動作的來源 IP 地址。
- 哪些動作因許可不足而失敗。

AWS CloudTrail 可用來記錄 AWS API 呼叫，以及由帳戶或代表 AWS 帳戶進行的相關事件。如需詳細資訊，請參閱使用 記錄 AWS CodeStar API 呼叫 AWS CloudTrail。

在 中使用專案 AWS CodeStar

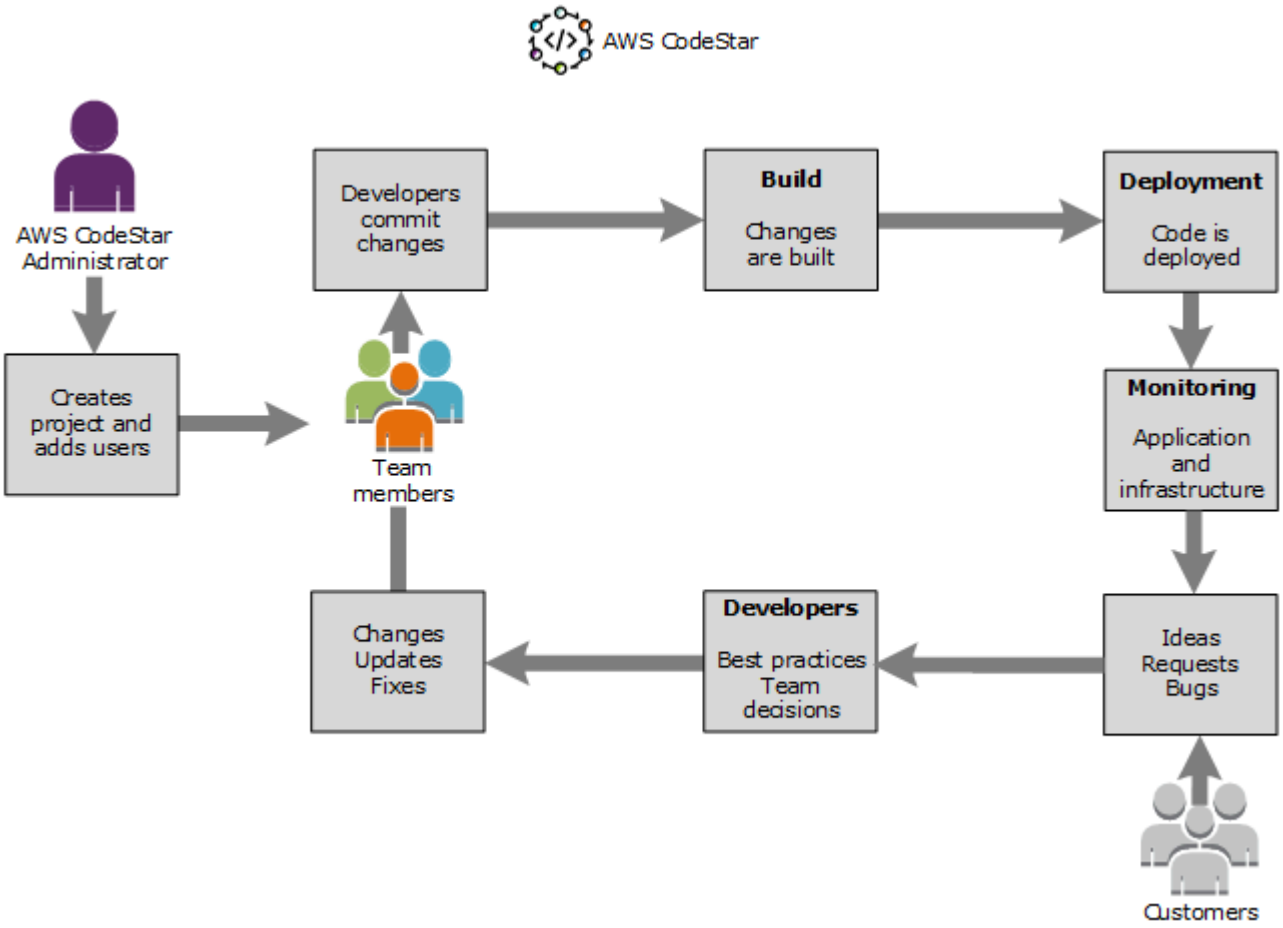
使用 AWS CodeStar 專案範本時，您可以快速建立已設定所需資源的專案，包括：

- 來源儲存庫
- 重建環境
- 部署和託管資源
- 程式設計語言

範本甚至包含範例原始程式碼，因此您可以立即開始使用您的專案。

您具有專案之後，可以新增或移除資源、自訂您的專案儀表板和監控進度。

下圖顯示 AWS CodeStar 專案中的基本工作流程。



圖表中的基本工作流程會顯示已套用政策的開發人員，該AWSCodeStarFullAccess政策會建立專案並新增團隊成員至該專案。他們一起編寫、建置、測試和部署程式碼。專案儀表板提供工具，可用於即

時檢視應用程式活動，並透過部署管道監控建置、程式碼流程及其他。團隊使用團隊 wiki 圖磚共享資訊、最佳實務和連結。他們整合問題追蹤軟體，協助追蹤進度和任務。當客戶提出請求和意見回饋時，團隊會將此資訊加入到專案，並整合到他們的專案規劃和開發。隨著專案擴增，團隊新增更多團隊成員來支援他們的程式碼基底。

在 中建立專案 AWS CodeStar

您可以使用 AWS CodeStar 主控台來建立專案。如果您使用專案範本，它會為您設定所需的資源。該範本還包含可讓您用來開始編寫程式碼的範例程式碼。

若要建立專案， AWS Management Console 請使用具有 `AWSCodeStarFullAccess`政策或同等許可的 IAM 使用者登入 。如需詳細資訊，請參閱設定 AWS CodeStar。

> **Note**
>
> 您必須完成 中的步驟，設定 AWS CodeStar才能完成本主題中的程序。

主題

- 在 AWS CodeStar 中建立專案 (主控台)
- 在 AWS CodeStar (AWS CLI) 中建立專案

在 AWS CodeStar 中建立專案 (主控台)

使用 AWS CodeStar 主控台建立專案。

在 中建立專案 AWS CodeStar

1. 登入 AWS Management Console，然後開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。

   請確定您已登入要建立專案及其資源的 AWS 區域。例如，若要在美國東部 （俄亥俄） 建立專案，請確定您已選取該 AWS 區域。如需 AWS CodeStar 可用 AWS 區域的相關資訊，請參閱《AWS 一般參考》中的區域和端點。
2. 在頁面上AWS CodeStar，選擇建立專案。
3. 在選擇專案範本頁面上，從專案範本清單中選擇 AWS CodeStar 專案類型。您可使用篩選條件搜尋列，以縮減選項。例如，對於以 Node.js 撰寫的 Web 應用程式專案，以部署到 Amazon EC2 執

行個體，請選取 Web 應用程式、Node.js 和 Amazon EC2 核取方塊。然後從符合這組選項的範本中選擇。

如需詳細資訊，請參閱AWS CodeStar 專案範本。

4. 選擇 Next (下一步)。
5. 在專案名稱文字輸入欄位中，輸入專案的名稱，例如*#######*。在專案 ID 中，專案的 ID 衍生自此專案名稱，但限制為 15 個字元。

   例如，名為 *My First Project* 之專案的預設 ID 為 *my-first-projec*。此專案 ID 是與專案相關聯的所有資源名稱的基礎。 AWS CodeStar 會使用此專案 ID 做為程式碼儲存庫 URL 的一部分，以及 IAM 中相關安全存取角色和政策的名稱。專案建立之後，就無法再變更其 ID。若要在建立專案之前編輯專案 ID，請在專案 ID 中輸入您要使用的 ID。

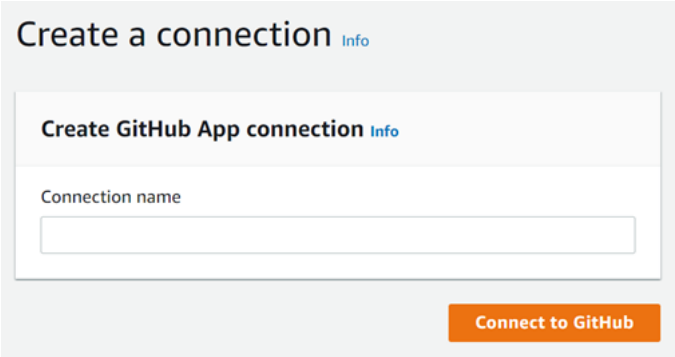
   如需專案名稱和專案 ID 限制的資訊，請參閱 中的限制 AWS CodeStar。

   > **Note**
   >
   > 專案 IDs 對於 AWS 區域中 AWS 的帳戶必須是唯一的。

6. 選擇儲存庫提供者、AWS CodeCommit 或 GitHub (GitHub)。
7. 如果您選擇 AWS CodeCommit，請針對儲存庫名稱接受預設 AWS CodeCommit 儲存庫名稱，或輸入不同的名稱。然後跳到步驟 9。
8. 如果您選擇 GitHub，則需要選擇或建立連線資源。如果您有現有的連線，請在搜尋欄位中選擇。否則，請立即建立新的連線。選擇連線到 GitHub。

   隨即顯示建立連線頁面。

   > **Note**
   >
   > 若要建立連線，您必須擁有 GitHub 帳戶。如果您要為組織建立連線，您必須是組織擁有者。



a. 在建立 GitHub 應用程式連線下，在連線名稱輸入文字欄位中輸入連線的名稱。選擇連線到 GitHub。

   連線至 GitHub 頁面會顯示 ，並顯示 GitHub 應用程式欄位。

b. 在 GitHub Apps (GitHub 應用程式) 底下，選擇應用程式安裝，或選擇 Install a new app (安裝新應用程式) 以建立安裝。

> **Note**
>
> 您可以為您連至特定供應商的所有連線安裝一個應用程式。如果您已安裝 AWS Connector for GitHub 應用程式，請選擇它並略過此步驟。

c. 在安裝 GitHub AWS 連接器頁面上，選擇您要安裝應用程式的帳戶。

> **Note**
>
> 如果您先前已安裝應用程式，可以選擇 Configure (設定)，繼續前往應用程式安裝的修改頁面，或者您可以使用上一步按鈕返回主控台。

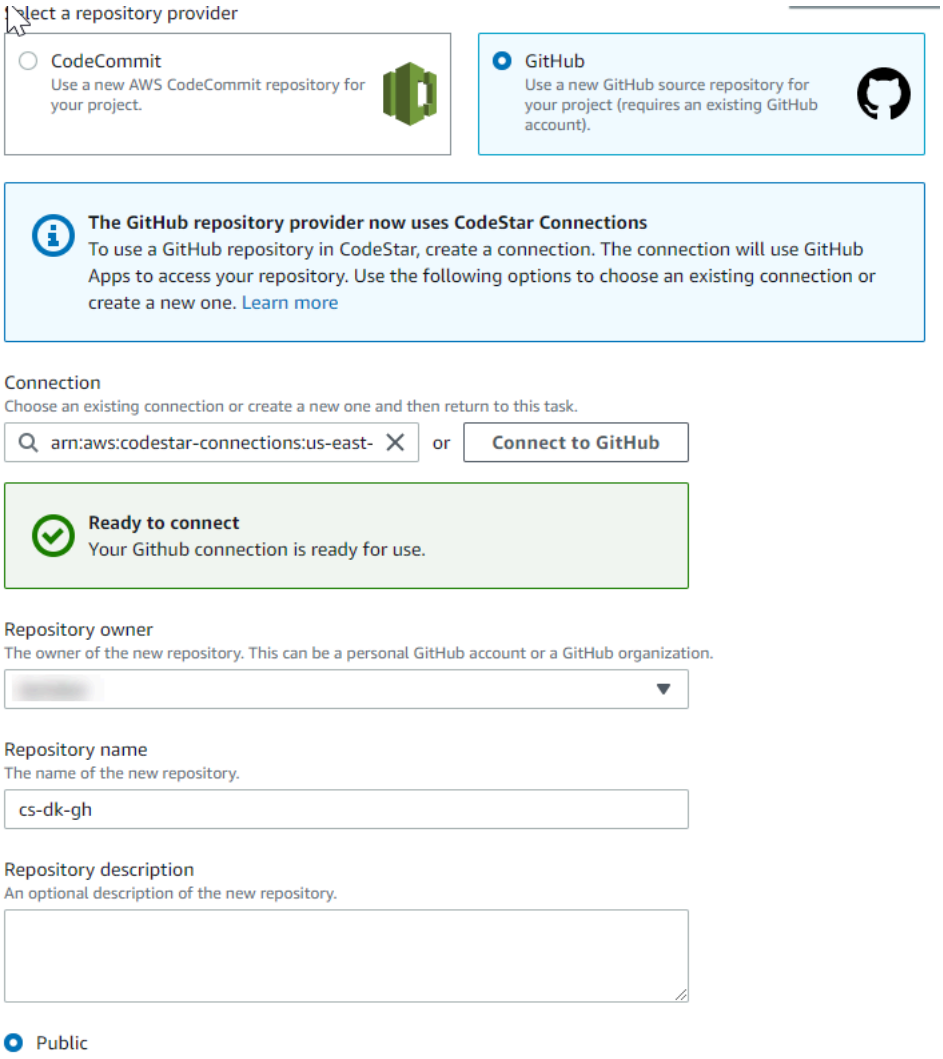
d. 如果顯示確認密碼以繼續頁面，請輸入您的 GitHub 密碼，然後選擇登入。

e. 在安裝 GitHub AWS 連接器頁面上，保留預設值，然後選擇安裝。

f. 在連線至 GitHub 頁面上，新安裝的安裝 ID 會顯示在 GitHub Apps 文字輸入欄位中。

   建立連線後，在 CodeStar 建立專案頁面中，即會顯示準備連線訊息。

> **Note**
>
> 您可以在開發人員工具主控台的設定下檢視連線。如需詳細資訊，請參閱連線入門。



g. 針對儲存庫擁有者，選擇 GitHub 組織或您的個人 GitHub 帳戶。

h. 在 Repository name (儲存庫名稱) 中，請接受預設 GitHub 儲存庫名稱，或輸入另一個名稱。

i. 選擇公有或私有。

> **Note**
>
> 若要使用 AWS Cloud9 做為您的開發環境，您必須選擇公有。

j. (選用) 在 Repository description (儲存庫描述) 中，輸入此 GitHub 儲存庫的描述。

> **Note**
>
> 如果您選擇 Alexa Skill 專案範本，則需要連接 Amazon 開發人員帳戶。如需使用 Alexa Skill 專案的詳細資訊，請參閱 教學課程：在 CodeStar 中 AWS 建立 Alexa 技能專案。

9. 如果您的專案已部署到 Amazon EC2 執行個體，且您想要進行變更，請在 Amazon EC2 組態中設定 Amazon EC2 執行個體。例如，您可為專案選擇可用的執行個體類型。

> **Note**
>
> 不同的 Amazon EC2 執行個體類型提供不同層級的運算能力，並且可能有不同的相關成本。如需詳細資訊，請參閱 Amazon EC2 執行個體類型和 Amazon EC2 定價。
> 如果您有多個虛擬私有雲端 (VPC) 或在 Amazon Virtual Private Cloud 中建立的多個子網路，您也可以選擇要使用的 VPC 和子網路。不過，如果您選擇專用執行個體上不支援的 Amazon EC2 執行個體類型，則無法選擇執行個體租用設定為專用的 VPC。
> 如需詳細資訊，請參閱什麼是 Amazon VPC？和專用執行個體基本概念。

在金鑰對中，選擇您在 中建立的 Amazon EC2 金鑰對步驟 4：建立專案的 AWS CodeStar Amazon EC2 金鑰對。選取我確認我有權存取私有金鑰檔案。

10. 選擇 Next (下一步)。
11. 檢閱資源和組態詳細資訊。
12. 選擇 Next (下一步) 或 Create project (建立專案)。(顯示的選項視您的專案範本而定。)

    建立專案可能需要幾分鐘的時間，包括 儲存庫。
13. 專案擁有儲存庫之後，您可以使用儲存庫頁面來設定其存取權。使用後續步驟中的連結來設定 IDE、設定問題追蹤，或將團隊成員新增至您的專案。

建立專案期間，您可透過命令列或慣用的 IDE 來新增團隊成員或針對專案儲存庫進行設定存取。

在 AWS CodeStar (AWS CLI) 中建立專案

AWS CodeStar 專案是原始程式碼與資源的組合，這些資源是用來部署程式碼。可協助您建置、發佈和部署程式碼的資源集合，稱為工具鏈資源。在專案建立時， AWS CloudFormation 範本會在持續整合/持續部署 (CI/CD) 管道中佈建您的工具鏈資源。

使用主控台來建立專案時，即會為您建立工具鏈範本。當您使用 AWS CLI 建立專案時，您可以建立工具鏈範本來建立工具鏈資源。

完整工具鏈需要以下建議的資源：

1. 包含原始碼的 CodeCommit 或 GitHub 儲存庫。
2. 設定為接聽儲存庫變更的 CodePipeline 管道。
   a. 當您使用 CodeBuild 執行單元或整合測試時，我們建議您將建置階段新增至管道，以建立建置成品。
   b. 建議您將部署階段新增至使用 CodeDeploy 的管道 AWS CloudFormation ，或將建置成品和原始程式碼部署至執行期基礎設施。

> **Note**
>
> 由於 CodePipeline 在管道中至少需要兩個階段，而第一個階段必須是來源階段，請新增建置或部署階段做為第二個階段。

AWS CodeStar 工具鏈定義為 CloudFormation 範本。

如需逐步介紹此任務和設定範例資源的教學，請參閱教學課程： AWS CodeStar 使用 在 中建立專案 AWS CLI。

先決條件：

建立專案時，您可以在輸入檔案中提供以下參數。如果未提供下列項目， 會 AWS CodeStar 建立空專案。

- 來源碼。如果此參數已包含在您的請求中，則還必須包含工具鏈範本。
  - 您的來源碼必須包含執行您的專案所需的應用程式的程式碼。
  - 您的原始碼必須包含任何必要的組態檔案，例如 CodeBuild 專案的 buildspec.yml 或 CodeDeploy 部署的 appspec.yml。
  - 您可以在原始程式碼中包含選用項目，例如 README 或非工具鏈 AWS 資源的 template.yml。

- 工具鏈範本。您的工具鏈範本會為您的專案佈建要管理 AWS 的資源和 IAM 角色。
- 來源位置。如果為您的專案指定來源碼和工具鏈範本，則必須提供位置。將您的來源檔案和工具鏈範本上傳至 Amazon S3 儲存貯體。 AWS CodeStar 會擷取檔案，並使用它們來建立專案。

> **Important**
>
> 請務必在 中設定偏好的 AWS 區域 AWS CLI。您的專案是在 中設定的 AWS 區域中建立 AWS CLI。

1. 執行 create-project 命令並納入 `--generate-cli-skeleton` 參數：

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

即會在輸出中顯示 JSON 格式化資料。將資料複製到本機電腦或執行個體上 AWS CLI 已安裝 的位置中的檔案 （ 例如 *input.json*)。如下所示修改複製的資料，並儲存您的結果。

```
{
    "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"
    }
}
```

取代以下項目：

- *project-name*：必要。此 AWS CodeStar 專案的易記名稱。
- *project-id*：必要。此專案的 AWS CodeStar 專案 ID。

> **Note**
>
> 建立專案時，您必須擁有唯一的專案 ID。如果提交具有已存在專案 ID 的輸入檔，您會遇到錯誤。

- *description*：選用。此 AWS CodeStar 專案的描述。
- *sourceCode*：選用。提供給專案的來源碼組態資訊。目前只支援單一 sourceCode 物件。每個sourceCode物件都包含擷取來源碼的位置， AWS CodeStar 以及填入來源碼的目的地的相關資訊。
  - *source*：必要。這可定義您上傳來源碼的位置。唯一支援的來源是 Amazon S3. AWS CodeStar retrieve the source code，並在建立專案後將其包含在儲存庫中。
    - *S3*：選用。原始程式碼的 Amazon S3 位置。
      - *bucket-name*：包含您的來源碼的儲存貯體。

  - *bucket-key*：指向包含您的來源碼的 .zip 檔案 (例如，src.zip) 的儲存貯體字首和物件金鑰。

- *destination*：選用。建立專案時您的來源碼要填入的目的地位置。原始碼支援的目的地為 CodeCommit 和 GitHub。

  您只可以提供這兩個選項中的一個：

  - *codeCommit*：唯一必要的屬性是 CodeCommit 儲存庫的名稱，其中應包含您的原始程式碼。這個儲存庫應該在您的工具鏈範本中。

    > **Note**
    >
    > 對於 CodeCommit，您必須提供您在工具鏈堆疊中定義的儲存庫名稱。 會使用您在 Amazon S3 中提供的原始碼 AWS CodeStar 初始化此儲存庫。

  - *gitHub*：此物件代表建立 GitHub 儲存庫並對它植入來源碼所需的資訊。如果您選擇 GitHub 儲存庫，則以下為必要值。

    > **Note**
    >
    > 對於 GitHub，您無法指定現有的 GitHub 儲存庫。 會為您 AWS CodeStar 建立一個儲存庫，並使用您上傳到 Amazon S3 的原始碼填入此儲存庫。 AWS CodeStar 會使用以下資訊在 GitHub 中建立您的儲存庫。

    - *name*：必要。您的 GitHub 儲存庫的名稱。
    - *description*：必要。您的 GitHub 儲存庫的描述。
    - *type*：必要。GitHub 儲存庫的類型。有效值為 User 或 Organization。
    - *owner*：必要。您的儲存庫擁有者的 GitHub 使用者 名稱。如果儲存庫應該由 GitHub 組織擁有，請提供組織名稱。
    - *privateRepository*：必要。希望此儲存庫是私有或公有。有效值為 true 或 false。
    - *issuesEnabled*：必要。是否要在 GitHub 中使用此儲存庫啟用問題。有效值為 true 或 false。
    - *Token*：選用。這是 AWS CodeStar 用於存取 GitHub 帳戶的個人存取字符。此字符必須包含以下範圍：repo、user 和 admin:repo_hook。若要從 GitHub 擷取個人存取字符，請參閱 GitHub 網站上的 Creating a Personal Access Token for the Command Line。

> **Note**
>
> 如果您使用 CLI 建立具有 GitHub 來源儲存庫的專案， AWS CodeStar 會使用您的字符透過 OAuth 應用程式存取儲存庫。如果您使用 主控台建立具有 GitHub 來源儲存庫的專案， AWS CodeStar 會使用連線資源，以 GitHub 應用程式存取儲存庫。

- *toolchain*：建立專案時要設定的 CI/CD 工具鏈的相關資訊。這包括您上傳工具鏈範本的位置。範本會建立 AWS CloudFormation 堆疊，其中包含您的工具鏈資源。這也包含 AWS CloudFormation 參考的任何參數覆寫，以及用於建立堆疊的角色。 AWS CodeStar 會擷取範本並使用 AWS CloudFormation 執行範本。
  - *source*：必要。工具鏈範本的位置。Amazon S3 是唯一支援的來源位置。
    - *S3*：選用。您上傳工具鏈範本的 Amazon S3 位置。
      - *bucket-name*：Amazon S3 儲存貯體名稱。
      - *bucket-key*：指向包含您的工具鏈範本的 .yml 或 .json 檔案 (例如，`files/toolchain.yml`) 的儲存貯體字首和物件金鑰。
  - *stackParameters*：選用。包含要傳送至 AWS CloudFormation的金鑰值對。這些是您的工具鏈範本設定要參考的參數 (如果有)。
  - *role*：選用。此角色用於建立在您的帳戶中建立工具鏈資源。需要的角色如下所示：
    - 如果未提供角色，則如果工具鏈是 AWS CodeStar 快速入門範本， AWS CodeStar 會使用為您的帳戶建立的預設服務角色。如果服務角色不存在於您的帳戶，則可以建立一個角色。如需相關資訊，請參閱 步驟 2：建立 AWS CodeStar 服務角色。
    - 如果您上傳並使用自己的自訂工具鏈範本，則必須提供該角色。您可以根據 AWS CodeStar 服務角色和政策陳述式來建立角色。如需此政策陳述式的範例，請參閱 AWSCodeStarServiceRole 政策。
- *tags*：選用。連接至 AWS CodeStar 專案的標籤。

> **Note**
>
> 這些標籤未連接到專案中包含的資源。

2. 切換到包含您剛儲存之檔案的目錄，然後再次執行 create-project 命令。納入 `--cli-input-json` 參數。

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

3. 若執行成功，則會在輸出中顯示與下列內容相似的資料：

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

- 輸出包含新專案的資訊：
  - id 值代表專案 ID。
  - arn 值代表專案的 ARN。

4. 使用 describe-project 命令來檢查專案建立的狀態。納入 --id 參數。

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

類似下列內容的資料會顯示在輸出中：

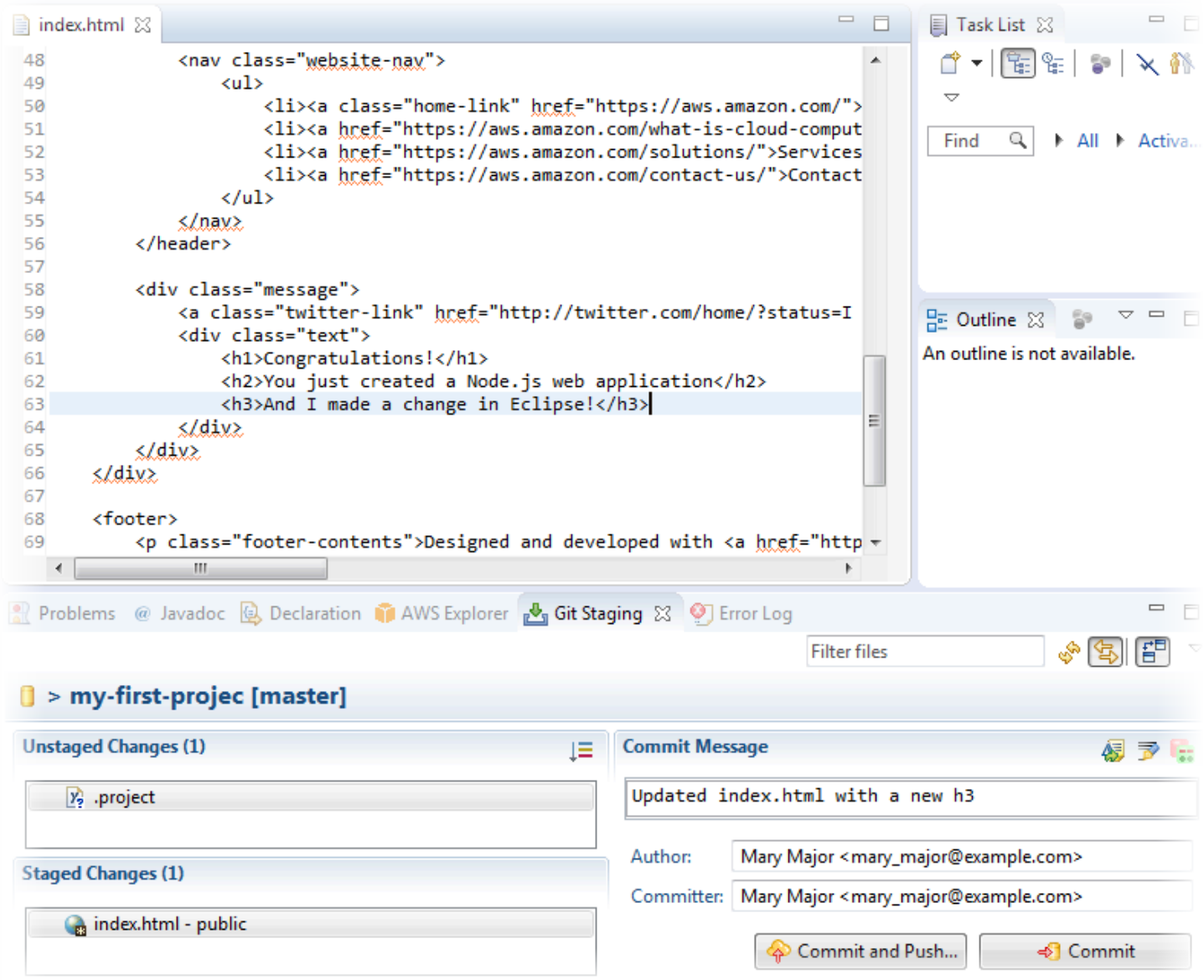
```
{
    "name": "MyProject",
    "id": "myproject",
    "arn": "arn:aws:codestar:us-east-1:account_ID:project/myproject",
    "description": "",
    "createdTimeStamp": 1539700079.472,
    "stackId": "arn:aws:cloudformation:us-east-1:account_ID:stack/awscodestar-
myproject/stack-ID",
    "status":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

- 輸出包含新專案的資訊：
  - state 值代表專案建立的狀態 (如 CreateInProgress 或 CreateComplete)。

建立專案期間，您可透過命令列或慣用的 IDE 來新增團隊成員或針對專案儲存庫進行設定存取。

搭配 使用 IDE AWS CodeStar

當您將 IDE 與 整合時 AWS CodeStar，您可以繼續在偏好的環境中撰寫和開發程式碼。每次您遞交和推送程式碼時， AWS CodeStar 專案中都會包含您所做的變更。



主題

- AWS Cloud9 搭配 使用 AWS CodeStar
- 搭配 使用 Eclipse AWS CodeStar
- 搭配 使用 Visual Studio AWS CodeStar

AWS Cloud9 搭配 使用 AWS CodeStar

您可以使用 AWS Cloud9 在 AWS CodeStar 專案中進行程式碼變更和開發軟體。 AWS Cloud9 是一種線上 IDE，您可以透過 Web 瀏覽器存取。IDE 提供豐富的程式碼編輯體驗，可支援多種程式設計語言和執行時間除錯器，以及內建終端機。在背景中，Amazon EC2 執行個體託管 AWS Cloud9 開發環境。此環境提供 AWS CodeStar 專案程式碼檔案的 AWS Cloud9 IDE 和存取權。如需詳細資訊，請參閱《AWS Cloud9 使用者指南》。

您可以使用 AWS CodeStar 主控台或 AWS Cloud9 主控台，為將程式碼存放在 CodeCommit 中的專案建立 AWS Cloud9 開發環境。對於在 GitHub 中存放程式碼的 AWS CodeStar 專案，您只能使用 AWS Cloud9 主控台。此主題說明如何同時使用兩個主控台。

若要使用 AWS Cloud9，您需要：

- 已新增為 AWS CodeStar 專案團隊成員的 IAM 使用者。
- 如果 AWS CodeStar 專案將其原始程式碼存放在 CodeCommit 中，則為 IAM 使用者的 AWS 憑證。

主題

建立專案 AWS Cloud9 的環境

請依照下列步驟來建立 AWS CodeStar 專案的 AWS Cloud9 開發環境。

1. 建立專案 如果您想要建立新專案，請遵循 中的步驟。
2. 在 AWS CodeStar 主控台中開啟專案。在導覽列上，選擇 IDE。選擇建立環境，然後使用下列步驟。

> **Important**
>
> 如果專案位於 AWS Cloud9 不支援 AWS 的區域，則導覽列的 IDE 索引標籤中不會顯示 AWS Cloud9 選項。不過，您可以使用 AWS Cloud9 主控台來建立開發環境、開啟新

環境，然後將其連接至專案的 AWS CodeCommit 儲存庫。略過下列步驟，並參閱AWS Cloud9 《 使用者指南》中的建立環境、開啟環境和AWS CodeCommit 範例。如需支援的 AWS 區域清單，請參閱 AWS Cloud9 中的 Amazon Web Services 一般參考。

在建立 AWS Cloud9 環境中，自訂專案預設值。

1. 若要變更 Amazon EC2 執行個體的預設類型以託管環境，請在執行個體類型中選擇執行個體類型。
2. AWS Cloud9 會在您的帳戶中使用 Amazon Virtual Private Cloud (Amazon VPC) AWS 與執行個體通訊。根據 AWS 您的帳戶中 Amazon VPC 的設定方式，執行下列其中一項操作。

該帳戶是否具有至少包含一個子網路的 VPC？	您想要在帳戶中 AWS Cloud9 使用預設 VPC 的 VPC 是？	此 VPC 是否具有單一子網路？	執行此作業
否	—	—	如果沒有 VPC 存在，請建立一個。請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，選擇 Create VPC (建立 VPC)，然後遵循頁面上的指示。如需詳細資訊，請參閱《 使用者指南》中的為 建立 Amazon VPC AWS Cloud9。 AWS Cloud9 如果 VPC 已存在但沒有子網路，請建立一個。請展開 Network settings (網路設定)。針對 Network (VPC) (網路 (VPC))，選擇 Create subnet (建立子網路)，然後遵循指示執行。如需詳細資訊，請參閱AWS Cloud9 《 使用者指南》中的為 建立子網路 AWS Cloud9。
是	是	是	跳到此程序的步驟 4。 (AWS Cloud9 使用預設 VPC 搭配其單一子網路。)

該帳戶是否具有至少包含一個子網路的 VPC？	您想要在帳戶中 AWS Cloud9 使用預設 VPC 的 VPC 是？	此 VPC 是否具有單一子網路？	執行此作業
是	是	否	針對 Subnet (子網路)，請選擇您希望 AWS Cloud9 在選取之預設 VPC 中使用的子網路。
是	否	是或否	針對網路 (VPC)，選擇 AWS Cloud9 您要使用的 VPC。針對子網路，選擇 AWS Cloud9 您要在該 VPC 中使用的子網路。

如需詳細資訊，請參閱《 AWS Cloud9 使用者指南》中的AWS Cloud9 開發環境的 Amazon VPC 設定。

3. 輸入環境名稱，並選擇性地新增環境描述。

> **Note**
>
> 每一名使用者的環境名稱必須是唯一的。

4. 若要變更未使用環境時 AWS Cloud9 關閉環境的預設期間，請展開節省成本設定，然後變更設定。
5. 選擇 Create environment (建立環境)。

若要開啟環境，請參閱開啟專案 AWS Cloud9 的環境。

您可以使用這些步驟來為專案建立一個以上的環境。例如，您可能想要使用一個環境來處理一部分的程式碼，並使用另一個環境處理具不同設定的相同程式碼的部分。

開啟專案 AWS Cloud9 的環境

請依照下列步驟，開啟您為 AWS CodeStar 專案建立的 AWS Cloud9 開發環境。

1. 在 AWS CodeStar 主控台中開啟專案時，在導覽列上選擇 IDE。

> **Important**
>
> 如果專案的原始碼存放在 GitHub 中，您就不會在導覽列上看到 IDE。不過，您可以使用 AWS Cloud9 主控台來開啟現有的環境。略過此程序的其餘部分，請參閱AWS Cloud9 《 使用者指南》和》中的開啟環境搭配 使用 GitHub AWS Cloud9。

2. 針對您的 AWS Cloud9 環境或共用 AWS Cloud9 環境，為您要開啟的環境選擇開啟 IDE。

您可以使用 AWS Cloud9 IDE 立即開始在專案的 AWS CodeCommit 儲存庫中使用程式碼。如需詳細資訊，請參閱AWS Cloud9 《 使用者指南》中的環境視窗、編輯器、標籤和窗格，以及AWS CodeCommit 《 使用者指南》中的「終端機」和「基本 Git 命令」。

與專案團隊成員共用 AWS Cloud9 環境

為 AWS CodeStar 專案建立 AWS Cloud9 開發環境後，您可以邀請 AWS 帳戶的其他使用者，包括專案團隊成員，來存取相同的環境。這非常適合用於配對程式設計，其中兩個程式設計師輪流編碼，並透過螢幕共用針對相同的程式碼提供建議，或坐在相同的工作站。環境成員可以使用共用的 AWS Cloud9 IDE 來查看程式碼編輯器中反白顯示的每個成員的程式碼變更，並在編碼時與其他成員進行文字聊天。

將團隊成員新增至專案並不會自動允許該成員參與專案的任何相關 AWS Cloud9 開發環境。若要邀請專案團隊成員存取專案的環境，您需要判斷正確的環境成員存取角色、將 AWS 受管政策套用至使用者，以及邀請使用者到您的環境。如需詳細資訊，請參閱《 AWS Cloud9 使用者指南》中的關於環境成員存取角色和邀請 IAM 使用者到您的環境。

當您邀請專案團隊成員存取專案的環境， AWS CodeStar 主控台會向該團隊成員顯示環境。環境會顯示在專案 AWS CodeStar 主控台中 IDE 索引標籤的共用環境清單中。若要顯示此清單，請讓團隊成員在主控台中開啟專案，然後在導覽列中選擇 IDE。

> **Important**
>
> 如果專案的原始碼存放在 GitHub 中，則不會在導覽列上看到 IDE。不過，您可以使用 AWS Cloud9 主控台邀請 AWS 您帳戶中的其他使用者，包括專案團隊成員，來存取環境。若要執行此操作，請參閱本指南搭配 使用 GitHub AWS Cloud9中的 ，並參閱《 AWS Cloud9 使用者指南》中的關於環境成員存取角色和邀請 IAM 使用者到您的環境。

您也可以邀請非專案團隊成員的使用者存取環境。例如，您可能希望使用者處理專案的程式碼，但沒有該專案的其他存取權。若要邀請這類使用者，請參閱《 AWS Cloud9 使用者指南》中的關於環境成員

存取角色和邀請 IAM 使用者到您的環境。當您邀請非專案團隊成員的使用者存取專案的環境時，該使用者可使用 AWS Cloud9 主控台來存取環境。如需詳細資訊，請參閱AWS Cloud9 《使用者指南》中的開啟環境。

從專案刪除 AWS Cloud9 環境

當您從中刪除專案及其所有 AWS 資源時 AWS CodeStar，也會刪除使用 AWS CodeStar 主控台建立的所有相關 AWS Cloud9 開發環境，且無法復原。您可以從專案刪除開發環境，但不刪除專案。

1. 在 AWS CodeStar 主控台中開啟專案時，在導覽列中選擇 IDE。

   > **Important**
   >
   > 如果專案的原始碼存放在 GitHub 中，您就不會在導覽列上看到 IDE。不過，您可以使用 AWS Cloud9 主控台來刪除開發環境。略過此程序的其餘部分，請參閱AWS Cloud9 《使用者指南》中的刪除環境。

2. 在 Cloud9 環境中選擇要刪除的環境，然後選擇刪除
3. 輸入 **`delete`** 以確認刪除開發環境，然後選擇刪除。

   > **Warning**
   >
   > 您無法恢復刪除後的開發環境。在環境中的所有未遞交的程式碼變更都會遺失。

搭配 使用 GitHub AWS Cloud9

對於在 GitHub 中存放其原始程式碼的 AWS CodeStar 專案， AWS CodeStar 主控台不支援直接使用 AWS Cloud9 開發環境。不過，您可以使用 AWS Cloud9 主控台來處理 GitHub 儲存庫中的原始程式碼。

1. 使用 AWS Cloud9 主控台建立 AWS Cloud9 開發環境。如需詳細資訊，請參閱AWS Cloud9 《使用者指南》中的建立環境。
2. 使用 AWS Cloud9 主控台開啟開發環境。如需詳細資訊，請參閱AWS Cloud9 《使用者指南》中的開啟環境。
3. 在 IDE 中，利用終端工作階段連接到 GitHub 儲存庫 (此程序稱為複製)。如果終端機工作階段停止執行，在 IDE 功能表列上選擇 Window, New Terminal (視窗、新增終端機)。如需用於複製 GitHub 儲存庫的命令，請參閱 GitHub 說明網站上的複製。

若要導覽至 GitHub 儲存庫的主頁面，在 AWS CodeStar 主控台中開啟專案時，在側邊導覽列上選擇程式碼。

4. 使用 IDE 中的 Environment (環境) 視窗及和編輯器標籤來檢視、變更和儲存程式碼。如需詳細資訊，請參閱AWS Cloud9 《 使用者指南》中的環境視窗和編輯器、標籤和窗格。
5. 使用 IDE 終端機工作階段中的 Git 推送程式碼變更至儲存庫，並且從儲存庫定期提取程式碼變更。如需詳細資訊，請參閱 GitHub Help 網站上的 Pushing to a Remote Respository 和 Fetching a Remote Repository。如需 Git 命令單，請參閱 GitHub Help 網站上的 Git Cheatsheet。

> **Note**
>
> 若要防止 Git 在您每次從儲存庫推送或提取程式碼時提示您輸入 GitHub 登入資料，您可以使用登入資料協助程式。如需詳細資訊，請參閱 GitHub Help 網站上的在 Git 中快取您的 GitHub 密碼。

其他資源

如需使用 的詳細資訊 AWS Cloud9，請參閱AWS Cloud9 《 使用者指南》中的下列內容：

- 教學課程
- 使用環境
- 使用 IDE
- 範例

搭配 使用 Eclipse AWS CodeStar

您可以使用 Eclipse 在 AWS CodeStar 專案中進行程式碼變更和開發軟體。您可以使用 Eclipse 編輯 AWS CodeStar 專案程式碼，然後將變更遞交並推送至 AWS CodeStar 專案的來源儲存庫。

> **Note**
>
> 本主題中的資訊僅適用於將原始程式碼存放在 CodeCommit 中的 AWS CodeStar 專案。如果您的 AWS CodeStar 專案將其原始程式碼存放在 GitHub 中，您可以使用 EGit for Eclipse 等工具。如需詳細資訊，請參閱 EGit 網站上的 EGit 文件。

如果 AWS CodeStar 專案將其原始程式碼存放在 CodeCommit 中，您必須安裝 AWS Toolkit for Eclipse 支援的 版本 AWS CodeStar。您也必須是具有擁有者或參與者角色的 AWS CodeStar 專案團隊成員。

若要使用 Eclipse，您還需要：

- 以團隊成員身分新增至 AWS CodeStar 專案的 IAM 使用者。
- 如果 AWS CodeStar 專案將其原始程式碼存放在 CodeCommit 中，則為 IAM 使用者的 Git 登入資料（登入登入資料）。
- 在本機 AWS Toolkit for Eclipse 電腦上安裝 Eclipse 和 的足夠許可。

主題

步驟 1：安裝 AWS Toolkit for Eclipse

Toolkit for Eclipse 是您可以新增至 Eclipse 的軟體套件。它的安裝和管理方式與 Eclipse 中的其他軟體套件的方式相同。 AWS CodeStar 工具組包含在 Toolkit for Eclipse 中。

使用 AWS CodeStar 模組安裝 Toolkit for Eclipse

1. 在您本機電腦安裝 Eclipse。支援的 Eclipse 版本包含 Luna、Mars 和 Neon。
2. 下載並安裝 Toolkit for Eclipse。如需詳細資訊，請參閱《AWS Toolkit for Eclipse 入門指南》。
3. 在 Eclipse 中，選擇說明，然後選擇安裝新軟體。
4. 在可用軟體中，選擇新增。
5. 在新增儲存庫中，選擇存檔、瀏覽到 .zip 檔案的儲存位置，並開啟檔案。將名稱留白，然後選擇確定。
6. 在可用軟體中，選擇全選以選取AWS 核心管理工具和開發人員工具，然後選擇下一步。
7. 在 Install Details (安裝詳細資訊) 中選擇 Next (下一步)。
8. 在 Review Licenses (審核授權) 中檢閱授權協議。選擇 I accept the terms of the license agreement (我接受授權合約的條款)，然後選擇完成。重新啟動 Eclipse。

步驟 2：將您的 AWS CodeStar 專案匯入 Eclipse

安裝 Toolkit for Eclipse 之後，您可以從 IDE 匯入 AWS CodeStar 專案並編輯、遞交和推送程式碼。

> **Note**
>
> 您可以將多個 AWS CodeStar 專案新增至 Eclipse 中的單一工作區，但您必須在從一個專案變更為另一個專案時更新您的專案登入資料。

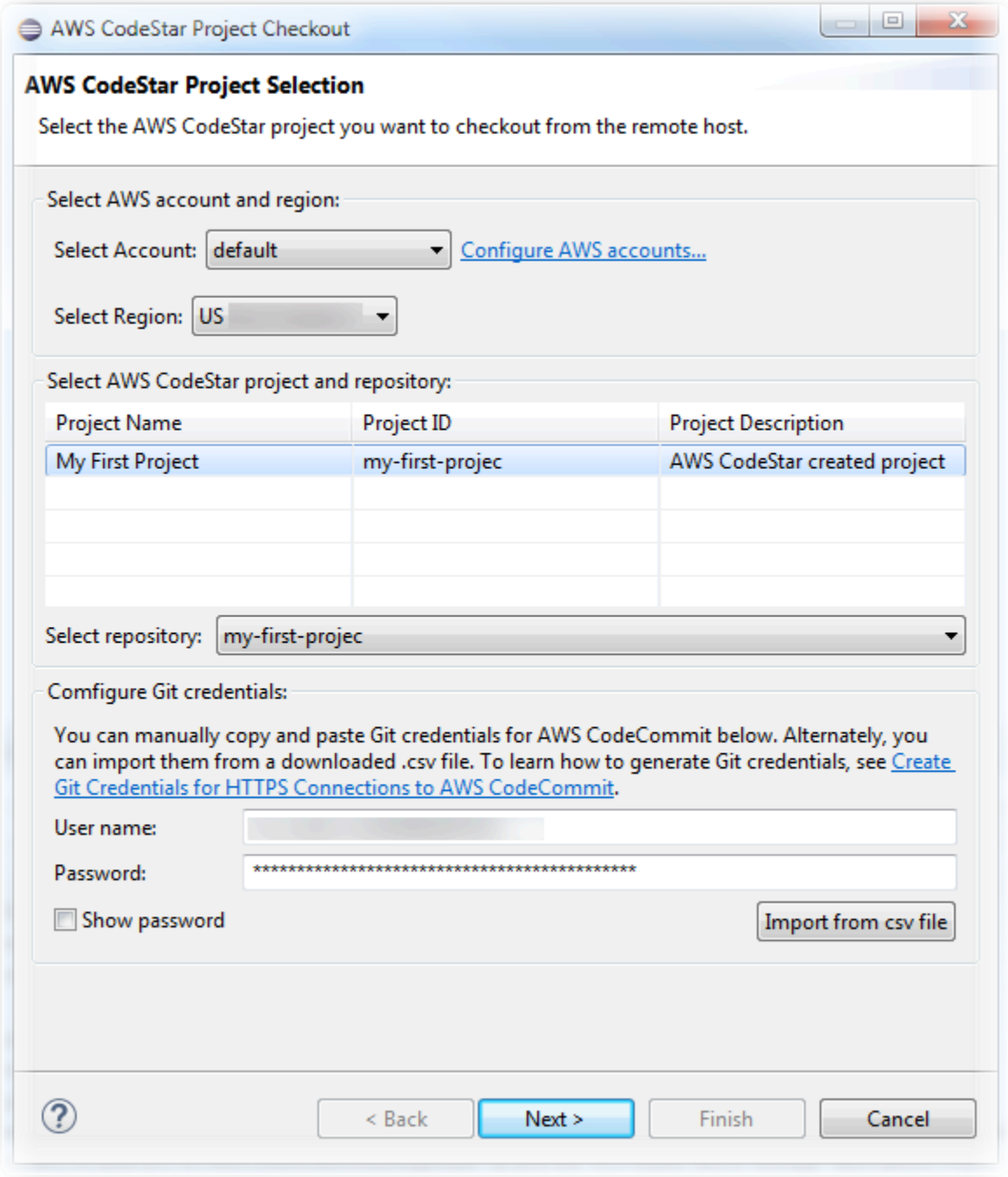
匯入 AWS CodeStar 專案

1. 從 AWS 功能表中，選擇匯入 AWS CodeStar 專案。或者，選擇檔案，然後選擇匯入。在選取中，展開 AWS，然後選擇 AWS CodeStar 專案。

   選擇 Next (下一步)。

2. 在AWS CodeStar 專案選擇中，選擇您的 AWS 設定檔和託管 AWS CodeStar 專案 AWS 的區域。如果您沒有在電腦上設定存取金鑰和私密金鑰的 AWS 設定檔，請選擇設定 AWS 帳戶並遵循指示。

   在選取 AWS CodeStar 專案和儲存庫中，選擇您的 AWS CodeStar 專案。在設定 Git 登入資料中，輸入您為存取專案的儲存庫而產生的登入登入資料。(如果您沒有 Git 登入資料，請參閱入門)。選擇 Next (下一步)。



3. 所有專案儲存庫的分支預設為已選定。如果您不想匯入一或多個分支，請清除方塊，然後選擇 Next (下一步)。
4. 在 Local Destination (本機目的地) 中，請選擇匯入精靈在您電腦上建立的本機儲存庫的目的地，然後選擇完成。
5. 在 Project Explorer 中，展開專案樹狀結構以瀏覽 AWS CodeStar 專案中的檔案。

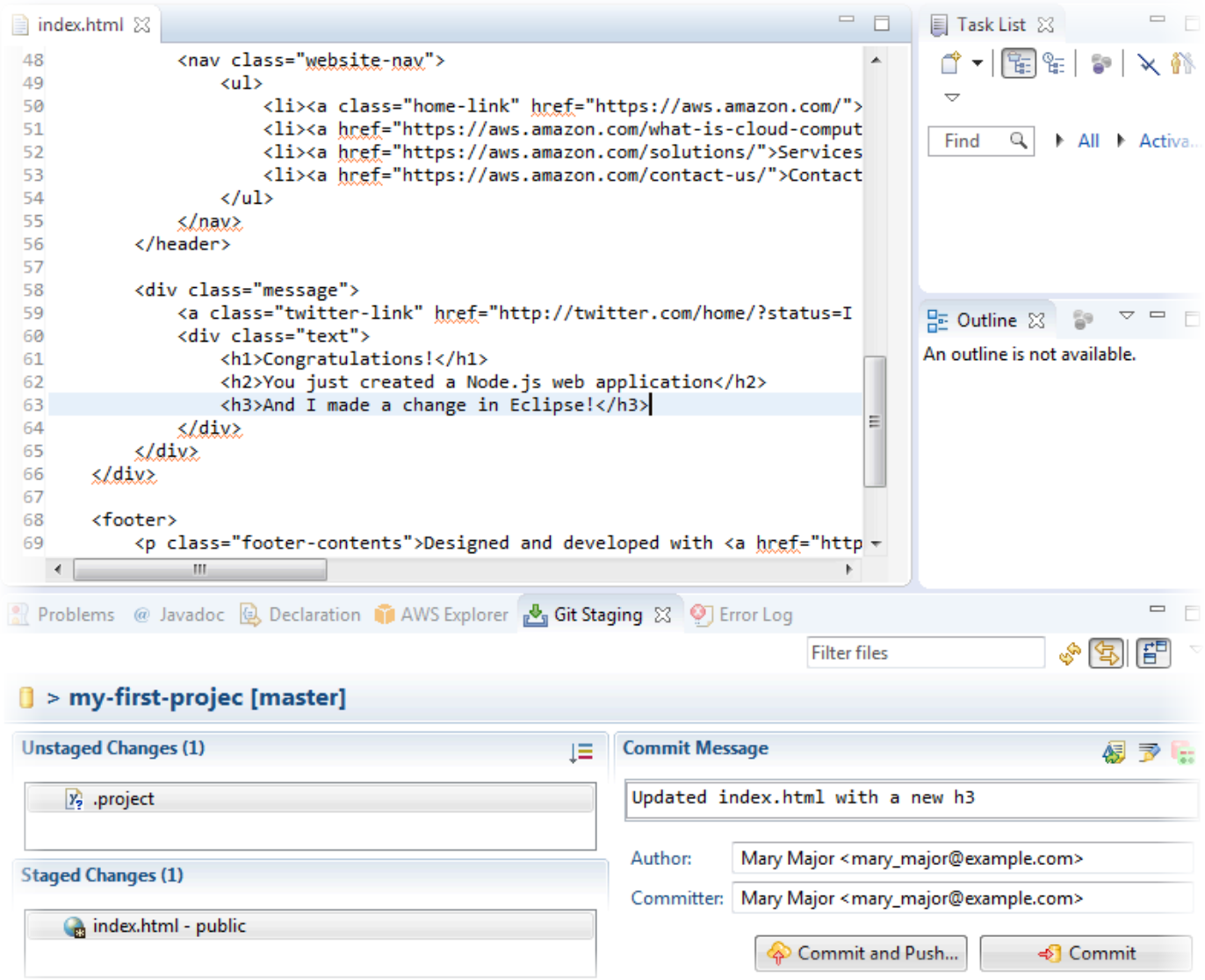
步驟 3：在 Eclipse 中編輯 AWS CodeStar 專案程式碼

將 AWS CodeStar 專案匯入 Eclipse 工作區後，您可以編輯專案的程式碼、儲存變更，以及將程式碼遞交並推送至專案的來源儲存庫。這使用 Eclipse 適用之 EGit 外掛程式，遵守任何 Git 儲存庫的相同程序。如需詳細資訊，請參閱 Eclipse 網站上的 EGit 使用者指南。

編輯專案程式碼，並對 AWS CodeStar 專案的來源儲存庫進行第一次遞交

1. 在 Project Explorer 中，展開專案樹狀結構以瀏覽 AWS CodeStar 專案中的檔案。
2. 編輯一或多個程式碼檔案並儲存變更。
3. 當您準備好認可變更，可開啟該檔案的內容功能表，選擇團隊，然後選擇遞交。

   如果已在您的專案檢視畫面開啟 Git Staging (Git 暫存) 視窗，您可以略過此步驟。
4. 在 Git Staging (Git 暫存) 中，透過將變更過的檔案移到 Staged Changes (暫存變更) 來暫存變更。在 Commit Message (遞交訊息) 中輸入遞交訊息，然後選擇 Commit and Push (遞交並推送)。



若要查看您的程式碼變更的部署情況，請返回您的專案儀表板。如需詳細資訊，請參閱步驟 3：檢視您的專案。

搭配 使用 Visual Studio AWS CodeStar

您可以使用 Visual Studio 在 AWS CodeStar 專案中進行程式碼變更和開發軟體。

> **Note**
>
> Visual Studio for Mac 不支援 AWS Toolkit，因此無法與 搭配使用 AWS CodeStar。
> 本主題中的資訊僅適用於將原始程式碼存放在 CodeCommit 中的 AWS CodeStar 專案。如果您的 AWS CodeStar 專案將其原始程式碼存放在 GitHub 中，您可以使用 GitHub Extension for

Visual Studio 等工具。如需詳細資訊，請參閱 GitHub Extension for Visual Studio 網站上的概觀頁面，以及 GitHub 網站上的 GitHub for Visual Studio 入門。

若要使用 Visual Studio 在 AWS CodeStar 專案的來源儲存庫中編輯程式碼，您必須安裝 AWS Toolkit for Visual Studio 支援的 版本 AWS CodeStar。您必須是具備擁有者參與者角色的 AWS CodeStar 專案團隊成員。

若要使用 Visual Studio，您還需要：

- 以團隊成員身分新增至 AWS CodeStar 專案的 IAM 使用者。
- AWS IAM 使用者的 憑證 （ 例如，您的存取金鑰和私密金鑰 ）。
- 在本機 AWS Toolkit for Visual Studio 電腦上安裝 Visual Studio 和 的足夠許可。

Toolkit for Visual Studio 是您可以新增至 Visual Studio 的軟體套件。其安裝和管理方式與 Visual Studio 中的其他軟體套件相同。

使用 AWS CodeStar 模組安裝 Toolkit for Visual Studio，並設定對專案儲存庫的存取

1. 在本機電腦上安裝 Visual Studio。
2. 下載並安裝 Toolkit for Visual Studio，並將 .zip 檔案儲存至本機資料夾或目錄。在開始使用頁面上 AWS Toolkit for Visual Studio，輸入或匯入您的 AWS 登入資料，然後選擇儲存並關閉。
3. 在 Visual Studio 中，開啟 Team Explorer。在託管服務供應商中，尋找 CodeCommit，然後選擇連線。
4. 在 Manage Connections (管理連線) 中，選擇 Clone (複製)。選擇專案的儲存庫，以及您要將儲存庫複製到本機電腦上的資料夾，然後選擇 OK (確定)。
5. 如果提示您建立 Git 登入資料，請選擇 Yes (是)。工具組會嘗試替您建立登入資料。將登入資料檔案儲存在安全的位置。這是您必須儲存這些登入資料的唯一機會。如果工具組無法替您建立登入資料，或您選擇 No (否)，則您必須建立並提供自己的 Git 登入資料。如需詳細資訊，請參閱設定您的電腦以確認變更 (IAM 使用者)，或依照線上指示進行。

   完成複製專案後，您就可以開始在 Visual Studio 中編輯程式碼，並在 CodeCommit 中遞交變更並推送至專案的儲存庫。

變更 專案中的 AWSAWS CodeStar 資源

在 中建立專案後 AWS CodeStar，您可以變更 AWS CodeStar 新增至專案的預設 AWS 資源集。

支援的資源變更

下表列出 專案中 AWS CodeStar 預設 AWS 資源的支援變更。

變更	備註
將階段新增至 AWS CodePipeline。	請參閱 將階段新增至 AWS CodePipeline。
變更 Elastic Beanstalk 環境設定。	請參閱 變更 AWS Elastic Beanstalk 環境設定。
在 Amazon API Gateway 中變更 AWS Lambda 函數的程式碼或設定、IAM 角色或其 API。Amazon API Gateway	請參閱 變更來源碼中的 AWS Lambda 函數。
將資源新增至 AWS Lambda 專案，並展開許可以建立和存取新資源。	請參閱 新增資源到專案。
使用 AWS Lambda 函數的 CodeDeploy 新增流量轉移。	請參閱 轉移 AWS Lambda 專案的流量。
新增 AWS X-Ray 支援	請參閱 啟用專案的追蹤。
編輯專案的 buildspec.yml 檔案，以新增單元測試建置階段 AWS CodeBuild 供 執行。	請參閱無伺服器專案教學課程中的步驟 7：新增單元測試到 Web 服務。
將您自己的 IAM 角色新增至您的專案中。	請參閱 將 IAM 角色新增至專案。
變更 IAM 角色定義。	針對應用程式堆疊中定義的角色。您無法變更工具鏈或 AWS CloudFormation 堆疊中定義的角色。
修改 Lambda 專案以新增端點。	
修改 EC2 專案以新增端點。	
修改 Elastic Beanstalk 專案以新增端點。	

變更	備註
編輯專案以新增生產階段和端點。	請參閱 新增生產階段和端點至專案。
在 AWS CodeStar 專案中安全地使用 SSM 參數。	請參閱 the section called “在 AWS CodeStar 專案中安全地使用 SSM 參數”。

不支援下列變更。

- 切換到不同的部署目標 （ 例如，部署到 AWS Elastic Beanstalk 而非 AWS CodeDeploy)。
- 新增適用的 Web 端點名稱。
- 變更 CodeCommit 儲存庫名稱 （ 適用於連線至 CodeCommit 的 AWS CodeStar 專案 ）。
- 對於連線至 GitHub 的 AWS CodeStar 專案，請中斷連接 GitHub 儲存庫，然後將儲存庫重新連線至該專案，或將任何其他儲存庫連線至該專案。您可以使用 CodePipeline 主控台 （ 而非主控台 ），AWS CodeStar 在管道的來源階段中斷連線並重新連線至 GitHub。不過，如果您將來源階段重新連線至不同的 GitHub 儲存庫，在專案的 AWS CodeStar 儀表板中，儲存庫和問題圖磚中的資訊可能錯誤或過時。中斷連接 GitHub 儲存庫並不會從遞交歷史記錄中移除該儲存庫的資訊，而且 GitHub 會在 AWS CodeStar 專案儀表板中發出圖磚。若要移除此資訊，請使用 GitHub 網站來停用專案對 GitHub 的存取 AWS CodeStar 。若要撤銷存取，可在 GitHub 網站上使用您的 GitHub 帳戶設定檔頁面的 Authorized OAuth Apps (已授權的 OAuth 應用程式) 區段。
- 中斷 CodeCommit 儲存庫的連線 （ 適用於連線至 CodeCommit 的 AWS CodeStar 專案 ），然後將儲存庫重新連線至該專案，或將任何其他儲存庫連線至該專案。

將階段新增至 AWS CodePipeline

您可以將新階段新增至在專案中 AWS CodeStar 建立的管道。如需詳細資訊，請參閱AWS CodePipeline 《 使用者指南》中的在 中編輯管道 AWS CodePipeline。

> **Note**
>
> 如果新階段取決於任何 AWS CodeStar 未建立 AWS 的資源，管道可能會中斷。這是因為 AWS CodeStar 建立的 IAM 角色預設 AWS CodePipeline 可能無法存取這些資源。
> 若要嘗試提供 AWS CodeStar 未建立 AWS 之資源的 AWS CodePipeline 存取權，您可能想要變更 AWS CodeStar 建立的 IAM 角色。這不受支援，因為 AWS CodeStar 可能會在對專案執行定期更新檢查時移除您的 IAM 角色變更。

變更 AWS Elastic Beanstalk 環境設定

您可以變更在專案中 AWS CodeStar 建立的 Elastic Beanstalk 環境設定。例如，您可能想要將 AWS CodeStar 專案中的預設 Elastic Beanstalk 環境從單一執行個體變更為負載平衡。若要執行此操作，請編輯專案儲存庫中的 `template.yml` 檔案。您可能還需要變更專案工作者角色的許可。在您推送範本變更後， AWS CodeStar 並為您 AWS CloudFormation 佈建資源。

如需編輯 `template.yml` 檔案的詳細資訊，請參閱 使用 Template.yml 檔案變更應用程式資源。如需 Elastic Beanstalk 環境的詳細資訊，請參閱《 AWS Elastic Beanstalk 開發人員指南》中的AWS Elastic Beanstalk 環境管理主控台。

變更來源碼中的 AWS Lambda 函數

您可以變更 Lambda 函數的程式碼或設定，或是其在專案中 AWS CodeStar 建立的 IAM 角色或 API Gateway API。若要這樣做，建議您使用 AWS 無伺服器應用程式模型 (AWS SAM) 搭配專案 CodeCommit 儲存庫中的 `template.yaml` 檔案。此`template.yaml`檔案定義函數的名稱、處理常式、執行時間、IAM 角色和 API 在 API Gateway 中。如需詳細資訊，請參閱 GitHub 網站上的如何使用 AWS SAM 建立無伺服器應用程式。

啟用專案的追蹤

AWS X-Ray 提供追蹤功能，可用來分析分散式應用程式的效能行為 （例如回應時間的延遲）。將追蹤新增至 AWS CodeStar 專案後，您可以使用 AWS X-Ray 主控台來檢視應用程式檢視和回應時間。

> **Note**
>
> 您可以對以下專案使用這些步驟，使用下列建立的專案支援變更：
>
> - 任何 Lambda 專案。
> - 對於 2018 年 8 月 3 日之後建立的 Amazon EC2 或 Elastic Beanstalk 專案， 已在專案儲存庫中佈建`/template.yml`檔案。 AWS CodeStar

每個 AWS CodeStar 範本都包含一個 AWS CloudFormation 檔案，可建立應用程式 AWS 執行時間相依性的模型，例如資料庫資料表和 Lambda 函數。此檔案存放於檔案 `/template.yml` 中的來源儲存庫。

您可以將 AWS X-Ray 資源新增至 Resources區段，以修改此檔案來新增追蹤。然後，您可以修改專案的 IAM 許可， AWS CloudFormation 以允許 建立資源。如需範本元素和格式的相關資訊，請參閱AWS 資源類型參考。

這些是自訂範本時可遵循的高階步驟。

1. 步驟 1：編輯 IAM 中的工作者角色以進行追蹤
2. 步驟 2：修改 template.yml 檔案以進行追蹤
3. 步驟 3：遞交和推送您的範本變更以進行追蹤
4. 步驟 4：監視 AWS CloudFormation 堆疊更新以進行追蹤

步驟 1：編輯 IAM 中的工作者角色以進行追蹤

您必須以系統管理員身分登入，才能執行步驟 1 和 4。此步驟顯示編輯 Lambda 專案許可的範例。

> **Note**
>
> 如果您的專案是使用許可界限政策來佈建，則可以略過此步驟。
> 對於 2018 年 12 月 6 日 PDT 之後建立的專案， 會使用許可界限政策 AWS CodeStar 佈建您的專案。

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/codestar/ 開啟 AWS CodeStar 主控台。
2. 建立專案或選擇使用 `template.yml file` 的現有專案，然後開啟 Project resources (專案資源) 頁面。
3. 在專案資源，尋找資源清單中為 CodeStarWorker/Lambda 角色建立的 IAM 角色。該角色名稱遵循此格式：`role/CodeStarWorker-`*`Project_name`*`-lambda-`*`Function_name`*。選擇角色的 ARN。
4. 在 IAM 主控台開啟該角色。選擇連接政策。搜尋 `AWSXrayWriteOnlyAccess` 政策，選取旁邊的方框，然後選擇 Attach Policy (連接政策)。

步驟 2：修改 template.yml 檔案以進行追蹤

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。

2. 選擇您的無伺服器專案，然後開啟程式碼頁面。在儲存庫的最上層，尋找和編輯 `template.yml` 檔案。在 `Resources` 下方將資源貼到 `Properties` 部分。

```
      Tracing: Active
```

此範例顯示修改過的範本：

```
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
```

步驟 3：遞交和推送您的範本變更以進行追蹤

- 遞交和推送 `template.yml` 檔案中的變更。

> **Note**
>
> 此會啟動您的管道。如果您在更新 IAM 許可之前遞交變更，您的管道會啟動、 AWS CloudFormation 堆疊更新發生錯誤，且堆疊更新會復原。如果發生這種情況，請修正權限，然後重新啟動您的管道。

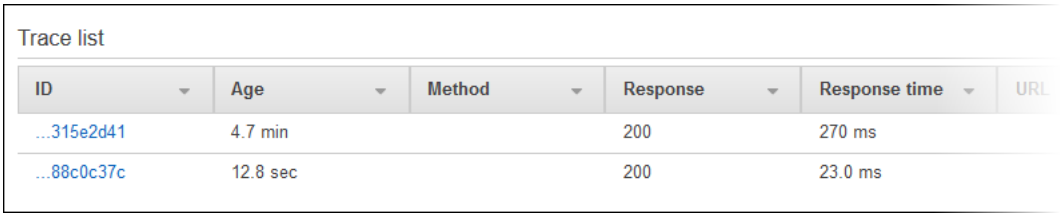
步驟 4：監視 AWS CloudFormation 堆疊更新以進行追蹤

1. 當專案的管道啟動部署階段時， AWS CloudFormation 堆疊更新就會開始。若要查看堆疊更新的狀態，請在 AWS CodeStar 儀表板上選擇管道中的 AWS CloudFormation 階段。

   如果 中的堆疊更新 AWS CloudFormation 傳回錯誤，請參閱 中的疑難排解準則AWS CloudFormation：遺失許可的回復建立堆疊。如果工作者角色遺漏許可，請編輯連接到您專案的 Lambda 工作者角色的政策。請參閱 步驟 1：編輯 IAM 中的工作者角色以進行追蹤。

2. 使用儀表板檢視成功完成的管道。您的應用程式現已啟用追蹤功能。

3. 在 Lambda 主控台檢視您的函數詳細資訊，確認追蹤功能已啟用。
4. 選擇專案的應用程式端點。與您應用程式的這項互動會被追蹤。您可以檢視 AWS X-Ray 主控台中的追蹤資訊。



新增資源到專案

所有專案的每個 AWS CodeStar 範本都隨附 AWS CloudFormation 檔案，可建立應用程式的 AWS 執行時間相依性模型，例如資料庫資料表和 Lambda 函數。此檔案存放於檔案 `/template.yml` 中的來源儲存庫。

> **Note**
>
> 您可以對以下專案使用這些步驟，使用下列建立的專案支援變更：
>
> - 任何 Lambda 專案。
> - 對於 2018 年 8 月 3 日之後建立的 Amazon EC2 或 Elastic Beanstalk 專案， 已在專案儲存庫中佈建`/template.yml`檔案。 AWS CodeStar

您可以透過將 AWS CloudFormation 資源新增至 `Resources`區段來修改此檔案。修改 `template.yml` 檔案可讓 AWS CodeStar 和 將新資源 AWS CloudFormation 新增至您的專案。有些資源要求您新增其他許可至您專案的 CloudFormation 工作者角色的政策。如需範本元素和格式的相關資訊，請參閱AWS 資源類型參考。

在判斷哪些資源必須新增到專案之後，需遵循這些高階步驟來自訂範本。如需 AWS CloudFormation 資源及其所需屬性的清單，請參閱AWS 資源類型參考。

1. 步驟 1：編輯 IAM 中的 CloudFormation 工作者角色 (如果需要)
2. 步驟 2：修改 template.yml 檔案
3. 步驟 3：遞交和推送您的範本變更
4. 步驟 4：監視 AWS CloudFormation 堆疊更新
5. 步驟 5：在資源許可新增內嵌政策

使用本節中的步驟來修改您的 AWS CodeStar 專案範本以新增資源，然後在 IAM 中展開專案的 CloudFormation 工作者角色許可。在本範例中，AWS:: SQS:: Queue 資源會加入到 `template.yml` 檔案。變更會在 中啟動自動回應 AWS CloudFormation ，將 Amazon Simple Queue Service 佇列新增至您的專案。

步驟 1：編輯 IAM 中的 CloudFormation 工作者角色

您必須以系統管理員身分登入，才能執行步驟 1 和 5。

> **Note**
>
> 如果您的專案是使用許可界限政策來佈建，則可以略過此步驟。
> 對於 2018 年 12 月 6 日 PDT 之後建立的專案， 會使用許可界限政策 AWS CodeStar 佈建您的專案。

1. 登入 AWS Management Console 並開啟 AWS CodeStar 主控台，網址為 https：//https://console.aws.amazon.com/codestar/。
2. 建立專案或選擇使用 `template.yml file` 的現有專案，然後開啟 Project resources (專案資源) 頁面。
3. 在專案資源下，找到資源清單中為 CodeStarWorker/AWS CloudFormation 角色建立的 IAM 角色。該角色名稱遵循此格式：`role/CodeStarWorker-`*`Project_name`*`-CloudFormation`。
4. 在 IAM 主控台開啟該角色。在 Permissions (許可) 標籤上，擴展 Inline Policies (內嵌政策)的服務角色政策列，並選擇 Edit Policy (編輯政策)。
5. 選擇 JSON 標籤以編輯政策。

   > **Note**
   >
   > 連接至工作者角色的政策是 `CodeStarWorkerCloudFormationRolePolicy`。

6. 在 JSON 欄位中，在 `Statement` 元素中新增下列政策聲明：

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

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

7. 選擇 Review policy (檢閱政策) 以確保政策沒有包含錯誤，然後選擇 Save changes (儲存變更)。

步驟 2：修改 template.yml 檔案

1. 在 https：//https://console.aws.amazon.com/codestar/ 開啟 AWS CodeStar 主控台。
2. 選擇您的無伺服器專案，然後開啟程式碼頁面。在最高階的儲存庫中，記下 `template.yml` 的位置。
3. 使用 IDE、主控台或本機儲存庫的命令列來編輯儲存庫中的 `template.yml` 檔案。將資源貼到 `Resources` 部分。在本範例中，當以下文字被複製，便會新增 `Resources` 部分。

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

此範例顯示修改過的範本：

```
Resources:
  HelloWorld:
    Type: AWS::Serverless::Function
    Properties:
      Handler: index.handler
      Runtime: python3.6
      Role:
        Fn::ImportValue:
          !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
      Events:
        GetEvent:
          Type: Api
          Properties:
            Path: /
            Method: get
        PostEvent:
          Type: Api
          Properties:
            Path: /
            Method: post

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

步驟 3：遞交和推送您的範本變更

- 遞交和推送在步驟 2 儲存的 `template.yml` 檔案中的變更。

> **Note**
>
> 此會啟動您的管道。如果您在更新 IAM 許可之前遞交變更，您的管道會啟動，且 AWS CloudFormation 堆疊更新發生錯誤，導致堆疊更新復原。如果發生這種情況，請修正權限，然後重新啟動您的管道。

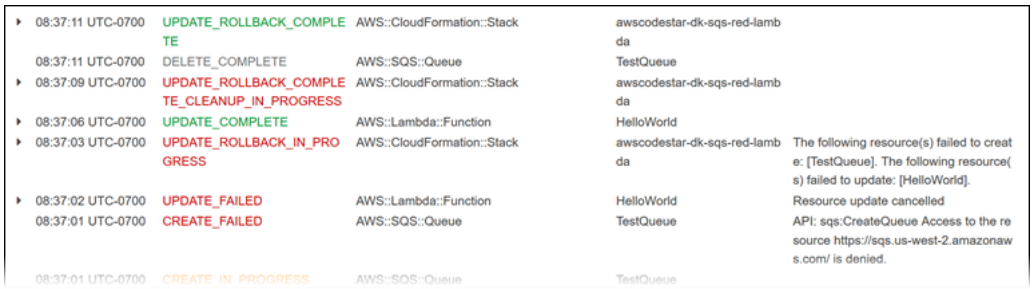
步驟 4：監視 AWS CloudFormation 堆疊更新

1. 當您專案的管道啟動部署階段時， AWS CloudFormation 堆疊更新就會開始。您可以在 AWS CodeStar 儀表板上選擇管道中的 AWS CloudFormation 階段，以查看堆疊更新。

   故障診斷：

   如果所需的資源許可權遺失，堆疊更新會失敗。在 AWS CodeStar 儀表板檢視中檢視專案管道的失敗狀態。

   選擇管道部署階段中的 CloudFormation 連結，在 AWS CloudFormation 主控台中排除故障。在主控台的 Events (事件) 清單中，選擇您的專案以檢視堆疊建立詳細資訊。有一個訊息顯示故障詳細資訊。在此範例中，`sqs:CreateQueue` 許可遺失。



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	

   透過編輯連接至專案 AWS CloudFormation 工作者角色的政策，新增任何缺少的許可。請參閱 步驟 1：編輯 IAM 中的 CloudFormation 工作者角色。

2. 成功執行您的管道之後， AWS CloudFormation 堆疊中會建立資源。在 中的資源清單中 AWS CloudFormation，檢視為您的專案建立的資源。在此範例中，TestQueue 佇列會列在資源部分。

   佇列 URL 可在 中使用 AWS CloudFormation。佇列 URL 遵循以下格式：

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

如需詳細資訊，請參閱傳送 Amazon SQS 訊息、接收來自 Amazon SQS 佇列的訊息，以及刪除來自 Amazon SQS 佇列的訊息。

步驟 5：在資源許可新增內嵌政策

授予團隊成員存取您的新資源的權限，做法是新增適當的內嵌政策並加入到使用者的角色。並不是所有資源都需要您新增許可。若要執行下列步驟，您必須以根使用者、 帳戶中的管理員使用者，或具有 AdministratorAccess 受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

若要使用 JSON 政策編輯器來建立政策

1. 登入 AWS Management Console ，並在 https://console.aws.amazon.com/iam/：//www. 開啟 IAM 主控台。
2. 在左側的導覽窗格中，選擇 Policies (政策)。

   如果這是您第一次選擇 Policies (政策)，將會顯示 Welcome to Managed Policies (歡迎使用受管政策) 頁面。選擇 Get Started (開始使用)。
3. 在頁面頂端，選擇 Create policy (建立政策)。
4. 在政策編輯器中，選擇 JSON 選項。
5. 輸入下列 JSON 政策文件：

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

6. 選擇 Next (下一步)。

> **Note**
>
> 您可以隨時切換視覺化與 JSON 編輯器選項。不過，如果您進行變更或在視覺化編輯器中選擇下一步，IAM 就可能會調整您的政策結構，以便針對視覺化編輯器進行最佳化。如需詳細資訊，請參閱 IAM 使用者指南中的調整政策結構。

7. 在檢視與建立頁面上，為您正在建立的政策輸入政策名稱與描述 (選用)。檢視此政策中定義的許可，來查看您的政策所授予的許可。
8. 選擇 Create policy (建立政策) 儲存您的新政策。

將 IAM 角色新增至專案

自 2018 年 12 月 6 日起，您可以在應用程式堆疊 (template.yml) 中定義自己的角色和政策。若要降低權限提升和破壞性動作的風險，您必須為所建立的每個 IAM 實體設定專案特定的許可界限。如果您有具有多個函數的 Lambda 專案，最佳實務是為每個函數建立 IAM 角色。

將 IAM 角色新增至專案

1. 編輯您的專案的 `template.yml` 檔案。
2. 在 `Resources:` 區段，使用以下範例中的格式來新增您的 IAM 資源：

```
  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. 透過管道釋出您的變更，並驗證成功執行。

新增生產階段和端點至專案

使用本節中的程序來將新的生產 (Prod) 階段新增至您的管道，以及在管道的部署和生產階段之間新增手動核准階段。這樣會在您的專案管道執行時建立額外的資源堆疊。

> **Note**
>
> 您可以在以下情況中使用這些程序：
>
> - 對於 2018 年 8 月 3 日之後建立的專案， 會使用專案儲存庫中的`/template.yml`檔案 AWS CodeStar 佈建您的 Amazon EC2、Elastic Beanstalk 或 Lambda 專案。
> - 對於 2018 年 12 月 6 日 PDT 之後建立的專案， 會使用許可界限政策 AWS CodeStar 佈建您的專案。

所有 AWS CodeStar 專案都使用 AWS CloudFormation 範本檔案來建立應用程式的 AWS 執行時間相依性模型，例如 Linux 執行個體和 Lambda 函數。`/template.yml` 檔案存放於來源儲存庫中。

在 `/template.yml` 檔案中，使用 `Stage` 參數來將資源堆疊新增至專案管道中的新階段。

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

`Stage` 參數會套用到具有資源中參考的專案 ID 的所有指定資源。例如，以下角色名稱是範本中指定的資源：

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

先決條件

使用 AWS CodeStar 主控台中的範本選項來建立專案。

請確定您的 IAM 使用者具有下列許可：

- `iam:PassRole` 專案 AWS CloudFormation 角色上的 。

- 專案工具鏈角色上的 `iam:PassRole`。
- `cloudformation:DescribeStacks`
- `cloudformation:ListChangeSets`

僅適用於 Elastic Beanstalk 或 Amazon EC2 專案：

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

主題

步驟 1：在 CodeDeploy 中建立新的部署群組（僅限 Amazon EC2 專案）

您可以選擇 CodeDeploy 應用程式，然後新增與新執行個體相關聯的新部署群組。

> **Note**
>
> 如果您的專案是 Lambda 或 Elastic Beanstalk 專案，您可以略過此步驟。

1. 開啟 CodeDeploy 主控台，網址為 https：//https://console.aws.amazon.com/codedeploy。
2. 選擇建立專案時為專案產生的 CodeDeploy 應用程式 AWS CodeStar。
3. 在 Deployment groups (部署群組) 下方，選擇 Create deployment group (建立部署群組)。
4. 在 Deployment group name (部署群組名稱) 中，輸入 ***<project-id>-prod-Env***。
5. 在服務角色中，為您的 AWS CodeStar 專案選擇工具鏈工作者角色。

6. 在 Deployment type (部署類型) 下，選擇 In-place (就地進行)。
7. 在 Environment configuration (環境組態) 下，選擇 Amazon EC2 Instances (Amazon EC2 執行個體) 索引標籤。
8. 在標籤群組下，於 Key (金鑰) 下，選擇 `aws:cloudformation:stack-name`。在 Value (值) 下，選擇 `awscodestar-<projectid>-infrastructure-prod` (要為 GenerateChangeSet 動作建立的堆疊)。
9. 在 Deployment settings (部署設定) 中，選擇 `CodeDeployDefault.AllAtOnce`。
10. 清除 Choose a load balancer (選擇負載平衡器)。
11. 選擇 Create deployment group (建立部署群組)。

    現在您已建立第二個部署群組。

步驟 2：新增生產階段的管道階段

使用與您的專案的部署階段使用相同的部署動作來新增階段。例如，Amazon EC2 專案的新生產階段應具有與為專案建立的部署階段相同的動作。

從部署階段複製參數和欄位

1. 從 AWS CodeStar 專案儀表板中，選擇管道詳細資訊，以在 CodePipeline 主控台中開啟管道。
2. 選擇編輯。
3. 在部署階段中，選擇 Edit stage (編輯階段)。
4. 在 GenerateChangeSet 動作上選擇編輯圖示。在下列欄位中記下這些值。建立新動作時，您會使用這些值。

   - Stack name (堆疊名稱)
   - Change set name (變更組合名稱)
   - Template (範本)
   - Template configuration (範本組態)
   - Input artifacts (輸入成品)
5. 展開 Advanced (進階)，然後在 Parameters (參數) 中，複製您的專案的參數。您可以將這些參數貼上到新動作中。例如，複製此處以 JSON 格式顯示的參數：

   - Lambda 專案：

```
{
```

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

- Amazon EC2 專案：

```
{

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

- Elastic Beanstalk 專案：

```
{
  "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. 在階段編輯窗格中，選擇 Cancel (取消)。

在新生產階段中建立 GenerateChangeSet 動作

> **Note**
>
> 當您新增新的動作但仍處於編輯模式時，如果您重新開啟新的編輯動作，某些欄位可能不會顯示。您可能也會看到下列訊息：堆疊 stack-name 不存在
> 此錯誤不會阻止您儲存管道。不過，若要還原遺失的欄位，您必須刪除新的動作，並再次新增。儲存並執行管道後，就能識別堆疊，不會再次出現錯誤。

1. 如果您的管道尚未顯示，請從 AWS CodeStar 專案儀表板選擇管道詳細資訊，以在主控台中開啟管道。
2. 選擇編輯。
3. 在圖表的底部，選擇 + Add stage (+ 新增階段)。
4. 輸入階段名稱 (例如，**Prod**)，然後選擇 + Add action group (+ 新增動作群組)。
5. 在 Action name (動作名稱) 中，輸入名稱 (例如，**GenerateChangeSet**)。
6. 在動作提供者中，選擇 AWS CloudFormation。
7. 在 Action mode (動作模式) 中，選擇 Create or replace a change set (建立或取代變更集)。
8. 在堆疊名稱中，輸入要由此動作建立之 AWS CloudFormation 堆疊的新名稱。名稱開頭與部署堆疊名稱完全相同，然後加上 **-prod**：

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

   > **Note**
   >
   > 堆疊名稱的開頭必須確切為 **awscodestar-<project_name>-**，否則堆疊建立會失敗。

9. 在 Change set name (變更組合名稱) 中，輸入現有的部署階段中所提供相同的變更組合名稱 (例如，**pipeline-changeset**)。
10. 在 Input artifacts (輸入成品) 中，選擇建置成品。
11. 在 Template (範本) 中，輸入現有的部署階段中所提供相同的範本名稱 (例如，**<project-ID>-BuildArtifact::template.yml**)。
12. 在 Template configuration (範本組態) 中，輸入部署階段中所提供的相同變更範本組態檔案名稱 (例如，**<project-ID>-BuildArtifact::template-configuration.json**)。
13. 在 Capabilities (功能) 欄位中，選擇 CAPABILITY_NAMED_IAM。
14. 在 Role name (角色名稱) 中，選擇您的專案的 AWS CloudFormation 工作者角色名稱。
15. 展開 Advanced (進階)，然後在 Parameters (參數) 中，貼上您的專案的參數。針對 Amazon EC2 專案，包含此處以 JSON 格式顯示的 `Stage` 參數：

```
{

  "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**
>
> 務必貼上專案的所有參數，而不只是新參數或您想要變更的參數。

16. 選擇 Save (儲存)。
17. 在 AWS CodePipeline 窗格中，選擇儲存管道變更，然後選擇儲存變更。

> **Note**
>
> 訊息可能會顯示，通知您正在刪除和新增變更偵測資源。確認訊息並繼續本教學課程的下一個步驟。

檢視已更新的管道。

在新生產階段中建立 ExecuteChangeSet 動作

1. 如果您尚未檢視管道，請從 AWS CodeStar 專案儀表板選擇管道詳細資訊，以在主控台中開啟管道。
2. 選擇編輯。
3. 在您的新生產階段中，於新的 GenerateChangeSet 動作後，選擇 + Add action group (+ 新增動作群組)。
4. 在 Action name (動作名稱) 中，輸入名稱 (例如，**`ExecuteChangeSet`**)。
5. 在動作提供者中，選擇 AWS CloudFormation。
6. 在 Action mode (動作模式) 中，選擇 Execute a change set (執行變更組合)。
7. 在堆疊名稱中，輸入您在 GenerateChangeSet 動作中輸入的 AWS CloudFormation 堆疊新名稱（例如，**`awscodestar-<project-ID>-infrastructure-prod`**)。

8. 在變更集名稱中，輸入部署階段中使用的相同變更集名稱（例如，**pipeline-changeset**）。
9. 選擇完成。
10. 在 AWS CodePipeline 窗格中，選擇儲存管道變更，然後選擇儲存變更。

> **Note**
>
> 訊息可能會顯示，通知您正在刪除和新增變更偵測資源。確認訊息並繼續本教學課程的下一個步驟。

檢視已更新的管道。

在新的生產階段建立 CodeDeploy 部署動作（僅限 Amazon EC2 專案）

1. 在您的生產階段中的新動作之後，選擇 + Action (+ 動作)。
2. 在 Action name (動作名稱) 中，輸入名稱 (例如，**Deploy**)。
3. 在動作提供者中，選擇 AWS CodeDeploy。
4. 在應用程式名稱中，為您的專案選擇 CodeDeploy 應用程式的名稱。
5. 在部署群組中，選擇您在步驟 2 中建立的新 CodeDeploy 部署群組名稱。
6. 在 Input artifacts (輸入成品) 中，選擇在現有的階段中使用的相同建置成品。
7. 選擇完成。
8. 在 AWS CodePipeline 窗格中，選擇儲存管道變更，然後選擇儲存變更。檢視已更新的管道。

步驟 3：新增手動核准階段

最佳實務是在新生產階段的前端新增手動核准階段。

1. 在左上角，選擇 Edit (編輯)。
2. 在管道圖表中，於部署和生產部署階段之間，選擇 + Add stage (+ 新增階段)。
3. 在 Edit stage (編輯階段) 上，輸入階段名稱 (例如，**Approval**)，然後選擇 + Add action group (+ 新增動作群組)。
4. 在 Action name (動作名稱) 中，輸入名稱 (例如，**Approval**)。
5. 在 Approval type (核准類型) 中，選擇 Manual approval (手動核准)。

6. (選用) 在 Configuration (組態) 下，於 SNS Topic ARN (SNS 主題 ARN) 中，選擇您已建立和訂閱的 SNS 主題。
7. 選擇 Add Action (新增動作)。
8. 在 AWS CodePipeline 窗格中，選擇儲存管道變更，然後選擇儲存變更。檢視已更新的管道。
9. 若要提交您的變更並啟動管道建置，請選擇 Release change (發行變更)，然後選擇 Release (發行)。

步驟 4：推送變更並監控 AWS CloudFormation 堆疊更新

1. 當您的管道執行時，您可以使用此處的步驟來遵循新階段的堆疊和端點建立。
2. 當管道啟動部署階段時， AWS CloudFormation 堆疊更新就會開始。您可以在 AWS CodeStar 儀表板上選擇管道中的 AWS CloudFormation 階段，以查看堆疊更新通知。若要檢視堆疊建立詳細資訊，請從 Events (事件) 清單中，選擇您的專案。
3. 成功完成管道後，即會在 AWS CloudFormation 堆疊中建立資源。在 AWS CloudFormation 主控台中，為您的專案選擇基礎設施堆疊。堆疊名稱會遵循此格式：

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

   在 AWS CloudFormation 主控台的資源清單中，檢視為您的專案建立的資源。在此範例中，新的 Amazon EC2 執行個體會出現在資源區段中。
4. 存取您的生產階段的端點：

   - 對於 Elastic Beanstalk 專案，在 AWS CloudFormation 主控台中開啟新堆疊並展開資源。選擇 Elastic Beanstalk 應用程式。連結會在 Elastic Beanstalk 主控台中開啟。選擇 Environments (環境)。在 URL 中選擇 URL 以在瀏覽器中開啟端點。
   - 對於 Lambda 專案，在 AWS CloudFormation 主控台中開啟新堆疊並展開資源。選擇 API Gateway 資源。連結會在 API Gateway 主控台中開啟。選擇 Stages (階段)。在 Invoke URL (呼叫 URL) 中選擇 URL 以在瀏覽器中開啟端點。
   - 對於 Amazon EC2 專案，請在 AWS CodeStar 主控台的專案資源清單中選擇新的 Amazon EC2 執行個體。連結會在 Amazon EC2 主控台的執行個體頁面上開啟。選擇 Description (描述) 索引標籤，複製 Public DNS (IPv4) (公有 DNS (IPv4)) 中的 URL，並在瀏覽器中開啟該 URL。
5. 驗證已部署您的變更。

在 AWS CodeStar 專案中安全地使用 SSM 參數

許多客戶會將登入資料等秘密存放在 Systems Manager 參數存放區參數中。現在您可以在 AWS CodeStar 專案中安全地使用這些參數。例如，您可能想要在 CodeBuild 的建置規格中使用 SSM 參數，或在工具鏈堆疊 (template.yml) 中定義應用程式資源時。

若要在 AWS CodeStar 專案中使用 SSM 參數，您必須使用 AWS CodeStar 專案 ARN 手動標記參數。您還必須為 AWS CodeStar 工具鏈工作者角色提供適當的許可，以存取您標記的參數。

開始之前

- 建立新的 或識別包含您要存取之資訊的現有 Systems Manager 參數。
- 識別您要使用的 AWS CodeStar 專案，或建立新的專案。
- 記下 CodeStar 專案 ARN。看起來如下：`arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*。

使用 AWS CodeStar 專案 ARN 標記參數

請參閱標記 Systems Manager 參數以取得逐步指示。

1. 在 Key (金鑰) 中，輸入 `awscodestar:projectArn`。
2. 在 Value (值) 中，輸入來自 CodeStar 的專案 ARN：`arn:aws:codestar:`*`region-id`*`:`*`account-id`*`:project/`*`project-id`*。
3. 選擇 Save (儲存)。

現在您可以在 template.yml 檔案中參考 SSM 參數。如果您想要將其搭配工具鏈工作者角色使用，您需要授予額外的許可。

授予在您的 AWS CodeStar Project Toolchain 中使用標記參數的許可

> **Note**
>
> 這些步驟僅適用於 2018 年 12 月 6 日 PDT 之後建立的專案。

1. 開啟您要使用之專案的 AWS CodeStar 專案儀表板。

2. 按一下 Project (專案) 檢視已建立的資源清單，以及尋找工具鏈工作者角色。這是 IAM 資源，具有以下的名稱格式：`role/CodeStarWorker-`*`project-id`*`-ToolChain`。
3. 按一下 ARN，以在 IAM 主控台中開啟。
4. 若必要，找到 ToolChainWorkerPolicy 並將其展開。
5. 按一下 Edit Policy (編輯政策)。
6. 在 `Action:` 下方新增以下行：

   ```
   ssm:GetParameter*
   ```

7. 按一下 Review policy (檢閱政策)，然後按一下 Save changes (儲存變更)。

對於在 2018 年 12 月 6 日之前建立的專案，您需要為每個服務將下列許可新增至工作者角色。

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

轉移 AWS Lambda 專案的流量

AWS CodeDeploy 支援無伺服器專案中 AWS CodeStar 函數的 AWS Lambda 函數版本部署。 AWS Lambda 部署會將傳入流量從現有的 Lambda 函數轉移到更新的 Lambda 函數版本。您可能想要測試更新的 Lambda 函數，做法是部署不同的版本，然後將部署轉返回第一個版本。

使用本節中的步驟來修改您的 AWS CodeStar 專案範本，並更新您的 CodeStarWorker 角色 IAM 許可。此任務會在 中 AWS CloudFormation 啟動自動回應，以建立別名 AWS Lambda 函數，然後指示 AWS CodeDeploy 將流量轉移到更新的 環境。

> **Note**
>
> 只有當您在 2018 年 12 月 12 日之前建立 AWS CodeStar 專案時，才完成這些步驟。

AWS CodeDeploy 有三個部署選項，可讓您將流量轉移到應用程式中的 AWS Lambda 函數版本：

- Canary：流量以兩個增量轉移。您可從預先定義的 canary 選項中選擇，這會指定流量轉移至您於第一次增量時更新的 Lambda 函式版本之百分比，以及在剩餘流量於第二次增量前轉移的間隔 (以分鐘計)。
- Linear (線性)：流量以每個增量之間的相等分鐘數以同等增量轉移。您可從預先指定的線性選項中指定每次增量的流量轉移百分比例，以及在每個增量之間的分鐘數。流量以每個增量之間的相等分鐘數以同等增量轉移。您可從預先指定的線性選項中指定每次增量的流量轉移百分比例，以及在每個增量之間的分鐘數。
- All-at-once (一次全部)：所有流量將從原始 Lambda 函式一次轉移至更新的 Lambda 函式版本。

部署偏好類型
Canary10Percent30Minutes
Canary10Percent5Minutes
Canary10Percent10Minutes
Canary10Percent15Minutes
Linear10PercentEvery10Minutes
Linear10PercentEvery1Minute
Linear10PercentEvery2Minutes
Linear10PercentEvery3Minutes
AllAtOnce

如需在 AWS Lambda 運算平台上 AWS CodeDeploy 部署的詳細資訊，請參閱 AWS Lambda 運算平台上的部署。

如需 SAM AWS 的詳細資訊，請參閱 GitHub 上的AWS 無伺服器應用程式模型 (AWS SAM)。

先決條件：

當您建立無伺服器專案，請選取任何範本與 Lambda 運算平台。您必須以系統管理員身分登入，才能執行步驟 4-6。

步驟 1：修改 SAM 範本以新增 AWS Lambda 版本部署參數

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 建立專案或選擇使用 `template.yml` 檔案的現有專案，然後開啟程式碼頁面。在儲存庫的最高層，留意要修改之名為 `template.yml` 的 SAM 範本位置。
3. 在 IDE 或本機儲存庫中開啟 `template.yml` 檔案。複製以下文字以新增 `Globals` 部分至檔案。本教學課程的範例文字選擇 `Canary10Percent5Minutes` 選項。

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

此範例顯示新增 `Globals` 部分之後的已修改範本：

```
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:
```

如需詳細資訊，請參閱 SAM 範本的全域區段參考指南。

步驟 2：編輯 AWS CloudFormation 角色以新增許可

1. 登入 AWS Management Console ，並在 https：//https://console.aws.amazon.com/codestar/ 開啟 AWS CodeStar 主控台。

> **Note**
>
> 您必須 AWS Management Console 使用與您在 中建立或識別的 IAM 使用者相關聯的登入資料來登入 設定 AWS CodeStar。此使用者必須**AWSCodeStarFullAccess**連接名為 的 AWS 受管政策。

2. 選擇您現有的無伺服器專案，然後開啟專案資源頁面。
3. 在資源下，選擇為 CodeStarWorker/AWS CloudFormation 角色建立的 IAM 角色。在 IAM 主控台開啟該角色。
4. 在 Permissions (許可) 標籤上，在 內嵌政策 的服務角色政策列中，選擇 編輯政策。選擇 JSON 標籤來編輯 JSON 格式的政策。

> **Note**
>
> 您的服務角色名為 `CodeStarWorkerCloudFormationRolePolicy`。

5. 在 JSON 欄位中，在 `Statement` 元素中新增下列政策聲明。將 *`region`* 與 *`id`* 預留位置取代為您的區域與帳戶 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. 選擇檢閱政策，確保政策沒有任何錯誤。當政策沒有任何錯誤時，選擇儲存變更。

步驟 3：遞交並推送您的範本變更以開始 AWS Lambda 版本轉移

1. 遞交和推送在步驟 1 儲存的 `template.yml` 檔案中的變更。

   > **Note**
   >
   > 此會啟動您的管道。如果您在更新 IAM 許可之前遞交變更，則您的管道會啟動，且 AWS CloudFormation 堆疊更新會遇到會復原堆疊更新的錯誤。如果發生這種情況，請在修正權限後重新啟動您的管道。

2. AWS CloudFormation 堆疊更新會在專案的管道啟動部署階段時開始。若要查看部署開始時的堆疊更新通知，請在 AWS CodeStar 儀表板上選取管道中的 AWS CloudFormation 階段。

   在堆疊更新期間， AWS CloudFormation 會自動更新專案資源，如下所示：

   - AWS CloudFormation 透過建立別名 Lambda 函數、事件掛鉤和資源來處理`template.yml`檔案。
   - AWS CloudFormation 會呼叫 Lambda 來建立新的 函數版本。
   - AWS CloudFormation 會建立 AppSpec 檔案和呼叫 AWS CodeDeploy 來轉移流量。

   如需在 SAM 中發佈別名 Lambda 函數的詳細資訊，請參閱AWS 無伺服器應用程式模型 (SAM) 範本參考。如需 AWS CodeDeploy AppSpec 檔案中事件掛鉤和資源的詳細資訊，請參閱適用於 AWS Lambda 部署的 AppSpec 'resources' 區段（僅限 Lambda 部署）和 AppSpec 'hooks' 區段 AWS。

3. 成功完成您的管道之後， AWS CloudFormation 堆疊中會建立資源。在專案頁面上的專案資源清單中，檢視 AWS CodeDeploy 應用程式、 AWS CodeDeploy 部署群組，以及為您的專案建立 AWS CodeDeploy 的服務角色資源。
4. 若要建立新的版本，可在您的儲存庫中變更 Lambda 函數。新部署根據 SAM 範本中指出的部署類型來啟動及轉移流量。若要查看正轉移到新版本的流量狀態，請在專案頁面的專案資源清單中，選擇到 AWS CodeDeploy 部署的連結。
5. 若要檢視每個修訂的詳細資訊，請在修訂下，選擇 AWS CodeDeploy 部署群組的連結。
6. 在本機工作目錄中，您可以變更 AWS Lambda 函數，並將變更遞交至專案儲存庫。 AWS CloudFormation 支援以相同方式 AWS CodeDeploy 管理下一個修訂。如需重新部署、停止或復原 Lambda 部署的詳細資訊，請參閱 AWS Lambda 運算平台上的部署。

將您的 AWS CodeStar 專案轉換為生產

使用 AWS CodeStar 專案建立應用程式並查看 AWS CodeStar 提供的內容之後，您可能想要將專案轉換為生產用途。其中一種方法是在 AWS CodeStar 外部複寫應用程式 AWS 的資源。您仍然需要儲存庫、建置專案、管道和部署，但不是讓 AWS CodeStar 為您建立它們，而是使用 重新建立它們 AWS CloudFormation。

> **Note**
>
> 使用其中一個 AWS CodeStar 快速啟動來建立或檢視類似的專案會很有幫助，並用它做為您自己專案的範本，以確保您包含所需的資源和政策。

AWS CodeStar 專案是原始程式碼與為部署程式碼而建立之資源的組合。可協助您建置、發佈和部署程式碼的資源集合，稱為工具鏈資源。在專案建立時， AWS CloudFormation 範本會在持續整合/持續部署 (CI/CD) 管道中佈建您的工具鏈資源。

使用主控台來建立專案時，即會為您建立工具鏈範本。當您使用 AWS CLI 建立專案時，您可以建立工具鏈範本來建立工具鏈資源。

完整工具鏈需要以下建議的資源：

1. 包含原始程式碼的 CodeCommit 或 GitHub 儲存庫。
2. 設定為接聽儲存庫變更的 CodePipeline 管道。
   a. 當您使用 AWS CodeBuild 執行單元或整合測試時，建議您將建置階段新增至管道，以建立建置成品。
   b. 建議您將部署階段新增至使用 CodeDeploy 的管道 AWS CloudFormation ，或將建置成品和原始程式碼部署至執行期基礎設施。

   > **Note**
   >
   > 由於 CodePipeline 在管道中至少需要兩個階段，而第一個階段必須是來源階段，請新增建置或部署階段做為第二個階段。

主題

建立 GitHub 儲存庫

您可以透過在工具鏈範本中進行定義來建立 GitHub 儲存庫。請確定您已為包含原始程式碼的 ZIP 檔案建立位置，以便可以將該程式碼上傳至儲存庫。此外，您必須已在 GitHub 中建立個人存取字符，以便 AWS 可以代表您連線到 GitHub。除了 GitHub 的個人存取字符之外，您還必須擁有傳入 `Code` 物件的 `s3.GetObject` 許可。

若要指定公有 GitHub 儲存庫，請在 AWS CloudFormation中將如下的程式碼新增至工具鏈範本中。

```
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
```

此程式碼指定以下資訊：

- 您要包含的程式碼位置，必須是 Amazon S3 儲存貯體。
- 是否要啟用 GitHub 儲存庫上的問題。
- 無論 GitHub 儲存庫是否是私有的。
- 您建立的 GitHub 個人存取字符。
- 您正在建立的儲存庫描述、名稱和擁有者。

如需要指定哪些資訊的完整詳細資訊，請參閱AWS CloudFormation 《 使用者指南》中的 AWS::CodeStar::GitHubRepository。

在 中使用專案標籤 AWS CodeStar

您可以 AWS CodeStar中將標籤與專案關聯。標籤可協助您管理專案。例如，您可以將含 `Release` 金鑰和 `Beta` 值的標籤加入到您組織正在處理的任何專案 Beta 版。

新增標籤到專案

1. 在 AWS CodeStar 主控台中開啟專案時，在側邊導覽窗格中，選擇設定。
2. 在標籤中，選擇編輯。
3. 在金鑰中，輸入標籤的名稱。在值中輸入標籤的值。
4. 選用：選擇新增標籤以新增更多標籤。
5. 新增標籤完成後，請選擇儲存。

從專案移除標籤

1. 在 AWS CodeStar 主控台中開啟專案時，在側邊導覽窗格中，選擇設定。
2. 在標籤中，選擇編輯。
3. 在標籤中，尋找您要移除的標籤，然後選擇移除標籤。
4. 選擇 Save (儲存)。

取得專案的標籤清單

使用 AWS CLI 執行 AWS CodeStar list-tags-for-project命令，指定專案的名稱：

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

若執行成功，標籤清單會出現在輸出中，內容與下列相似：

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

刪除 AWS CodeStar 專案

如果您不再需要專案，可以將它及其資源一併刪除，如此 AWS就不會再產生任何費用。當您刪除專案時，所有的團隊成員都會從該專案移除。他們的專案角色會從其 IAM 使用者中移除，但其在 中的使用者設定檔 AWS CodeStar 不會變更。您可以使用 AWS CodeStar 主控台或 AWS CLI 來刪除專案。刪除專案需要 AWS CodeStar 服務角色 `aws-codestar-service-role`，必須未經修改且可由 擔任 AWS CodeStar。

Important

在 中刪除專案 AWS CodeStar 無法復原。根據預設，您 AWS 帳戶中的所有專案 AWS 資源都會遭到刪除，包括：

- 專案的 CodeCommit 儲存庫，以及存放在該儲存庫中的任何項目。
- 為 AWS CodeStar 專案及其資源設定的專案角色和相關聯的 IAM 政策。
- 為專案建立的任何 Amazon EC2 執行個體。
- 部署應用程式和相關資源，例如：
  - CodeDeploy 應用程式和相關聯的部署群組。
  - AWS Lambda 函數和相關聯的 API Gateway APIs。
  - AWS Elastic Beanstalk 應用程式和相關聯的環境。
- CodePipeline 中專案的連續部署管道。
- 與專案相關聯的 AWS CloudFormation 堆疊。
- 使用 AWS CodeStar 主控台建立的任何 AWS Cloud9 開發環境。在環境中的所有未遞交的程式碼變更都會遺失。

若要刪除所有專案資源以及專案，請選取刪除資源核取方塊。如果您清除此選項，專案會在 中刪除 AWS CodeStar，而啟用這些資源存取權的專案角色會在 IAM 中刪除，但所有其他資源都會保留。您可能會繼續對其中的這些資源收取費用 AWS。如果您決定不再需要這些資源中的一個或多個，您必須手動刪除它們。如需詳細資訊，請參閱專案刪除： AWS CodeStar 專案已刪除，但資源仍然存在。
如果您在刪除專案時決定保留資源，最好的做法是在專案詳細資訊頁面將資源清單複製起來。利用這種方式，您可以記錄所有保留的資源，即使專案已不存在。

主題

在 中刪除 AWS CodeStar 專案 （ 主控台 ）

您可以使用 AWS CodeStar 主控台來刪除專案。

在 中刪除專案 AWS CodeStar

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 在導覽窗格中選擇專案。
3. 選取您要刪除的專案，然後選擇刪除。

   或者，開啟專案，然後從主控台左側的導覽窗格中選擇設定。在專案詳細資訊頁面上，選擇 Delete project (刪除專案)。
4. 在刪除確認頁面中，輸入刪除。如果您想要刪除專案資源，請保持選取刪除資源。選擇 刪除 。

   刪除專案可能需要幾分鐘的時間。刪除後，專案不會再出現在 AWS CodeStar 主控台的專案清單中。

> **Important**
>
> 如果您的專案使用 以外的資源 AWS （ 例如，GitHub 儲存庫或 Atlassian JIRA 中的問題 ），即使您選取核取方塊，也不會刪除這些資源。
> 如果任何 AWS CodeStar 受管政策已手動連接至非 IAM 使用者的角色，則無法刪除您的專案。在專案受管政策是連接至聯合身分使用者角色的情況下，您必須先分離該政策，才能刪除專案。如需詳細資訊，請參閱???。

在 AWS CodeStar (AWS CLI) 中刪除專案

您可以使用 AWS CLI 來刪除專案。

在 中刪除專案 AWS CodeStar

1. 在終端機 (Linux、macOS 或 Unix) 或命令提示字元 (Windows) 執行 delete-project命令，包括專案的名稱。例如，刪除 ID 為 *my-2nd-project* 的專案：

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

此命令會傳回類似以下的輸出：

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

專案不會立即被刪除。

2. 執行 describe-project 命令，包括專案名稱。例如，若要檢查 ID 為 *my-2nd-project* 之專案的狀態：

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

如果專案尚未刪除，此命令會傳回類似以下輸出：

```
{
    "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"
    }
}
```

如果專案已刪除，此命令會傳回類似以下輸出：

```
    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. 執行 list-projects 命令，並確認已刪除的專案不會再出現在與您的 AWS 帳戶相關的專案中。

```
aws codestar list-projects
```

使用 AWS CodeStar 團隊

在您建立開發專案後，將存取權授與其他人，如此您便可以和他們共同作業。在 中 AWS CodeStar，每個專案都有一個專案團隊。使用者可以屬於多個 AWS CodeStar 專案，並在每個專案中具有不同的 AWS CodeStar 角色（因此具有不同的許可）。在 AWS CodeStar 主控台中，使用者會看到與 AWS 您的帳戶相關聯的所有專案，但他們只能檢視和處理他們是團隊成員的那些專案。

團隊成員可以為自己選擇易記的名稱。他們也可以新增電子郵件地址，方便其他團隊成員聯絡。非擁有者的團隊成員，無法變更他們在專案中的 AWS CodeStar 角色。

中的每個專案 AWS CodeStar 都有三個角色：

AWS CodeStar 專案中的角色和許可

角色名稱	檢視專案儀表板和狀態	新增/移除/存取專案資源	新增/移除團隊成員	刪除專案
Owner	x	x	x	x
參與者	x	x		
檢視者	x			

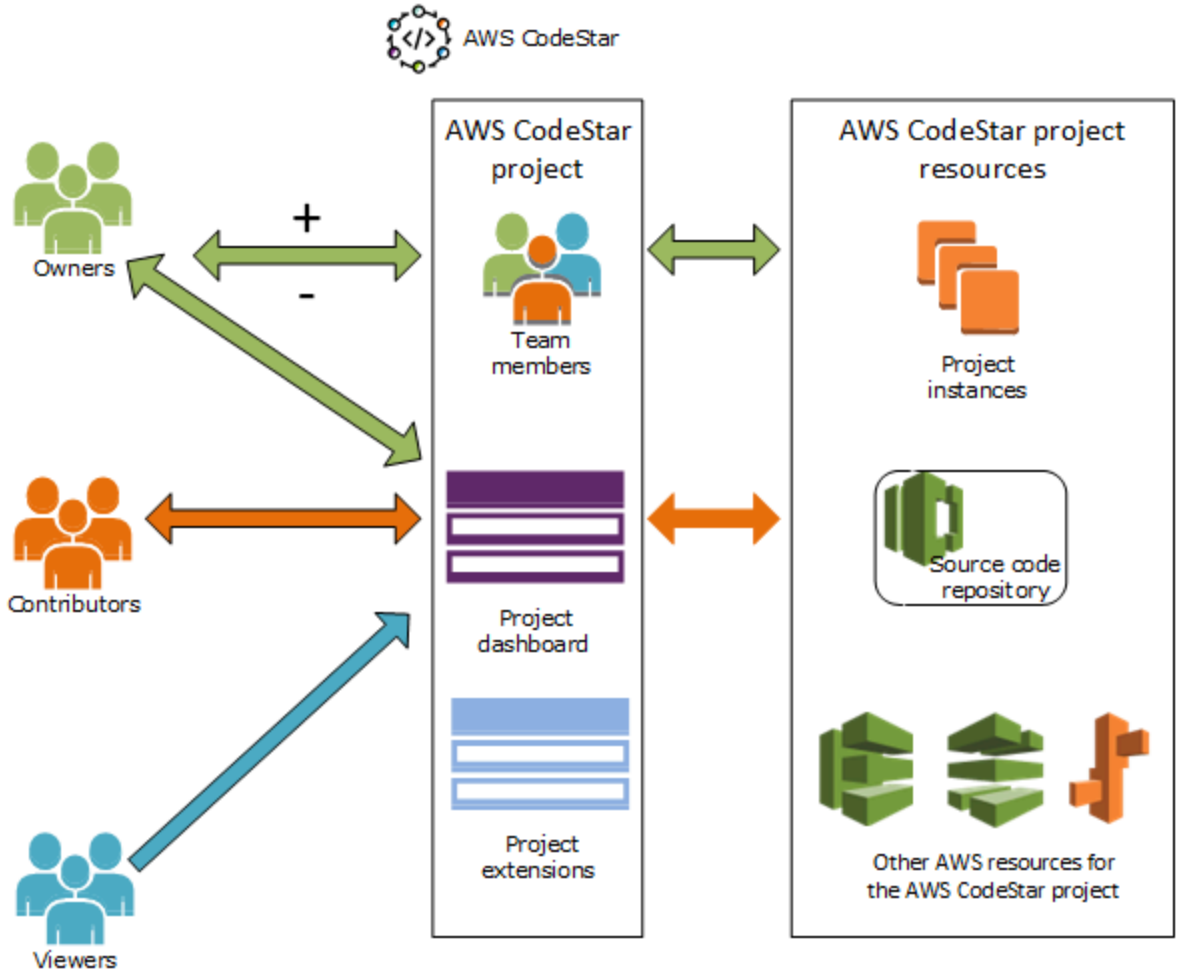
- 擁有者：如果程式碼存放在 CodeCommit 中，可以新增和移除其他團隊成員、將程式碼貢獻至專案儲存庫、授予或拒絕其他團隊成員遠端存取任何執行與專案相關聯的 Linux Amazon EC2 執行個體、設定專案儀表板，以及刪除專案。
- 貢獻者：可以新增和移除儀表板資源，例如 JIRA 圖磚、在程式碼存放在 CodeCommit 中時將程式碼貢獻至專案儲存庫，以及與儀表板完全互動。無法新增或移除團隊成員、授與或拒絕遠端存取資源，或刪除專案。這是您應該為大部分團隊成員選擇的角色。
- 檢視器：可以檢視專案儀表板、如果 存放在 CodeCommit 中的程式碼，以及儀表板圖磚上的專案狀態及其資源。

> **Important**
>
> 如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），則這些資源的存取是由資源提供者控制，而不是由資源提供者控制 AWS CodeStar。如需詳細資訊，請參閱資源提供者的文件。

有權存取 AWS CodeStar 專案的任何人都可以使用 AWS CodeStar 主控台來存取專案外部 AWS 但與專案相關的資源。
AWS CodeStar 不會自動允許專案團隊成員參與專案的任何相關 AWS Cloud9 開發環境。若要允許團隊成員參與共享的環境，請參閱與專案團隊成員共用 AWS Cloud9 環境。

IAM 政策與每個專案角色相關聯。此政策是為您的專案自訂的，可反映它的資源。如需這些政策的詳細資訊，請參閱AWS CodeStar 身分型政策範例。

下圖顯示每個角色和 AWS CodeStar 專案之間的關係。



主題

- 將團隊成員新增至 AWS CodeStar 專案
- 管理 AWS CodeStar 團隊成員的許可
- 從 AWS CodeStar 專案中移除團隊成員

將團隊成員新增至 AWS CodeStar 專案

如果您在 AWS CodeStar 專案中有擁有者角色，或將AWSCodeStarFullAccess政策套用到您的 IAM 使用者，您可以將其他 IAM 使用者新增至專案團隊。這是將 AWS CodeStar 角色（擁有者、參與者或檢視器）套用至使用者的簡單程序。這些角色會根據每個專案和自訂。例如，專案 A 的參與者團隊成員可能有和專案 B 之參與者團隊成員不同的資源的許可權限。一個團隊成員在專案中只能有一個角色。在您新增團隊成員後，該成員便可立即與您的專案在角色所定義的層級互動。

AWS CodeStar 角色和團隊成員的優勢包括：

- 您不需要手動設定團隊成員在 IAM 中的許可。
- 您可以輕鬆地變更團隊成員的專案存取權層級。
- 只有當使用者是團隊成員時，才能在 AWS CodeStar 主控台中存取專案。
- 使用者對專案的存取由角色定義。

如需團隊和 AWS CodeStar 角色的詳細資訊，請參閱 使用 AWS CodeStar 團隊和 使用 AWS CodeStar 使用者設定檔 。

若要將團隊成員新增至專案，您必須擁有專案的 AWS CodeStar 擁有者角色或AWSCodeStarFullAccess政策。

> **Important**
>
> 新增團隊成員不會影響該成員對 外部資源的存取 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題）。這些存取許可是由資源提供者控制，而不是由資源提供者控制 AWS CodeStar。如需詳細資訊，請參閱資源提供者的文件。
>
> 有權存取 AWS CodeStar 專案的任何人都可以使用 AWS CodeStar 主控台來存取該專案外部 AWS 但與該專案相關的資源。
>
> 將團隊成員新增至專案並不會自動允許該成員參與專案的任何相關 AWS Cloud9 開發環境。若要允許團隊成員參與共享的環境，請參閱與專案團隊成員共用 AWS Cloud9 環境。
>
> 授與聯合身分使用者存取專案的權限，牽涉到手動附加 AWS CodeStar 擁有者、參與者或檢視者受管政策到聯合身分使用者所擔任的角色。如需詳細資訊，請參閱聯合身分使用者存取 AWS CodeStar。

主題

- 新增團隊成員 (主控台)

- 新增和檢視團隊成員 (AWS CLI)

新增團隊成員 (主控台)

您可以使用 AWS CodeStar 主控台將團隊成員新增至您的專案。如果 IAM 使用者已存在於您要新增的人員，您可以新增 IAM 使用者。否則，您可以在將 IAM 使用者新增至專案時，為該使用者建立 IAM 使用者。

將團隊成員新增至 AWS CodeStar 專案 （ 主控台 ）

1. 在 https AWS CodeStar ：//https://console.aws.amazon.com/codestar/。
2. 從導覽窗格中選擇專案，然後選擇您的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。
4. 在 Team members (團隊成員) 頁面上，選擇 Add team member (新增團隊成員)。
5. 在 Choose user (選擇使用者) 中，執行下列其中一項操作：

   - 如果您要新增的人員已有 IAM 使用者，請從清單中選擇 IAM 使用者。

     > **Note**
     >
     > 已新增至另一個 AWS CodeStar 專案的使用者會出現在現有 AWS CodeStar 使用者清單中。

     在專案角色中，為此使用者選擇 AWS CodeStar 角色 （ 擁有者、貢獻者或檢視器 ）。這屬於 AWS CodeStar 專案層級角色，唯有專案的擁有者能進行變更。套用至 IAM 使用者時，該角色會提供存取 AWS CodeStar 專案資源所需的所有許可。它會套用為存放在 IAM 中的 CodeCommit 中的程式碼建立和管理 Git 登入資料，或為 IAM 中的使用者上傳 Amazon EC2 SSH 金鑰所需的政策。

     > **Important**
     >
     > 除非您以該使用者身分登入主控台，否則您無法提供或變更 IAM 使用者的顯示名稱或電子郵件資訊。如需詳細資訊，請參閱管理 AWS CodeStar 使用者設定檔的顯示資訊 。

     選擇新增團隊成員。

- 如果您要新增至專案的人員沒有 IAM 使用者，請選擇建立新的 IAM 使用者。系統會將您重新導向至 IAM 主控台，您可以在其中建立新的 IAM 使用者，如需詳細資訊，請參閱《IAM 使用者指南》中的建立 IAM 使用者。建立 IAM 使用者後，返回 AWS CodeStar 主控台，重新整理使用者清單，然後從下拉式清單中選擇您建立的 IAM 使用者。輸入您要套用至此新使用者的 AWS CodeStar顯示名稱、電子郵件地址和專案角色，然後選擇新增團隊成員。

> **Note**
>
> 為了方便管理，您應該將專案的 Owner (擁有者) 角色指派給至少一個使用者。

6. 傳送下列資訊給新的團隊成員：

   - AWS CodeStar 專案的連線資訊。
   - 如果原始程式碼存放在 CodeCommit 中，則使用 Git 登入資料設定從本機電腦存取 CodeCommit 儲存庫的指示。 CodeCommit
   - 有關使用者如何管理其顯示名稱、電子郵件地址和公有 Amazon EC2 SSH 金鑰的資訊，如 中所述使用 AWS CodeStar 使用者設定檔 。
   - 一次性密碼和連線資訊，如果使用者是初次使用 AWS ，而且您為該使用者建立了 IAM 使用者。此密碼會在使用者首次登入後過期，因此使用者必須選擇新的密碼。

新增和檢視團隊成員 (AWS CLI)

您可以使用 AWS CLI 將團隊成員新增至您的專案團隊。您也可以檢視所有專案團隊成員的相關資訊。

新增團隊成員

1. 開啟終端機或命令視窗。
2. 使用 `--project-id`, `-user-arn` 和 `--project-role` 參數執行 associate-team-member 命令。您也可以指定使用者是否擁有遠端存取專案執行個體的權限，包括 `--remote-access-allowed` 或 `--no-remote-access-allowed` 參數。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
```

此命令不會傳回輸出。

檢視所有團隊成員 (AWS CLI)

1. 開啟終端機或命令視窗。
2. 使用 `--project-id` 參數執行 list-team-members 命令。例如：

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

此命令會傳回類似以下的輸出：

```
{
    "teamMembers":[

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

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::111111111
Jane_Doe"},

 {"projectRole":"Contributor","remoteAccessAllowed":true,"userArn":"arn:aws:iam::111111111
John_Doe"},

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

管理 AWS CodeStar 團隊成員的許可

您可以透過變更團隊成員 AWS CodeStar 的角色來變更其許可。每個團隊成員只能指派給 AWS CodeStar 專案中的一個角色，但許多使用者可以指派給相同的角色。您可以使用 AWS CodeStar 主控台或 AWS CLI 來管理許可。

> **Important**
>
> 若要變更團隊成員的角色，您必須擁有該專案的 AWS CodeStar 擁有者角色或套用`AWSCodeStarFullAccess`政策。
> 變更團隊成員的許可不會影響該團隊成員存取 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題）以外的任何資源。這些存取權限是由資源提供者所控制，不是 AWS CodeStar。如需詳細資訊，請參閱資源提供者的文件。

有權存取 AWS CodeStar 專案的任何人都可以使用 AWS CodeStar 主控台來存取 外部 AWS 但與該專案相關的資源。
變更專案的團隊成員角色不會自動允許或阻止該成員參與專案的任何 AWS Cloud9 開發環境。若要允許或預防團隊成員參與共享的環境，請參閱與專案團隊成員共用 AWS Cloud9 環境。

您也可以授予許可，讓使用者從遠端存取與專案相關聯的任何 Amazon EC2 Linux 執行個體。授予此許可後，使用者必須上傳與其 AWS CodeStar 使用者設定檔相關聯的 SSH 公有金鑰，才能跨所有團隊專案。要成功連接到 Linux 執行個體，使用者必須設定 SSH，本機電腦上必須有私有金鑰。

主題

管理團隊許可 (主控台)

您可以使用 AWS CodeStar 主控台來管理團隊成員的角色。您也可以管理團隊成員是否可以遠端存取與您的專案相關聯的 Amazon EC2 執行個體。

變更團隊成員的角色

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 從導覽窗格中選擇專案，然後選擇您的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。
4. 在團隊成員頁面上，選擇團隊成員，然後選擇編輯。
5. 在專案角色中，選擇您要授與此使用者 AWS CodeStar 的角色（擁有者、參與者或檢視器）。

   如需 AWS CodeStar 角色及其許可的詳細資訊，請參閱 使用 AWS CodeStar 團隊。

   選擇編輯團隊成員。

授予團隊成員對 Amazon EC2 執行個體的遠端存取許可

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 從導覽窗格中選擇專案，然後選擇您的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。

4. 在團隊成員頁面上，選擇團隊成員，然後選擇編輯。
5. 選取允許 SSH 存取專案執行個體，然後選擇編輯團隊成員。
6. （選用）通知團隊成員，如果他們尚未上傳 AWS CodeStar SSH 公有金鑰給使用者，他們應該上傳。如需詳細資訊，請參閱將公有金鑰新增至您的 AWS CodeStar 使用者設定檔 。

管理團隊許可 (AWS CLI)

您可以使用 AWS CLI 來管理指派給團隊成員的專案角色。您可以使用相同的 AWS CLI 命令來管理該團隊成員是否可以遠端存取與您的專案相關聯的 Amazon EC2 執行個體。

管理團隊成員的許可

1. 開啟終端機或命令視窗。
2. 使用 `--project-id`, `-user-arn` 和 `--project-role` 參數執行 update-team-member 命令。您也可以指定使用者是否擁有遠端存取專案執行個體的權限，包括 `--remote-access-allowed` 或 `--no-remote-access-allowed` 參數。例如，若要更新名為 John_Doe 之 IAM 使用者的專案角色，並將他的許可變更為無法存取專案 Amazon EC2 執行個體的檢視器：

```
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
```

此命令會傳回類似以下的輸出：

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

從 AWS CodeStar 專案中移除團隊成員

從 AWS CodeStar 專案中移除使用者後，使用者仍會出現在專案儲存庫的遞交歷史記錄中，但不再能夠存取 CodeCommit 儲存庫或任何其他專案資源，例如專案管道。（此規則的例外是 IAM 使用者，其擁有授予這些資源存取權的其他政策。）使用者無法存取專案儀表板，且專案不再出現在使用者在 AWS CodeStar 儀表板上看到的專案清單中。您可以使用 AWS CodeStar 主控台或 AWS CLI 將團隊成員從專案團隊中移除。

> **Important**
>
> 雖然從專案中移除團隊成員會拒絕遠端存取專案 Amazon EC2 執行個體，但不會關閉任何使用者的作用中 SSH 工作階段。
> 移除團隊成員不會影響該團隊成員存取 之外的任何資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題）。這些存取許可是由資源提供者控制，而不是由資源提供者控制 AWS CodeStar。如需詳細資訊，請參閱資源提供者的文件。
> 從專案中移除團隊成員不會自動刪除該團隊成員的相關 AWS Cloud9 開發環境，也不會阻止該成員參與他們獲邀請的任何相關 AWS Cloud9 開發環境。若要刪除開發環境的詳細資訊，請參閱從專案刪除 AWS Cloud9 環境。若要預防團隊成員參與共享的環境，請參閱與專案團隊成員共用 AWS Cloud9 環境。

若要從專案中移除團隊成員，您必須擁有該專案的 AWS CodeStar 擁有者角色，或將`AWSCodeStarFullAccess`政策套用至您的帳戶。

主題

移除團隊成員 (主控台)

您可以使用 AWS CodeStar 主控台從專案團隊中移除團隊成員。

從專案移除團隊成員

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 從導覽窗格中選擇專案，然後選擇您的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。
4. 在團隊成員頁面上，選擇團隊成員，然後選擇移除。

移除團隊成員 (AWS CLI)

您可以使用 AWS CLI 從您的專案團隊中移除團隊成員。

移除團隊成員

1. 開啟終端機或命令視窗。
2. 使用 `--project-id` 和 `-user-arn` 執行 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
```

此命令會傳回類似以下的輸出：

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

使用 AWS CodeStar 使用者設定檔

您的 AWS CodeStar 使用者設定檔與您的 IAM 使用者相關聯。此設定檔包含顯示名稱和電子郵件地址，用於您所屬的所有 AWS CodeStar 專案。您可以上傳 SSH 公有金鑰以與您的設定檔相關聯。此公有金鑰是您連線到與您所屬 AWS CodeStar 專案相關聯的 Amazon EC2 執行個體時所使用的 SSH 公有私有金鑰對的一部分。

> **Note**
>
> 這些主題中的資訊僅涵蓋您的 AWS CodeStar 使用者設定檔。如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），這些資源提供者可能會使用自己的使用者設定檔，這些設定檔可能會有不同的設定。如需詳細資訊，請參閱資源提供者的文件。

主題

管理 AWS CodeStar 使用者設定檔的顯示資訊

您可以使用 AWS CodeStar 主控台或 AWS CLI 來變更使用者設定檔中的顯示名稱和電子郵件地址。使用者描述檔非專案特有的。它與您的 IAM 使用者相關聯，並會套用至您在 區域中所屬的 AWS CodeStar 專案 AWS 。如果您屬於多個 區域中的專案 AWS ，則有單獨的使用者設定檔。

您只能在 AWS CodeStar 主控台中管理自己的使用者設定檔。如果您有 `AWSCodeStarFullAccess`政策，您可以使用 AWS CLI 來檢視和管理其他設定檔。

> **Note**
>
> 本主題中的資訊僅涵蓋您的 AWS CodeStar 使用者設定檔。如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），這些資源提供者可能會使用自己的使用者設定檔，這些設定檔可能會有不同的設定。如需詳細資訊，請參閱資源提供者的文件。

主題

管理您的使用者描述檔 (主控台)

您可以在 AWS CodeStar 主控台中，透過導覽至您身為團隊成員的任何專案，並變更您的設定檔資訊，來管理您的使用者設定檔。由於使用者描述檔是使用者特定的，而不是專案特定的，因此您的使用者描述檔變更會顯示在您身為團隊成員的 AWS 區域中的每個專案中。

> **Important**
>
> 若要使用 主控台變更使用者的顯示資訊，您必須以該 IAM 使用者的身分登入。其他使用者，即使是具有專案 AWS CodeStar 擁有者角色或套用`AWSCodeStarFullAccess`政策的使用者，也無法變更您的顯示資訊。

變更 AWS 區域中所有專案中的顯示資訊

1. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。
2. 從導覽窗格中選擇專案，然後選擇您身為團隊成員的專案。
3. 在專案的側邊導覽窗格中，選擇團隊。
4. 在團隊成員頁面上，選擇 IAM 使用者，然後選擇編輯。
5. 編輯顯示名稱、電子郵件地址或兩者，然後選擇編輯團隊成員。

> **Note**
>
> 顯示名稱和電子郵件地址是必要資料。如需詳細資訊，請參閱中的限制 AWS CodeStar。

管理使用者描述檔 (AWS CLI)

您可以使用 AWS CLI 在 中建立和管理使用者設定檔 AWS CodeStar。您也可以使用 AWS CLI 來檢視您的使用者設定檔資訊，以及檢視 區域中為 AWS 您的帳戶設定的所有使用者設定檔 AWS 。

請確定您的 AWS 設定檔已針對您要建立、管理或檢視使用者設定檔的區域設定。

建立使用者描述檔

1. 開啟終端機或命令視窗。
2. 使用 `user-arn`, `display-name` 和 `email-address` 參數執行 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"
```

此命令會傳回類似以下的輸出：

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

檢視您的顯示資訊

1. 開啟終端機或命令視窗。
2. 使用 `user-arn` 參數執行 describe-user-profile 命令。例如：

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

此命令會傳回類似以下的輸出：

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

變更您的顯示資訊

1. 開啟終端機或命令視窗。
2. 使用 user-arn 參數及您要變更的設定檔參數執行 update-user-profile 命令，例如 display-name 或 email-address。例如，如果具有該顯示名稱 Jane Doe 的使用者想要將她的顯示名稱變更為 Jane Mary Doe：

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

此命令會傳回類似以下的輸出：

```
{
 "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 AWS 帳戶中 AWS 區域中的所有使用者設定檔

1. 開啟終端機或命令視窗。
2. 執行 aws codestar list-user-profiles 命令。例如：

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

此命令會傳回類似以下的輸出：

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

將公有金鑰新增至您的 AWS CodeStar 使用者設定檔

您可以上傳公有 SSH 金鑰，以當做您建立和管理的部分公有-私有金鑰對。您可以使用此 SSH 公有私有金鑰對來存取執行 Linux 的 Amazon EC2 執行個體。如果專案擁有者授予您遠端存取權限，您只能存取這些與專案相關聯的執行個體。您可以使用 AWS CodeStar 主控台或 AWS CLI 來管理您的公有金鑰。

> **Important**
>
> AWS CodeStar 專案擁有者可以授予專案擁有者、參與者和檢視器對專案 Amazon EC2 執行個體的 SSH 存取權，但只有個別（擁有者、參與者或檢視器）可以設定 SSH 金鑰。若要這樣做，使用者必須登入為個別擁有者、參與者或檢視者。
>
> AWS CodeStar 不會管理 AWS Cloud9 環境的 SSH 金鑰。

主題

管理您的公有金鑰 (主控台)

雖然您無法在 主控台中產生公有/私有金鑰對，但您可以在本機建立一個金鑰對，然後透過 AWS CodeStar 主控台新增或管理它做為使用者設定檔的一部分。

管理您的 SSH 公有金鑰

1. 從終端機或 Bash 模擬器窗口，執行 ssh-keygen 命令以在本機電腦產生 SSH 公有-私有金鑰對儲存。您可以產生 Amazon EC2 允許的任何格式的金鑰。如需可接受格式的資訊，請參閱將您自己的公有金鑰匯入 Amazon EC2。最理想的狀況是產生 OpenSSH 格式的 SSH-2 RSA，並且包含 2048 位元。公有金鑰會存放在副檔名為 .pub 的檔案中。
2. 開啟 AWS CodeStar 主控台，網址為 https://console.aws.amazon.com/codestar/：//。

   選擇您為其團隊成員的專案。
3. 在導覽窗格中，選擇團隊。
4. 在團隊成員頁面上，尋找您的 IAM 使用者名稱，然後選擇編輯。
5. 在編輯團隊成員頁面的遠端存取下，啟用允許 SSH 存取專案執行個體。
6. 在 SSH 公有金鑰方塊中，貼上公有金鑰，然後選擇編輯團隊成員。

> **Note**
>
> 您可以變更您的公有金鑰，做法是刪除此欄位中的舊金鑰，並貼上新的金鑰。您可以刪除此欄位的內容，然後選擇編輯團隊成員，以刪除公有金鑰。

若變更或刪除公有金鑰，就會變更您的使用者描述檔。它不是根據每個專案進行變更。由於您的金鑰與您的設定檔相關聯，它會在所有您已被授與遠端存取權的專案中變更 (或刪除)。

刪除您的公有金鑰會移除您在獲得遠端存取權限的所有專案中對執行 Linux 的 Amazon EC2 執行個體的存取權。不過，使用該金鑰不會關閉任何 SSH 工作階段。請確實關閉任何開啟的工作階段。

管理您的公有金鑰 (AWS CLI)

您可以使用 AWS CLI 來管理 SSH 公有金鑰，做為使用者設定檔的一部分。

管理您的公有金鑰

1. 從終端機或 Bash 模擬器窗口，執行 ssh-keygen 命令以在本機電腦產生 SSH 公有-私有金鑰對儲存。您可以產生 Amazon EC2 允許的任何格式的金鑰。如需可接受格式的資訊，請參閱將您自己的公有金鑰匯入 Amazon EC2。最理想的狀況是產生 OpenSSH 格式的 SSH-2 RSA，並且包含 2048 位元。公有金鑰會存放在副檔名為 .pub 的檔案中。
2. 若要在 AWS CodeStar 使用者設定檔中新增或變更 SSH 公有金鑰，請使用 `--ssh-public-key` 參數執行 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
```

此命令會傳回類似以下的輸出：

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

使用您的私有金鑰連線至 Amazon EC2 執行個體

請確定您已建立 Amazon EC2 金鑰對。將公有金鑰新增至 中的使用者設定檔 AWS CodeStar。若要建立金鑰對，請參閱步驟 4：建立專案的 AWS CodeStar Amazon EC2 金鑰對。若要新增您的公開金鑰至您的使用者描述檔，請參閱此主題稍早的指示。

使用私有金鑰連線至 Amazon EC2 Linux 執行個體

1. 在 AWS CodeStar 主控台中開啟您的專案時，在導覽窗格中，選擇專案。
2. 在專案資源中，選擇類型為 Amazon EC2 且名稱開頭為執行個體之資料列中的 ARN 連結。
3. 在 Amazon EC2 主控台中，選擇連線。
4. 請遵循連結到您的執行個體對話方塊中的指示。

   對於使用者名稱，請使用 `ec2-user`。如果使用錯誤的使用者名稱，您無法連接到執行個體。

如需詳細資訊，請參閱《Amazon EC2 使用者指南》中的下列資源。

- 使用 SSH 連接至您的 Linux 執行個體
- 使用 PuTTY 從 Windows 連接至您的 Linux 執行個體
- 使用 MindTerm 連線至您的 Linux 執行個體

中的安全性 AWS CodeStar

的雲端安全 AWS 是最高優先順序。身為 AWS 客戶，您可以受益於資料中心和網路架構，這些架構專為滿足最安全敏感組織的需求而建置。

安全是 AWS 與您之間的共同責任。共同責任模型將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 – AWS 負責保護在 Cloud AWS 中執行 AWS 服務的基礎設施。 AWS 也為您提供可安全使用的服務。在AWS 合規計畫 中，第三方稽核人員會定期測試和驗證我們安全的有效性。若要了解適用於 的合規計劃 AWS CodeStar，請參閱合規計劃的 AWS Services in Scope by Compliance Program。
- 雲端的安全性 – 您的責任取決於您使用 AWS 的服務。您也必須對其他因素負責，包括資料的機密性、您公司的要求和適用法律和法規。

本文件可協助您了解如何在使用 時套用共同責任模型 AWS CodeStar。下列主題說明如何設定 AWS CodeStar 以符合您的安全與合規目標。您也會了解如何使用其他 AWS 服務來協助您監控和保護 AWS CodeStar 資源。

當您在 中建立自訂政策並使用許可界限時 AWS CodeStar，請僅授予執行任務所需的許可，並將許可縮減至目標資源，以確保最低權限的存取。為了防止其他專案的成員存取專案中的資源，請授予組織成員每個 AWS CodeStar 專案的個別許可。最佳實務是為每個成員建立專案帳戶，然後將角色型存取權指派給該帳戶。

例如，您可以使用 AWS Control Tower with AWS Organizations 等服務，為 DevOps 群組下的每個開發人員角色佈建帳戶。然後，您可以將許可指派給這些帳戶。整體許可適用於帳戶，但使用者對專案外部資源的存取有限。

如需使用多帳戶策略管理最低權限 AWS 資源存取的詳細資訊，請參閱AWS 《Control Tower 使用者指南》中的登陸區域的 AWS 多帳戶策略。

主題

- 中的資料保護 AWS CodeStar
- AWS CodeStar 的 Identity and Access Management
- 使用 記錄 AWS CodeStar API 呼叫 AWS CloudTrail
- 的合規驗證 AWS CodeStar

- 中的彈性 AWS CodeStar
- 中的基礎設施安全 AWS CodeStar

中的資料保護 AWS CodeStar

AWS 共同責任模型適用於 AWS CodeStar 中的資料保護。如此模型所述， AWS 負責保護執行所有 的全域基礎設施 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務 的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱資料隱私權常見問答集。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 AWS 共同的責任模型和 GDPR 部落格文章。

基於資料保護目的，我們建議您保護 AWS 帳戶 登入資料，並使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用 設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案，以及其中的所有預設安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在 AWS 透過命令列界面或 API 存取 時需要 FIPS 140-3 驗證的密碼編譯模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱聯邦資訊處理標準 (FIPS) 140-3。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用 CodeStar 或其他 AWS 服務 使用主控台、API AWS CLI或 AWS SDKs時。您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

中的資料加密 AWS CodeStar

根據預設， 會 AWS CodeStar 加密其存放的專案相關資訊。專案 ID 以外的所有項目是在靜態時加密，例如專案名稱、描述和使用者電子郵件。避免將個人資訊放入您的專案 IDs。 AWS CodeStar 也會依預設加密傳輸中的資訊。客戶不需採取任何動作，即可進行靜態加密或傳輸中的加密。

AWS CodeStar 的 Identity and Access Management

AWS Identity and Access Management (IAM) 是 AWS 服務 ，可協助管理員安全地控制對 AWS 資源的存取。IAM 管理員可控制誰可以進行身分驗證 （ 登入 ） 和授權 （ 具有許可 ） 來使用 AWS CodeStar 資源。IAM 是您可以免費使用 AWS 服務 的 。

主題

目標對象

您使用 AWS Identity and Access Management (IAM) 的方式會有所不同，取決於您在 AWS CodeStar 中執行的工作。

服務使用者 – 如果您使用 AWS CodeStar 服務來執行任務，您的管理員會為您提供所需的登入資料和許可。當您使用更多 AWS CodeStar 功能來執行工作時，您可能需要額外的許可。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 AWS CodeStar 中的功能，請參閱 故障診斷 AWS CodeStar Identity and Access。

服務管理員 – 如果您在公司負責 AWS CodeStar 資源，您可能擁有 AWS CodeStar 的完整存取權。您的任務是判斷服務使用者應存取的 AWS CodeStar 功能和資源。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何搭配 AWS CodeStar 使用 IAM，請參閱 AWS CodeStar 如何與 IAM 搭配使用。

IAM 管理員 – 如果您是 IAM 管理員，建議您了解撰寫政策以管理 AWS CodeStar 存取的詳細資訊。若要檢視您可以在 IAM 中使用的 AWS CodeStar 身分型政策範例，請參閱 AWS CodeStar 身分型政策範例。

使用身分來驗證

身分驗證是您 AWS 使用身分憑證登入 的方式。您必須以 AWS 帳戶根使用者身分、IAM 使用者身分或擔任 IAM 角色來驗證 （ 登入 AWS)。

您可以使用透過身分來源提供的憑證，以聯合身分 AWS 身分身分身分身分登入 。 AWS IAM Identity Center (IAM Identity Center) 使用者、您公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料，都是聯合身分的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使用聯合 AWS 身分存取 時，您會間接擔任角色。

視您的使用者類型而定，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需登入的詳細資訊 AWS，請參閱AWS 登入 《 使用者指南》中的如何登入您的 AWS 帳戶 。

如果您以 AWS 程式設計方式存取 ， AWS 會提供軟體開發套件 (SDK) 和命令列界面 (CLI)，以使用您的 憑證以密碼編譯方式簽署您的請求。如果您不使用 AWS 工具，則必須自行簽署請求。如需使用建議方法自行簽署請求的詳細資訊，請參閱《IAM 使用者指南》中的簽署 AWS API 請求。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如， AWS 建議您使用多重要素驗證 (MFA) 來提高帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的多重要素驗證和 IAM 使用者指南中的在 AWS中使用多重要素驗證 (MFA)。

AWS 帳戶 根使用者

建立 時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取帳戶中的所有 AWS 服務 和資源。此身分稱為 AWS 帳戶 Theroot 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱《IAM 使用者指南》中的需要根使用者憑證的任務。

IAM 使用者和群組

IAM 使用者是 中的身分 AWS 帳戶 ，具有單一人員或應用程式的特定許可。建議您盡可能依賴臨時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 IAM 使用者指南中的為需要長期憑證的使用案例定期輪換存取金鑰。

IAM 群組是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱IAM 使用者指南中的建立 IAM 使用者 (而非角色) 的時機。

IAM 角色

IAM 角色是 中具有特定許可 AWS 帳戶 的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以切換角色，暫時在 中擔任 AWS Management Console IAM 角色。您可以透過呼叫 AWS CLI 或 AWS API 操作或使用自訂 URL 來擔任角色。如需使用角色的方法詳細資訊，請參閱 IAM 使用者指南中的使用 IAM 角色。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 — 如需向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 IAM 使用者指南中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的許可集。
- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權：您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的主體) 存取您帳戶的資源。角色是授予跨帳戶存取權的主要方式。不過，對於某些 AWS 服務，您可以將政策直接連接到資源（而不是使用角色做為代理）。如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。
- 跨服務存取 – 有些 AWS 服務 使用其他 中的功能 AWS 服務。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉送存取工作階段 (FAS) – 當您使用 IAM 使用者或角色在其中執行動作時 AWS，您被視為委託人。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 使用呼叫 的委託人許可 AWS 服務，結合 AWS 服務 請求向下游服務提出請求。只有當服務收到需要與其他 AWS 服務 或 資源互動才能完成的請求時，才會提出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱《轉發存取工作階段》。
  - 服務角色 – 服務角色是服務擔任的 IAM 角色，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需詳細資訊，請參閱 IAM 使用者指南中的建立角色以委派許可給 AWS 服務服務。

- 服務連結角色 – 服務連結角色是一種連結至 的服務角色類型 AWS 服務。服務可以擔任代表您執行動作的角色。服務連結角色會出現在您的 中 AWS 帳戶 ，並由服務擁有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 – 您可以使用 IAM 角色來管理在 EC2 執行個體上執行之應用程式的臨時登入資料，以及提出 AWS CLI 或 AWS API 請求。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體，並將其提供給其所有應用程式，您可以建立連接至執行個體的執行個體描述檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需詳細資訊，請參閱 IAM 使用者指南中的利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式。

如需了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的建立 IAM 角色 (而非使用者) 的時機。

使用政策管理存取權

您可以透過建立政策並將其連接到身分或資源 AWS 來控制 AWS 中的存取。政策是 中的物件，當與身分或資源建立關聯時， AWS 會定義其許可。當委託人 （使用者、根使用者或角色工作階段） 發出請求時， 會 AWS 評估這些政策。政策中的許可決定是否允許或拒絕請求。大多數政策會以 JSON 文件 AWS 的形式存放在 中。如需 JSON 政策文件結構和內容的詳細資訊，請參閱 IAM 使用者指南中的 JSON 政策概觀。

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該政策的使用者可以從 AWS Management Console AWS CLI、 或 AWS API 取得角色資訊。

身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的建立 IAM 政策。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管政策是獨立的政策，您可以連接到 中的多個使用者、群組和角色 AWS 帳戶。受管政策包括 AWS 受

管政策和客戶受管政策。如需了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的在受管政策和內嵌政策間選擇。

資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中指定主體。委託人可以包括帳戶、使用者、角色、聯合身分使用者或 AWS 服務。

資源型政策是位於該服務中的內嵌政策。您無法在資源型政策中使用來自 IAM 的 AWS 受管政策。

存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

Amazon S3 AWS WAF和 Amazon VPC 是支援 ACLs 的服務範例。如需進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的存取控制清單 (ACL) 概觀。

其他政策類型

AWS 支援其他較不常見的政策類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 `Principal` 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可界限的詳細資訊，請參閱 IAM 使用者指南中的 IAM 實體許可界限。
- 服務控制政策 SCPs) – SCPs是 JSON 政策，可指定 in. 中組織或組織單位 (OU) 的最大許可 AWS Organizations。 AWS Organizations 是一種用於分組和集中管理您企業擁有 AWS 帳戶 之多個 的服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 會限制成員帳戶中實體的許可，包括每個實體 AWS 帳戶根使用者。如需 Organizations 和 SCP 的詳細資訊，請參閱《AWS Organizations 使用者指南》中的服務控制政策。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過撰寫程式的方式建立角色或聯合使用者的暫時工作階段時，做為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需詳細資訊，請參閱 IAM 使用者指南中的工作階段政策。

多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。若要了解如何 AWS 在涉及多種政策類型時決定是否允許請求，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS CodeStar 如何與 IAM 搭配使用

使用 IAM 管理 AWS CodeStar 的存取權之前，您應該先了解哪些 IAM 功能可與 AWS CodeStar 搭配使用。若要全面了解 AWS CodeStar 和其他 AWS 服務如何與 IAM 搭配使用，請參閱《IAM 使用者指南》中的與 IAM AWS 搭配使用的 服務。

主題

- AWS CodeStar 身分型政策
- AWS CodeStar 資源型政策
- 根據 AWS CodeStar 標籤的授權
- AWS CodeStar IAM 角色
- IAM 使用者存取 AWS CodeStar
- 聯合身分使用者存取 AWS CodeStar
- 搭配 AWS CodeStar 使用暫時登入資料
- 服務連結角色
- 服務角色

AWS CodeStar 身分型政策

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及允許或拒絕動作的條件。 AWS CodeStar 會代表您建立數個身分型政策， AWS CodeStar 允許 在 AWS CodeStar 專案範圍內建立和管理資源。AWS CodeStar 支援特定動作、資源和條件索引鍵。若要了解您在 JSON 政策中使用的所有元素，請參閱 IAM 使用者指南中的 JSON 政策元素參考。

動作

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 `Action` 元素描述您可以用來允許或拒絕政策中存取的動作。政策動作通常具有與相關聯 AWS API 操作相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

AWS CodeStar 中的政策動作在動作之前使用下列字首：codestar:。例如，若要允許指定的 IAM 使用者編輯 AWS CodeStar 專案的屬性，例如其專案描述，您可以使用下列政策陳述式：

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

政策陳述式必須包含 Action 或 NotAction 元素。AWS CodeStar 會定義自己的一組動作，描述您可以使用此服務執行的任務。

若要在單一陳述式中指定多個動作，請用逗號分隔，如下所示：

```
"Action": [
      "codestar:action1",
      "codestar:action2"
```

您也可以使用萬用字元 (*) 來指定多個動作。例如，若要指定開頭是 List 文字的所有動作，請包含以下動作：

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

若要查看 AWS CodeStar 動作的清單，請參閱《IAM 使用者指南》中的 AWS CodeStar 定義的動作。

資源

管理員可以使用 AWS JSON 政策來指定誰可以存取內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 Amazon Resource Name (ARN) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (*) 來表示陳述式適用於所有資源。

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

AWS CodeStar 專案資源具有下列 ARN：

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

如需 ARNs 格式的詳細資訊，請參閱 Amazon Resource Name (ARNs) AWS 和服務命名空間。

例如，以下指定名為 的 AWS CodeStar 專案*my-first-projec*已註冊至 AWS 區域 111111111111中的 AWS 帳戶us-east-2：

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

以下指定任何以my-proj註冊到 AWS 區域 111111111111中 AWS 帳戶的名稱開頭的 AWS CodeStar 專案us-east-2：

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

有些 AWS CodeStar 動作無法在資源上執行，例如列出專案。在這些情況下，您必須使用萬用字元 (*)。

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

若要查看 AWS CodeStar 資源類型及其 ARNs 的清單，請參閱《IAM 使用者指南》中的 AWS CodeStar 定義的資源。若要了解您可以使用哪些動作指定每個資源的 ARN，請參閱 AWS CodeStar 定義的動作。

條件金鑰

AWS CodeStar 不提供任何服務特定的條件金鑰，但支援使用一些全域條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的AWS 全域條件內容金鑰。

範例

若要檢視 AWS CodeStar 身分型政策的範例，請參閱 AWS CodeStar 身分型政策範例。

AWS CodeStar 資源型政策

AWS CodeStar 不支援以資源為基礎的政策。

根據 AWS CodeStar 標籤的授權

您可以將標籤連接至 AWS CodeStar 專案，或在請求中將標籤傳遞至 AWS CodeStar。如需根據標籤控制存取，請使用 `codestar:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件索引鍵，在政策的條件元素中，提供標籤資訊。如需標記 AWS CodeStar 資源的詳細資訊，請參閱 the section called “使用專案標籤”。

若要檢視以身分為基礎的政策範例，以根據 AWS CodeStar 專案上的標籤限制對 專案的存取，請參閱根據標籤檢視 AWS CodeStar 專案。

AWS CodeStar IAM 角色

IAM 角色是您 AWS 帳戶中具有特定許可的實體。

您可以使用 AWS CodeStar 做為 IAM 使用者、聯合身分使用者、根使用者或擔任的角色。具有適當許可的所有使用者類型都可以管理其 AWS 資源的專案許可，但會自動 AWS CodeStar 管理 IAM 使用者的專案許可。IAM 政策和角色會根據專案角色授予許可和存取權給該使用者。您可以使用 IAM 主控台來建立其他政策，將 AWS CodeStar 和其他許可指派給 IAM 使用者。

例如，您可能想要讓使用者檢視但不能變更 AWS CodeStar 專案。在此情況下，您會將 IAM 使用者新增至具有檢視器角色的 AWS CodeStar 專案。每個 AWS CodeStar 專案都有一組政策，可協助您控制對專案的存取。此外，您可以控制哪些使用者可以存取 AWS CodeStar。

AWS CodeStar IAM 使用者和聯合身分使用者的存取處理方式不同。只有 IAM 使用者可以新增到團隊。若要對 IAM 使用者授與專案許可，您可以將使用者新增至專案團隊，並為使用者指派角色。若要將聯合身分使用者許可授予專案，您可以手動將 AWS CodeStar 專案角色的受管政策連接至聯合身分使用者的角色。

下表總結各種存取類型可用的工具。

許可功能	IAM 使用者	聯合身分使用者	根使用者
Amazon EC2 和 Elastic Beanstalk 專案遠端存取的 SSH 金鑰管理	✓		

許可功能	IAM 使用者	聯合身分使用者	根使用者
AWS CodeCommit SSH 存取	✓		
由 管理的 IAM 使用者許可 AWS CodeStar	✓		
手動管理的專案許可		✓	✓
使用者可新增至專案做為團隊成員	✓		

IAM 使用者存取 AWS CodeStar

當您將 IAM 使用者新增至專案，並為使用者選擇角色時， AWS CodeStar 會自動將適當的政策套用至 IAM 使用者。對於 IAM 使用者，您不需要直接連接或管理 IAM 中的政策或許可。如需將 IAM 使用者新增至 AWS CodeStar 專案的詳細資訊，請參閱 將團隊成員新增至 AWS CodeStar 專案 。如需從 AWS CodeStar 專案中移除 IAM 使用者的資訊，請參閱 從 AWS CodeStar 專案中移除團隊成員 。

將內嵌政策連接至 IAM 使用者

當您將使用者新增至專案時， AWS CodeStar 會自動為符合使用者角色的專案連接受管政策。您不應該手動將專案的 AWS CodeStar 受管政策連接至 IAM 使用者。除了 之外`AWSCodeStarFullAccess`，我們不建議您連接變更 AWS CodeStar 專案中 IAM 使用者許可的政策。如果您決定建立和連接自己的政策，請參閱《IAM 使用者指南》中的新增和移除 IAM 身分許可。

聯合身分使用者存取 AWS CodeStar

您可以利用來自企業使用者目錄 AWS Directory Service、Web 身分提供者或擔任角色的 IAM 使用者，來建立 IAM 使用者或使用根使用者。這些稱為「聯合身分使用者」。

透過 AWS CodeStar 手動將AWS CodeStar 專案層級政策和許可中所述的受管政策連接至使用者的 IAM 角色，授予聯合身分使用者存取專案的權限。在 AWS CodeStar 建立專案資源和 IAM 角色之後，您會連接擁有者、參與者或檢視器政策。

先決條件：

- 您必須已設定身分提供者。例如，您可以設定 SAML 身分提供者，並透過提供者設定 AWS 身分驗證。如需設定身分提供者的詳細資訊，請參閱建立 IAM 身分提供者。如需有關 SAML 聯合身分詳細資訊，請參閱關於以 SAML 2.0 為基礎的聯合身分。
- 透過身分提供者身分提供者請求存取時，您必須已經建立要讓聯合身分使用者擔任的角色。STS 信任政策必須連接至允許聯合身分使用者擔任的角色。如需詳細資訊，請參閱《IAM 使用者指南》中的聯合身分使用者與角色。
- 您必須已建立 AWS CodeStar 專案並知道專案 ID。

如需有關為身分提供者建立角色的詳細資訊，請參閱為第三方身分提供者 (聯合身分) 建立角色。

將 AWSCodeStarFullAccess 受管政策連接至聯合身分使用者的角色

連接 `AWSCodeStarFullAccess` 受管政策以授予聯合身分使用者建立專案的許可。若要執行這些步驟，您必須以根使用者、帳戶中的管理員使用者，或是具有相關聯`AdministratorAccess`受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

> **Note**
>
> 在您建立專案之後，不會自動套用您的專案擁有者權限。使用具有您帳戶的管理許可的角色，連接擁有者受管政策，如將專案的 AWS CodeStar Viewer/Contributor/Owner受管政策連接至聯合身分使用者的角色中所述。

1. 開啟 IAM 主控台。在導覽窗格中，選擇政策。
2. 在搜尋欄位中輸入 `AWSCodeStarFullAccess`。政策名稱隨即顯示，政策類型為 AWS 受管。您可以擴展政策以查看政策陳述式中的許可。
3. 選取政策旁的圓圈，然後在 Policy actions (政策動作) 下方選擇 Attach (連接)。
4. 在 Summary (摘要) 頁面上選擇 Attached entities (連接的實體) 索引標籤。選擇 Attach (連接)。
5. 在 Attach Policy (連接政策) 頁面上，在搜尋欄位中篩選聯合身分使用者的角色。選取角色名稱旁邊的方塊，然後選擇 Attach policy (連接政策)。Attached entities (連接的實體) 索引標籤將會顯示新的連接。

將專案的 AWS CodeStar Viewer/Contributor/Owner受管政策連接至聯合身分使用者的角色

將適當的 擁有者、作者群或檢視者受管政策連接至使用者的角色，藉此授予聯合身分使用者存取您的專案。受管政策可提供適當的許可層級。不同於 IAM 使用者，您必須為聯合身分使用者手動連接及分離受管政策。這相當於將專案許可指派給其中的團隊成員 AWS CodeStar。若要執行這些步驟，您必

須以根使用者、帳戶中的管理員使用者，或是具有相關聯AdministratorAccess受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

先決條件：

- 您必須已經建立您的聯合身分使用者將要擔任的角色或已擁有現有的角色。
- 您必須知道您要授予哪個許可層級。連接至擁有者、作者群及檢視者角色的受管政策，為您的專案提供以角色為基礎的許可。
- 必須已建立您的 AWS CodeStar 專案。在建立專案之前，IAM 中無法使用 受管政策。

1. 開啟 IAM 主控台。在導覽窗格中，選擇政策。
2. 在搜尋欄位中輸入您的專案 ID。此時將會顯示符合您專案的政策名稱，其政策類型為 Customer managed (客戶受管)。您可以擴展政策以查看政策陳述式中的許可。
3. 選擇下列其中一項受管政策。選取政策旁的圓圈，然後在 Policy actions (政策動作) 下方選擇 Attach (連接)。
4. 在 Summary (摘要) 頁面上選擇 Attached entities (連接的實體) 索引標籤。選擇 Attach (連接)。
5. 在 Attach Policy (連接政策) 頁面上，在搜尋欄位中篩選聯合身分使用者的角色。選取角色名稱旁邊的方塊，然後選擇 Attach policy (連接政策)。Attached entities (連接的實體) 索引標籤將會顯示新的連接。

從聯合身分使用者的角色分離 AWS CodeStar 受管政策

刪除 AWS CodeStar 專案之前，您必須手動分離連接到聯合身分使用者角色的任何受管政策。若要執行這些步驟，您必須以根使用者、帳戶中的管理員使用者，或是具有相關聯AdministratorAccess受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

1. 開啟 IAM 主控台。在導覽窗格中，選擇政策。
2. 在搜尋欄位中輸入您的專案 ID。
3. 選取政策旁的圓圈，然後在 Policy actions (政策動作) 下方選擇 Attach (連接)。
4. 在 Summary (摘要) 頁面上選擇 Attached entities (連接的實體) 索引標籤。
5. 在搜尋欄位中篩選聯合身分使用者的角色。請選擇分離。

將 AWS Cloud9 受管政策連接至聯合身分使用者的角色

如果您使用的是 AWS Cloud9 開發環境，請將 AWSCloud9User受管政策連接至使用者的角色，授予聯合身分使用者存取權。不同於 IAM 使用者，您必須為聯合身分使用者手動連接及分

離受管政策。若要執行這些步驟，您必須以根使用者、帳戶中的管理員使用者，或是具有相關聯AdministratorAccess受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

先決條件：

- 您必須已經建立您的聯合身分使用者將要擔任的角色或已擁有現有的角色。
- 您必須知道您要授予哪個許可層級：
  - AWSCloud9User 受管政策允許使用者執行下列動作：
    - 建立自己的 AWS Cloud9 開發環境。
    - 取得其環境的相關資訊。
    - 變更其環境的設定。
  - AWSCloud9Administrator 受管政策允許使用者為自己或其他使用者執行下列動作：
    - 建立環境。
    - 取得環境的相關資訊。
    - 刪除環境。
    - 變更環境的設定。

1. 開啟 IAM 主控台。在導覽窗格中，選擇政策。
2. 在搜尋欄位中輸入政策名稱。隨即顯示受管政策，政策類型為 受AWS 管。您可以擴展政策以查看政策陳述式中的許可。
3. 選擇下列其中一項受管政策。選取政策旁的圓圈，然後在 Policy actions (政策動作) 下方選擇 Attach (連接)。
4. 在 Summary (摘要) 頁面上選擇 Attached entities (連接的實體) 索引標籤。選擇 Attach (連接)。
5. 在 Attach Policy (連接政策) 頁面上，在搜尋欄位中篩選聯合身分使用者的角色。選擇角色名稱旁邊的方塊，然後選擇 Attach policy (連接政策)。Attached entities (連接的實體) 索引標籤將會顯示新的連接。

從聯合身分使用者的角色分離 AWS Cloud9 受管政策

如果您使用的是 AWS Cloud9 開發環境，您可以透過分離授予存取權的政策來移除聯合身分使用者的存取權。若要執行這些步驟，您必須以根使用者、帳戶中的管理員使用者，或是具有相關聯AdministratorAccess受管政策或同等政策的 IAM 使用者或聯合身分使用者身分登入 主控台。

1. 開啟 IAM 主控台。在導覽窗格中，選擇政策。

2. 在搜尋欄位中輸入您的專案名稱。
3. 選取政策旁的圓圈，然後在 Policy actions (政策動作) 下方選擇 Attach (連接)。
4. 在 Summary (摘要) 頁面上選擇 Attached entities (連接的實體) 索引標籤。
5. 在搜尋欄位中篩選聯合身分使用者的角色。請選擇分離。

搭配 AWS CodeStar 使用暫時登入資料

您可以搭配聯合使用暫時憑證、擔任 IAM 角色，或是擔任跨帳戶角色。您可以透過呼叫 AssumeRole 或 GetFederationToken 等 AWS STS API 操作來取得臨時安全登入資料。

AWS CodeStar 支援使用臨時登入資料，但 AWS CodeStar 團隊成員功能不適用於聯合身分存取。AWS CodeStar 團隊成員功能僅支援將 IAM 使用者新增為團隊成員。

服務連結角色

服務連結角色可讓 AWS 服務存取其他服務中的資源，以代表您完成 動作。服務連結角色會顯示在您的 IAM 帳戶中，並由該服務所擁有。 管理員可以檢視，但不能編輯服務連結角色的許可。

AWS CodeStar 不支援服務連結角色。

服務角色

此功能可讓服務代表您擔任服務角色。此角色可讓服務存取其他服務中的資源，以代表您完成動作。服務角色會出現在您的 IAM 帳戶中，且由該帳戶所擁有。這表示 管理員可以變更此角色的許可。不過，這樣可能會破壞此服務的功能。

AWS CodeStar 支援服務角色。在為您的專案建立和管理資源時， AWS CodeStar 會使用服務角色 aws-codestar-service-role。如需詳細資訊，請參閱《IAM 使用者指南》中的角色術語和概念。

> **Important**
>
> 您必須以 管理員使用者或根帳戶身分登入來建立此服務角色。如需詳細資訊，請參閱《IAM 使用者指南》中的僅限第一次存取：您的根使用者登入資料和建立您的第一個管理員使用者和群組。

當您第一次在其中建立專案時，就會為您建立此角色 AWS CodeStar。此服務角色將代表您：

- 建立您在建立專案時選擇的資源。
- 在 AWS CodeStar 專案儀表板中顯示這些資源的相關資訊。

它也可以在您管理專案的資源時代表您。如需此政策陳述式的範例，請參閱 AWSCodeStarServiceRole 政策。

此外， 會根據專案類型 AWS CodeStar 建立多個專案特定的服務角色。 AWS CloudFormation 和 工具鏈角色會針對每個專案類型建立。

- AWS CloudFormation 角色 AWS CodeStar 允許 存取 AWS CloudFormation 來建立和修改 AWS CodeStar 專案的堆疊。
- 工具鏈角色 AWS CodeStar 允許 存取其他 AWS 服務，為您的 AWS CodeStar 專案建立和修改資源。

AWS CodeStar 專案層級政策和許可

當您建立專案時， 會 AWS CodeStar 建立管理專案資源所需的 IAM 角色和政策。政策可分為三個類別：

- 用於專案團隊成員的 IAM 政策。
- 用於工作者角色的 IAM 政策。
- 用於執行時間執行角色的 IAM 政策。

用於專案團隊成員的 IAM 政策

當您建立專案時， 會為專案的擁有者、參與者和檢視者 AWS CodeStar 建立三個客戶受管政策。所有 AWS CodeStar 專案都包含這三個存取層級的 IAM 政策。這些存取層級是專案特定的，並由具有標準名稱的 IAM 受管政策定義，其中 *project-id* 是 AWS CodeStar 專案的 ID （例如*my-first-projec*）：

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

> **Important**
>
> 這些政策可能會有所變更 AWS CodeStar。不得以手動方式編輯這些類別。如果您想要新增或變更許可，請將其他政策連接至 IAM 使用者。

新增專案團隊成員 (IAM 使用者) 到專案並選擇他們的存取層級時，對應的政策會連接到該 IAM 使用者，授與該使用者一組適當許可，能夠對專案資源執行動作。在大多數情況下，您不需要直接連接或管理 IAM 中的政策或許可。不建議手動將 AWS CodeStar 存取層級政策連接至 IAM 使用者。如果絕對需要， AWS CodeStar 您可以建立自己的受管或內嵌政策，將自己的許可層級套用至 IAM 使用者。

政策將緊密限定在專案資源和特定動作。當新的資源新增至基礎設施堆疊時，如果新資源是其中一個支援的資源類型， 會 AWS CodeStar 嘗試更新團隊成員政策，以包含存取新資源的許可。

> **Note**
>
> AWS CodeStar 專案中存取層級的政策僅適用於該專案。這有助於確保使用者只能在其角色決定的層級，查看他們擁有許可的 AWS CodeStar 專案並與之互動。只有建立 AWS CodeStar 專案的使用者才能套用允許存取所有 AWS CodeStar 資源的政策，無論專案為何。

所有 AWS CodeStar 存取層級政策都不同，取決於與存取層級相關聯之專案相關聯的 AWS 資源。不同於其他 AWS 服務，這些政策是在建立專案時自訂的，並隨著專案資源變更而更新。因此，並沒有正式的擁有者、作者群或檢視者受管政策。

AWS CodeStar 擁有者角色政策

`CodeStar_project-id_Owner` 客戶受管政策可讓使用者在 AWS CodeStar 專案中執行所有動作，而不受限制。這是唯一允許使用者新增或移除團隊成員的政策。政策的內容可能因與專案關聯的資源而有所不同。如需範例，請參閱AWS CodeStar 擁有者角色政策。

具有此政策的 IAM 使用者可以執行專案中的所有 AWS CodeStar 動作，但與具有`AWSCodeStarFullAccess`政策的 IAM 使用者不同，使用者無法建立專案。`codestar:*` 許可範圍僅限於特定資源 （與該 AWS CodeStar 專案 ID 相關聯的專案）。

AWS CodeStar 貢獻者角色政策

`CodeStar_project-id_Contributor` 客戶受管政策允許使用者構成專案並變更專案儀表板，但不允許使用者新增或移除團隊成員。政策的內容可能因與專案關聯的資源而有所不同。如需範例，請參閱AWS CodeStar 貢獻者角色政策。

AWS CodeStar 檢視者角色政策

`CodeStar_project-id_Viewer` 客戶受管政策允許使用者檢視 AWS CodeStar中的專案，但不能變更其資源或新增或移除團隊成員。政策的內容可能因與專案關聯的資源而有所不同。如需範例，請參閱AWS CodeStar 檢視器角色政策 。

用於工作者角色的 IAM 政策

如果您在 2018 年 12 月 6 日 PDT 之後建立 AWS CodeStar 專案，AWS CodeStar 會建立兩個工作者角色 `CodeStar-project-id-ToolChain`和 `CodeStar-project-id-CloudFormation`。工作者角色是專案特定的 IAM 角色，可 AWS CodeStar 建立 以傳遞至服務。它會授予許可，讓服務可以在 AWS CodeStar 專案內容中建立資源和執行動作。工具鏈工作者角色與 CodeBuild、CodeDeploy 和 CodePipeline 等工具鏈服務建立信任關係。專案團隊成員 (擁有者和貢獻者) 會獲授與許可，可將工作者角色傳遞至信任的下游服務。如需此角色的內嵌政策陳述式範例，請參閱AWS CodeStar 工具鏈工作者角色政策 (2018 年 12 月 6 日 PDT 之後)。

CloudFormation 工作者角色包含 所支援之所選資源的許可 AWS CloudFormation，以及在應用程式堆疊中建立 IAM 使用者、角色和政策的許可。它也有與 建立的信任關係 AWS CloudFormation。為了降低權限提升和破壞性動作的風險， AWS CloudFormation 角色政策包含的條件需要在基礎設施堆疊中建立的每個 IAM 實體 （使用者或角色） 的專案特定許可界限。如需此角色的內嵌政策陳述式範例，請參閱AWS CloudFormation 工作者角色政策。

對於 2018 年 12 月 6 日之前建立的 AWS CodeStar 專案，PDT 會為 CodePipeline、CodeBuild 和 CloudWatch Events 等工具鏈資源 AWS CodeStar 建立個別工作者角色，並建立 AWS CloudFormation 支援一組有限資源的工作者角色。這每個角色已與對應的服務建立信任關係。專案團隊成員 (擁有者和貢獻者) 和與些其他工作者角色會獲授與許可，可將角色傳遞至信任的下游服務。工作者角色的許可會在內嵌政策中定義，它可將範圍限縮在角色可以對一組專案資源執行的一組基本動作。這些許可是靜態的。它們包含專案建立時包含資源的許可，但不會在對專案新增新資源時更新。如需這些政策陳述式的範例，請參閱：

- AWS CloudFormation 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- AWS CodePipeline 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- AWS CodeBuild 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- Amazon CloudWatch Events 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)

執行角色的 IAM 政策

對於 2018 年 12 月 6 日 PDT 之後建立的專案，AWS CodeStar 會為應用程式堆疊中的範例專案建立一般執行角色。系統會使用許可界限政策，將該角色範圍限縮在專案資源。當您展開範例專案時，您可以建立其他 IAM 角色，而且 AWS CloudFormation 角色政策要求使用許可界限來縮小這些角色的範圍，以避免權限升級。如需詳細資訊，請參閱將 IAM 角色新增至專案。

對於在 2018 年 12 月 6 日之前建立的 Lambda 專案， AWS CodeStar 會建立 Lambda 執行角色，該角色具有內嵌政策，並附加對專案 AWS SAM 堆疊中資源採取行動的許可。當新資源新增至 SAM 範本

時，如果新資源是其中一個支援的資源類型， 會 AWS CodeStar 嘗試更新 Lambda 執行角色政策，以包含新資源的許可。

IAM 許可界限

2018 年 12 月 6 日 PDT 之後，當您建立專案時，AWS CodeStar 會建立客戶受管政策，並將該政策指派為專案中的 IAM 角色的 IAM 許可界限。AWS CodeStar 要求在應用程式堆疊中建立的所有 IAM 實體具有許可界限。許可界限可控制角色可以有的最大許可，但無法提供該角色任何許可。許可政策可定義角色的許可。這表示無論新增多少額外的許可至角色，使用該角色的任何人都無法執行超過許可界限中包含的動作。如需有關如何評估許可政策和許可界限的資訊，請參閱《IAM 使用者指南》中的政策評估邏輯。

AWS CodeStar 使用專案特定的許可界限，以防止將權限提升至專案外部的資源。AWS CodeStar 許可界限包含專案資源ARNs。如需此政策陳述式的範例，請參閱AWS CodeStar 許可界限政策。

當您透過應用程式堆疊 () 從專案新增或移除支援的資源時，AWS CodeStar 轉換會更新此政策`template.yml`。

新增 IAM 許可界限至現有的專案

如果您有在 2018 年 12 月 6 日之前建立的 AWS CodeStar 專案，您應該手動將許可界限新增至專案中的 IAM 角色。做為最佳實務，建議您使用只包含專案中資源的專案特定界限，以防止將權限提升至專案外部的資源。請依照下列步驟使用隨著專案演進而更新的 AWS CodeStar 受管許可界限。

1. 登入 AWS CloudFormation 主控台，並尋找專案中工具鏈堆疊的範本。此範本名為 `awscodestar-`*`project-id`*。
2. 依序選擇範本、Actions (動作) 和 View/Edit template in Designer (在 Designer 中檢視/編輯範本)。
3. 找到 `Resources` 區段，並在區段上方包含下列程式碼片段。

```
 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}-*'
```

您可能需要額外的 IAM 許可，才能從 AWS CloudFormation 主控台更新堆疊。

4. （選用） 如果您想要建立應用程式特定的 IAM 角色，請完成此步驟。從 IAM 主控台，更新連接至專案 AWS CloudFormation 角色的內嵌政策，以包含下列程式碼片段。您可能需要其他 IAM 資源來更新政策。

```
      {
           "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. 透過專案管道推送變更，讓 AWS CodeStar 使用適當的許可更新許可界限。

如需詳細資訊，請參閱將 IAM 角色新增至專案。

AWS CodeStar 身分型政策範例

根據預設，IAM 使用者和角色沒有建立或修改 AWS CodeStar 資源的許可。他們也無法使用 AWS Management Console AWS CLI或 AWS API 執行任務。管理員必須建立 IAM 政策，授與使用者和角色在指定資源上執行特定 API 操作所需的許可。管理員接著必須將這些政策連接至需要這些許可的 IAM 使用者或群組。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的在 JSON 標籤上建立政策。

主題

- 政策最佳實務
- AWSCodeStarServiceRole 政策
- AWSCodeStarFullAccess 政策
- AWS CodeStar 擁有者角色政策
- AWS CodeStar 貢獻者角色政策
- AWS CodeStar 檢視器角色政策
- AWS CodeStar 工具鏈工作者角色政策 (2018 年 12 月 6 日 PDT 之後)
- AWS CloudFormation 工作者角色政策

- AWS CloudFormation 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- AWS CodePipeline 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- AWS CodeBuild 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- Amazon CloudWatch Events 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)
- AWS CodeStar 許可界限政策
- 列出專案的資源
- 使用 AWS CodeStar 主控台
- 允許使用者檢視自己的許可
- 更新 AWS CodeStar 專案
- 新增團隊成員到專案
- 列出與 AWS 帳戶相關聯的使用者設定檔
- 根據標籤檢視 AWS CodeStar 專案
- AWS CodeStarAWS 受管政策的更新

政策最佳實務

身分型政策會判斷您帳戶中的某個人員是否可以建立、存取或刪除 AWS CodeStar 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管政策並邁向最低權限許可 – 若要開始將許可授予您的使用者和工作負載，請使用將許可授予許多常見使用案例的 AWS 受管政策。它們可在您的 中使用 AWS 帳戶。我們建議您定義特定於使用案例 AWS 的客戶受管政策，以進一步減少許可。如需更多資訊，請參閱 IAM 使用者指南中的 AWS 受管政策或任務職能的AWS 受管政策。
- 套用最低權限許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的許可。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 IAM 中的政策和許可。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。如果透過特定 使用服務動作 AWS 服務，您也可以使用條件來授予存取，例如 AWS CloudFormation。如需詳細資訊，請參閱 IAM 使用者指南中的 IAM JSON 政策元素：條件。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您撰寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 IAM Access Analyzer 政策驗證。

- 需要多重要素驗證 (MFA) – 如果您的案例需要 IAM 使用者或 中的根使用者 AWS 帳戶，請開啟 MFA 以提高安全性。如需在呼叫 API 操作時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 IAM 使用者指南中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 IAM 安全最佳實務。

AWSCodeStarServiceRole 政策

`aws-codestar-service-role` 政策會連接到服務角色， AWS CodeStar 允許 與其他 服務一起執行動作。第一次登入時 AWS CodeStar，您會建立 服務角色。您只需要建立一次。政策會在您建立服務角色之後自動連接到服務角色。

```
{
    "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 政策

在設定 AWS CodeStar指示中，您已將名為 的政策連接至 AWSCodeStarFullAccess IAM 使用者。此政策陳述式可讓使用者在 中執行所有可用的動作， AWS CodeStar 以及與 AWS 帳戶相關聯的所有可用 AWS CodeStar 資源。這包括建立和刪除專案。下列範例是代表性 AWSCodeStarFullAccess 政策的程式碼片段。實際政策會根據您在啟動新 AWS CodeStar 專案時選取的範本而有所不同。

AWS CloudFormation 在呼叫 時需要cloudformation::ListStacks許可，cloudformation::DescribeStacks而不需要目標堆疊。

許可詳細資訊

此政策包含執行下列動作的許可：

- ec2–擷取 EC2 執行個體的相關資訊以建立 AWS CodeStar 專案。
- cloud9–擷取 AWS Command Line Interface 環境的相關資訊。
- cloudformation–擷取 AWS CodeStar 專案堆疊的相關資訊。
- codestar–在 AWS CodeStar 專案中執行動作。

```
{
  "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-*"
      ]
    }
  ]
}
```

您可能不會想要提供所有使用者如此多的存取權限。反之，您可以使用 管理的專案角色來新增專案層級許可 AWS CodeStar。角色會授予特定層級的 AWS CodeStar 專案存取權，並命名如下：

- Owner
- 作者群
- 觀眾

AWS CodeStar 擁有者角色政策

AWS CodeStar 擁有者角色政策可讓使用者在 AWS CodeStar 專案中執行所有動作，而不受限制。AWS CodeStar 會將`CodeStar_`*`project-id`*`_Owner`政策套用到具有擁有者存取層級的專案團隊成員。

```
...
{
  "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 貢獻者角色政策

AWS CodeStar 貢獻者角色政策可讓使用者貢獻專案，並變更專案儀表板。AWS CodeStar 會將CodeStar_*project-id*_Contributor政策套用到具有貢獻者存取層級的專案團隊成員。具有參與者存取的使用者可以參與專案和變更專案儀表板，但無法新增或移除專案成員。

```
...
{
  "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 檢視器角色政策

AWS CodeStar 檢視器角色政策允許使用者在 AWS CodeStar 中檢視專案。AWS CodeStar 會將CodeStar_*project-id*_Viewer政策套用至具有檢視器存取層級的專案團隊成員。具有檢視器存取權的使用者可以在 AWS CodeStar 中檢視專案，但無法變更其資源，或新增或移除團隊成員。

```
...
{
  "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 工具鏈工作者角色政策 (2018 年 12 月 6 日 PDT 之後)

對於 2018 年 12 月 6 日 PDT 之後建立的 AWS CodeStar 專案，AWS CodeStar 會為工作者角色建立內嵌政策，以在其他 AWS 服務中為您的專案建立資源。政策的內容取決於您要建立的專案類型。以下政策為一個範例。如需詳細資訊，請參閱用於工作者角色的 IAM 政策。

```
{
  "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 工作者角色政策

對於 2018 年 12 月 6 日 PDT 之後建立的 AWS CodeStar 專案，AWS CodeStar 會為工作者角色建立內嵌政策，以為您的 AWS CodeStar 專案建立 AWS CloudFormation 資源。政策的內容取決於您的專案所需的資源類型。以下政策為一個範例。如需詳細資訊，請參閱用於工作者角色的 IAM 政策。

```
{
{
    "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 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)

如果您的 AWS CodeStar 專案是在 2018 年 12 月 6 日之前建立的，AWS CodeStar 會為 AWS CloudFormation 工作者角色建立內嵌政策。下列政策陳述式一個範例。

```
{
```

```
    "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 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)

如果您的 AWS CodeStar 專案是在 2018 年 12 月 6 日之前建立的，AWS CodeStar 會為 CodePipeline 工作者角色建立內嵌政策。下列政策陳述式一個範例。

```
{
    "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 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)

如果您的 AWS CodeStar 專案是在 2018 年 12 月 6 日之前建立的，AWS CodeStar 會為 CodeBuild 工作者角色建立內嵌政策。下列政策陳述式一個範例。

```
{
    "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 工作者角色政策 (2018 年 12 月 6 日之前的 PDT)

如果您的 AWS CodeStar 專案是在 2018 年 12 月 6 日之前建立的，AWS CodeStar 會為 CloudWatch Events 工作者角色建立內嵌政策。下列政策陳述式一個範例。

```
{
    "Statement": [
        {
            "Action": [
                "codepipeline:StartPipelineExecution"
            ],
            "Resource": [
                "arn:aws:codepipeline:us-east-1:account-id:project-id-Pipeline"
            ],
            "Effect": "Allow"
        }
    ]
}
```

AWS CodeStar 許可界限政策

如果您在 2018 年 12 月 6 日 PDT 之後建立 AWS CodeStar 專案，AWS CodeStar 會為您的專案建立許可界限政策。此政策可防止將權限提升至專案外部的資源。這是一個動態政策，會隨著專案演進更新。政策的內容取決於您要建立的專案類型。以下政策為一個範例。如需詳細資訊，請參閱IAM 許可界限。

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

列出專案的資源

在此範例中，您想要授予 AWS 帳戶中指定的 IAM 使用者存取權，以列出 AWS CodeStar 專案的資源。

```
{
  "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 主控台

存取 AWS CodeStar 主控台不需要特定許可，但除非您有 AWSCodeStarFullAccess政策或其中一個 AWS CodeStar 專案層級角色：擁有者、貢獻者或檢視器，否則無法執行任何有用的動作。如需 AWSCodeStarFullAccess 的詳細資訊，請參閱 AWSCodeStarFullAccess 政策。如需專案層級政策的詳細資訊，請參閱用於專案團隊成員的 IAM 政策。

對於僅呼叫 AWS CLI 或 AWS API 的使用者，您不需要允許最低主控台許可。反之，只需允許存取符合您嘗試執行之 API 操作的動作就可以了。

允許使用者檢視自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此政策包含在主控台上完成此動作的許可，或使用 AWS CLI 或 AWS API 以程式設計方式完成此動作的許可。

```
{
    "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 專案

在此範例中，您想要授予 AWS 帳戶中指定的 IAM 使用者存取權，以編輯 AWS CodeStar 專案的屬性，例如其專案描述。

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:UpdateProject"
      ],
      "Resource" : "arn:aws:codestar:us-east-2:project/my-first-projec"
    }
  ]
}
```

新增團隊成員到專案

在此範例中，您想要授予指定的 IAM 使用者將團隊成員新增至 AWS CodeStar 專案 ID *my-first-projec* 的專案，但明確拒絕該使用者移除團隊成員的能力：

```
{
  "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"
    }
      ]

  ]
}
```

列出與 AWS 帳戶相關聯的使用者設定檔

在此範例中，您可以允許已連接此政策的 IAM AWS CodeStar 使用者列出與 AWS 帳戶相關聯的所有使用者設定檔：

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:ListUserProfiles",
              ],
      "Resource" : "*"
    }
  ]
}
```

根據標籤檢視 AWS CodeStar 專案

您可以在身分型政策中使用條件，根據標籤控制對 AWS CodeStar 專案的存取。此範例會示範如何建立政策，允許檢視專案。但是，只有在專案標籤 `Owner` 的值是該使用者的使用者名稱時，才會授予許可。此政策也會授予在主控台完成此動作的必要許可。

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

您可以將此政策連接到您帳戶中的 IAM 使用者。如果名為 的使用者`richard-roe`嘗試檢視 AWS CodeStar 專案，則該專案必須加上標籤 `Owner=richard-roe`或 `owner=richard-roe`。否則他便會被拒絕存取。條件標籤鍵 `Owner` 符合 `Owner` 和 `owner`，因為條件索引鍵名稱不區分大小寫。如需詳細資訊，請參閱《IAM 使用者指南》中的 IAM JSON 政策元素：條件。

AWS CodeStarAWS 受管政策的更新

檢視自此服務開始追蹤這些變更以來，AWS CodeStar AWS 受管政策更新的詳細資訊。如需此頁面變更的自動提醒，請訂閱 AWS CodeStar 文件歷史記錄頁面上的 RSS 摘要。

變更	描述	日期
AWSCodeStarFullAccess 政策 – 更新 AWSCodeStarFullAccess 政策	已更新 AWS CodeStar 存取角色政策。政策的結果相同，但除了已經需要的 DescribeStacks 之外， cloudformation 還需要 ListStacks。DescribeStacks	2023 年 3 月 24 日
AWSCodeStarServiceRole 政策 – 更新 AWSCodeStarServiceRole 政策	AWS CodeStar 服務角色的政策已更新，以更正政策陳述式中的備援動作。 服務角色政策允許 AWS CodeStar 服務代表您執行動作。	2021 年 9 月 23 日

變更	描述	日期
AWS CodeStar 開始追蹤變更	AWS CodeStar 開始追蹤其 AWS 受管政策的變更。	2021 年 9 月 23 日

故障診斷 AWS CodeStar Identity and Access

使用以下資訊來協助您診斷和修正使用 AWS CodeStar 和 IAM 時可能遇到的常見問題。

主題

我無權在 AWS CodeStar 中執行動作

如果 AWS Management Console 告訴您無權執行動作，請聯絡您的管理員尋求協助。您的管理員提供您的簽署憑證。

以下範例錯誤會在 `mateojackson` IAM 使用者嘗試使用主控台檢視 *`widget`* 的詳細資訊，但卻沒有 `codestar:`*`GetWidget`* 許可時發生。

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

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *`my-example-widget`* 動作存取 `codestar:`*`GetWidget`* 資源。

我未獲得執行 iam:PassRole 的授權

如果您收到錯誤，告知您無權執行`iam:PassRole`動作，則必須更新您的政策，以允許您將角色傳遞給 AWS CodeStar。

有些 AWS 服務 可讓您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為 的 IAM `marymajor` 使用者嘗試使用主控台在 AWS CodeStar 中執行動作時，會發生下列範例錯誤。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

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

在這種情況下，Mary 的政策必須更新，允許她執行 `iam:PassRole` 動作。

如果您需要協助，請聯絡您的 AWS 管理員。您的管理員提供您的簽署憑證。

我想要允許 AWS 帳戶外的人員存取我的 AWS CodeStar 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要了解 AWS CodeStar 是否支援這些功能，請參閱 AWS CodeStar 如何與 IAM 搭配使用。
- 若要了解如何在您擁有 AWS 帳戶 的 資源之間提供存取權，請參閱《IAM 使用者指南》中的在您擁有 AWS 帳戶 的另一個 中提供存取權給 IAM 使用者。
- 若要了解如何將資源的存取權提供給第三方 AWS 帳戶，請參閱《IAM 使用者指南》中的將存取權提供給第三方 AWS 帳戶 擁有的 。
- 如需了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的將存取權提供給在外部進行身分驗證的使用者 (聯合身分)。
- 如需了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱《IAM 使用者指南》中的 IAM 中的跨帳戶資源存取。

使用 記錄 AWS CodeStar API 呼叫 AWS CloudTrail

AWS CodeStar 已與 服務整合 AWS CloudTrail，此服務提供由使用者、角色或 AWS 服務在其中採取的動作記錄 AWS CodeStar。CloudTrail 會將 的所有 API 呼叫擷取 AWS CodeStar 為事件。擷取的呼叫包括從 AWS CodeStar 主控台的呼叫，以及對 AWS CodeStar API 操作的程式碼呼叫。如果您建立線索，您可以啟用 CloudTrail 事件持續交付至 S3 儲存貯體，包括 的事件 AWS CodeStar。即使您未設定追蹤，依然可以透過 CloudTrail 主控台中的事件歷史記錄檢視最新事件。使用 CloudTrail 收集的資訊，您可以判斷提出的請求 AWS CodeStar、提出請求的 IP 地址、提出請求的人員、提出請求的時間，以及其他詳細資訊。

若要進一步了解 CloudTrail，請參閱AWS CloudTrail 《使用者指南》。

AWS CodeStar CloudTrail 中的資訊

當您建立 AWS 帳戶時，會在您的帳戶上啟用 CloudTrail。當活動在 中發生時 AWS CodeStar，該活動會記錄在 CloudTrail 事件中，以及事件歷史記錄中的其他服務 AWS 事件。您可以在 AWS 帳戶中檢視、搜尋和下載最近的事件。如需詳細資訊，請參閱《使用 CloudTrail 事件歷史記錄檢視事件》https://docs.aws.amazon.com/awscloudtrail/latest/userguide/view-cloudtrail-events.html。

若要持續記錄您 AWS 帳戶中的事件，包括 的事件 AWS CodeStar，請建立追蹤。根據預設，當您在主控台中建立追蹤時，該追蹤會套用至所有 AWS 區域。線索會記錄 AWS 分割區中所有區域的事件，並將日誌檔案交付至您指定的 S3 儲存貯體。您可設定其他 AWS 服務，進一步分析和處理 CloudTrail 記錄中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- 建立追蹤的概觀
- CloudTrail 支援的服務和整合
- 設定 CloudTrail 的 Amazon SNS 通知
- 從多個區域接收 CloudTrail 日誌檔案，以及從多個帳戶接收 CloudTrail 日誌檔案

CloudTrail 會記錄所有 AWS CodeStar 動作，並記錄在 AWS CodeStar API 參考中。例如，對 `DescribeProject`、`UpdateProject` 以及 `AssociateTeamMember` 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌專案都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 請求是否由其他 AWS 服務提出。

如需詳細資訊，請參閱 CloudTrail userIdentity 元素。

了解 AWS CodeStar 日誌檔案項目

CloudTrail 日誌檔案包含一或多個日誌專案。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔並非依公有 API 呼叫的堆疊追蹤排序，因此不會以任何特定順序出現。

下列範例顯示 CloudTrail 日誌項目，示範正在呼叫`CreateProject`的操作 AWS CodeStar：

```
{
```

```
  "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"
}
```

的合規驗證 AWS CodeStar

AWS CodeStar 不在任何 AWS 合規計劃的範圍內。

如需特定合規計劃範圍內 AWS 的服務清單，請參閱AWS 合規計劃範圍內的服務。如需一般資訊，請參閱 AWS 合規計劃。

您可以使用 下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱在 AWS 成品中下載報告。

中的彈性 AWS CodeStar

AWS 全球基礎設施是以 AWS 區域和可用區域為基礎建置的。 AWS 區域提供多個實體隔離和隔離的可用區域，這些區域以低延遲、高輸送量和高度備援聯網連接。透過可用區域，您所設計與操作的應用程式和資料庫，就能夠在可用區域之間自動容錯移轉，而不會發生中斷。可用區域的可用性、容錯能力和擴充能力，均較單一或多個資料中心的傳統基礎設施還高。

如需 AWS 區域和可用區域的詳細資訊，請參閱 AWS 全球基礎設施。

中的基礎設施安全 AWS CodeStar

作為受管服務，AWS CodeStar 受到 AWS 全球網路安全的保護。如需 AWS 安全服務以及 如何 AWS 保護基礎設施的相關資訊，請參閱AWS 雲端安全。若要使用基礎設施安全的最佳實務來設計您的 AWS 環境，請參閱安全支柱 AWS Well-Architected Framework 中的基礎設施保護。

您可以使用 AWS 已發佈的 API 呼叫，透過網路存取 CodeStar。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman)或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 AWS Security Token Service (AWS STS) 來產生暫時安全憑證來簽署請求。

根據預設， AWS CodeStar 不會隔離服務流量。除非您透過 Amazon EC2、API Gateway 或 Elastic Beanstalk 手動修改存取設定，否則使用 建立的專案 AWS CodeStar 會開放給公有網際網路。這是刻意的。您可以將 Amazon EC2、API Gateway 或 Elastic Beanstalk 中的存取設定修改為您想要的程度，包括防止所有網際網路存取。

AWS CodeStar 根據預設， 不提供 VPC 端點 (AWS PrivateLink) 的支援，但您可以直接在專案資源上設定該支援。

中的限制 AWS CodeStar

下表說明 in. AWS CodeStar AWS CodeStar depends 對 AWS 專案資源其他服務的限制。這些服務限制有些可以變更。如需可變更之限制的詳細資訊，請參閱 AWS 服務限制。

專案數目	AWS 帳戶中最多 333 個專案。實際限制會根據其他服務相依性的層級而有所不同 （例如，您 AWS 帳戶在 CodePipeline 中允許的管道數量上限）。
IAM 使用者可以所屬的 AWS CodeStar 專案數量	每個 IAM 使用者最多 10 個。
專案 ID	專案 IDs AWS 帳戶中必須是唯一的。專案 ID 必須至少有 2 個字元，且不能超過 15 個字元。允許的字元包含： 字母 a 到 z，內含。 數字 0 到 9，內含。 特殊字元 - (負號)。 不允許任何其他字元，例如大寫字母、空格、. (句號)、@ (at 符號) 或 _ (底線)。
專案名稱	專案名稱不能超過 100 個字元，而且不能以空格開始或結束。
專案說明	字元的任何組合，長度介於 0 到 1,024 個字元之間。專案說明為選擇性。
AWS CodeStar 專案中的團隊成員	100
使用者描述檔中的顯示名稱	字元的任何組合，長度介於 1 到 100 個字元之間。顯示名稱必須至少包含一個字元。該字元不得為空格。顯示名稱不能以空格開始或結束。

使用者描述檔中的電子郵件地址	電子郵件地址必須包含 @，且結尾是有效的網域域名。
AWS CodeStar的聯合身分存取權、根帳戶存取權或暫時存取權	AWS CodeStar 支援聯合身分使用者和使用臨時存取憑證。不建議 AWS CodeStar 搭配根帳戶使用 。
IAM 角色	連接到 IAM 角色的任何受管政策中最多 5，120 個字元。

故障診斷 AWS CodeStar

以下資訊可能有助於診斷 AWS CodeStar內的常見問題。

主題

專案建立失敗：專案未建立

問題：當嘗試建立專案時，您看到訊息表示建立失敗。

可行的修正：最常見的故障原因為：

- AWS 帳戶中已存在具有該 ID 的專案，可能位於不同的 AWS 區域。
- 您用來登入 的 IAM 使用者 AWS Management Console 沒有建立專案所需的許可。
- AWS CodeStar 服務角色缺少一或多個必要的許可。
- 您已達到專案一或多個資源的上限（例如 IAM、Amazon S3 儲存貯體或 CodePipeline 中管道的客戶受管政策限制）。

建立專案之前，請確認您已將AWSCodeStarFullAccess政策套用至 IAM 使用者。如需詳細資訊，請參閱 AWSCodeStarFullAccess 政策。

當您建立專案時，確保 ID 是唯一且符合 AWS CodeStar 需求。請確定您已選取AWS CodeStar 想要代表您管理 AWS 資源的許可核取方塊。

若要疑難排解其他問題，請開啟 AWS CloudFormation 主控台，選擇您嘗試建立之專案的堆疊，然後選擇事件索引標籤。可能會有一個以上的專案堆疊。堆疊名稱以 awscodestar- 開頭，後面緊接著專案 ID。堆疊可能會在刪除篩選條件檢視下方。檢閱堆疊事件中的任何故障訊息，並更正列為這些故障的問題原因。

專案建立：我在建立專案時嘗試編輯 Amazon EC2 組態時看到錯誤

問題：當您在專案建立期間編輯 Amazon EC2 組態選項時，您會看到錯誤訊息或灰色選項，無法繼續建立專案。

可行的修正：最常見的錯誤訊息原因為：

- AWS CodeStar 專案範本中的 VPC （預設 VPC 或編輯 Amazon EC2 組態時使用的 VPC) 具有專用執行個體租用，且專用執行個體不支援執行個體類型。選擇不同的執行個體類型或不同的 Amazon VPC。
- AWS 您的帳戶沒有 Amazon VPCs。您可能已刪除預設的 VPC，而不是建立任何其他項目。開啟 Amazon VPC 主控台，網址為 https：// https://console.aws.amazon.com/vpc/ VPC，選擇您的 VPCs，並確定您至少設定一個 VPC。若沒有，請建立一個。如需詳細資訊，請參閱《Amazon VPC 入門指南》中的 Amazon Virtual Private Cloud 概觀。
- Amazon VPC 沒有任何子網路。選擇不同的 VPC，或建立 VPC 的子網路。如需詳細資訊，請參閱 VPC 和子網路基本概念。

專案刪除： AWS CodeStar 專案已刪除，但資源仍然存在

問題： AWS CodeStar 專案已刪除，但為該專案建立的資源仍然存在。根據預設， AWS CodeStar 會在刪除專案時刪除專案資源。即使使用者選取刪除資源核取方塊，Amazon S3 儲存貯體等部分資源仍會保留，因為儲存貯體可能包含資料。

可能的修正：開啟 AWS CloudFormation 主控台，並尋找一或多個用於建立專案的 AWS CloudFormation 堆疊。堆疊名稱以 awscodestar- 開頭，後面緊接著專案 ID。堆疊可能在刪除篩選條件檢視下方。檢閱與堆疊相關聯的事件，探索為專案建立的資源。在您 AWS 建立 AWS CodeStar 專案的區域中，開啟每個資源的 主控台，然後手動刪除資源。

專案資源可能仍包括：

- Amazon S3 中的一或多個專案儲存貯體。與其他專案資源不同，選取刪除相關聯的資源與專案核取方塊時，不會刪除 Amazon S3 中的專案儲存貯體。 AWS AWS CodeStar

  開啟位於 https://console.aws.amazon.com/s3/ 的 Amazon S3 主控台。
- CodeCommit 中專案的來源儲存庫。

  前往 https://console.aws.amazon.com/codecommit/ 開啟 CodeCommit 主控台。
- CodePipeline 中專案的管道。

  前往 https://console.aws.amazon.com/codepipeline/ 開啟 CodePipeline 主控台。
- CodeDeploy 中的應用程式和相關聯的部署群組。

  前往 https://console.aws.amazon.com/codedeploy/ 開啟 CodeDeploy 主控台。
- AWS Elastic Beanstalk中的應用程式和相關聯的環境。

  開啟 Elastic Beanstalk 主控台，網址為 https://console.aws.amazon.com/elasticbeanstalk/：//。
- AWS Lambda的函數。

  在 https AWS Lambda ：//https://console.aws.amazon.com/lambda/。
- APIs中的一或多個 API。

  在以下網址開啟 API Gateway 主控台：https://console.aws.amazon.com/apigateway/。
- IAM 中的一或多個 IAM 政策或角色。

  登入 AWS Management Console ，並在 https://console.aws.amazon.com/iam/：//www. 開啟 IAM 主控台。
- Amazon EC2 中的執行個體。

  在 https://console.aws.amazon.com/ec2/ 開啟 Amazon EC2 主控台。
- 一或多個開發環境 AWS Cloud9。

  若要檢視、存取和管理開發環境，請開啟 AWS Cloud9 主控台，網址為 https：//https://console.aws.amazon.com/cloud9/。

如果您的專案使用 以外的資源 AWS （例如，GitHub 儲存庫或 Atlassian JIRA 中的問題），即使已選取刪除關聯的資源與 CodeStar 專案方塊，也不會刪除這些 AWS 資源。

團隊管理失敗：無法將 IAM 使用者新增至 AWS CodeStar 專案中的團隊

問題：當嘗試將使用者加入到專案時，您看到錯誤訊息表示附加失敗。

可能的修正：此錯誤的最常見原因是使用者已達到可套用至 IAM 中使用者的受管政策限制。如果您在嘗試新增使用者的 AWS CodeStar 專案中沒有擁有者角色，或者 IAM 使用者不存在或已刪除，則也可能會收到此錯誤。

請確定您以該 AWS CodeStar 專案中擁有者的身分登入。如需詳細資訊，請參閱將團隊成員新增至 AWS CodeStar 專案 。

若要疑難排解其他問題，請開啟 IAM 主控台，選擇您嘗試新增的使用者，並檢查有多少受管政策套用到該 IAM 使用者。

如需詳細資訊，請參閱 &IAM; 實體和物件的限制。如需可變更之限制資訊，請參閱 AWS 服務限制。

存取失敗：聯合身分使用者無法存取 AWS CodeStar 專案

問題：聯合身分使用者無法在 AWS CodeStar 主控台中看到專案。

可行的修正：如果您以聯合身分使用者身分登入，請確定您有適當的受管政策連接到要登入的所擔任角色。如需詳細資訊，請參閱將專案的 AWS CodeStar Viewer/Contributor/Owner受管政策連接至聯合身分使用者的角色。

手動連接政策，將聯合身分使用者新增至您的 AWS Cloud9 環境。請參閱 將 AWS Cloud9 受管政策連接至聯合身分使用者的角色。

存取失敗：聯合身分使用者無法存取或建立 AWS Cloud9 環境

問題：聯合身分使用者無法在 AWS Cloud9 主控台中查看或建立 AWS Cloud9 環境。

可行的修正：如果您以聯合身分使用者身分登入，請確定您有適當的受管政策連接到聯合身分使用者的角色。

您可以透過手動將政策連接至聯合身分使用者的角色，將聯合身分使用者新增至您的 AWS Cloud9 環境。請參閱 將 AWS Cloud9 受管政策連接至聯合身分使用者的角色。

存取失敗：聯合身分使用者可以建立 AWS CodeStar 專案，但無法檢視專案資源

問題：聯合身分使用者能夠建立專案，但無法檢視專案資源，例如專案管道。

可能的修正：如果您已連接 **AWSCodeStarFullAccess**受管政策，則具有在其中建立專案的許可 AWS CodeStar。不過，若要存取所有專案資源，您必須連接擁有者受管政策。

AWS CodeStar 建立專案資源後，擁有者、貢獻者和檢視器受管政策中會提供所有專案資源的專案許可。若要存取所有資源，您必須手動將擁有者政策連接到您的角色。請參閱 步驟 3：設定使用者的 IAM 許可。

服務角色問題：無法建立服務角色

問題：當您嘗試在 中建立專案時 AWS CodeStar，您會看到一則訊息，提示您建立服務角色。當您選擇選項來建立角色時，您看到錯誤。

可能的修正：此錯誤的最常見原因是您使用沒有足夠的許可來建立服務角色 AWS 的帳戶登入 。若要建立 AWS CodeStar 服務角色 (`aws-codestar-service-role`)，您必須以管理使用者身分登入或使用根帳戶登入。登出 主控台，並使用已套用 `AdministratorAccess`受管政策的 IAM 使用者登入。

服務角色問題：此服務角色無效或遺失

問題：開啟 AWS CodeStar 主控台時，您會看到一則訊息，指出 AWS CodeStar 服務角色遺失或無效。

可行的修正：此錯誤最常見的原因是，管理使用者已編輯或刪除服務角色 (`aws-codestar-service-role`)。如果服務角色已刪除，系統會提示您建立該角色。您必須以管理使用者身分登入，或使用根帳戶登入，如此才能建立角色。如果該角色已遭編輯，便不再有效。以管理使用者身分登入 IAM 主控台，在角色清單中尋找服務角色，然後刪除該角色。切換到 AWS CodeStar 主控台，並依照指示建立服務角色。

專案角色問題： AWS CodeStar 專案中執行個體 AWS Elastic Beanstalk 的運作狀態檢查失敗

問題：如果您在 2017 年 9 月 22 日之前建立包含 Elastic Beanstalk 的 AWS CodeStar 專案，則 Elastic Beanstalk 運作狀態檢查可能會失敗。如果您自建立專案以來尚未變更 Elastic Beanstalk 組

態，運作狀態檢查會失敗並報告灰色狀態。儘管運作狀況檢查失敗，您的應用程式仍應如預期執行。如果您在建立專案後變更 Elastic Beanstalk 組態，運作狀態檢查會失敗，而且您的應用程式可能無法正確執行。

修正：一或多個 IAM 角色缺少必要的 IAM 政策陳述式。新增遺失的政策到您的 AWS 帳戶中受影響的角色。

1. 登入 AWS Management Console ，並在 https://console.aws.amazon.com/iam/：//www. 開啟 IAM 主控台。

   （如果您無法執行此操作，請洽詢您的 AWS 帳戶管理員以取得協助。）
2. 在導覽窗格中，選擇角色。
3. 在角色清單中，選擇 CodeStarWorker-***Project-ID***-EB，其中 *Project-ID* 是其中一個受影響專案的 ID。(如果您無法輕鬆找到清單中的角色，請在搜尋方塊中輸入部分或全部的角色名稱)。
4. 在 [權限] 索引標籤上，選擇 [連接政策]。
5. 在策略清單，選取 AWSElasticBeanstalkEnhancedHealth 和 AWSElasticBeanstalkService。(如果您無法輕鬆找到清單中的政策，請在搜尋方塊中輸入部分或全部的政策名稱)。
6. 選擇 Attach Policy (連接政策)。
7. 針對每個受影響角色重複步驟 3 到 6，其名稱格式為 CodeStarWorker-***Project-ID***-EB。

專案角色問題：服務角色無效或遺失

問題：當您嘗試新增使用者至專案，看到錯誤訊息，其表示附加失敗，因為專案角色政策遺失或無效。

可能的修正：此錯誤的最常見原因是在 IAM 中編輯或刪除一或多個專案政策。專案政策對 AWS CodeStar 專案是唯一的，無法重新建立。專案無法使用。在 中建立專案 AWS CodeStar，然後將資料遷移至新專案。從無法使用的專案的儲存庫複製專案程式碼，並將該程式碼推送到新專案的儲存庫。從舊專案將團隊 wiki 資訊複製到新的專案。新增使用者到新的專案。當您確定已遷移所有資料和設定，請刪除不可用的專案。

專案擴充：無法連接到 JIRA

問題：當您使用 Atlassian JIRA 延伸模組嘗試將 AWS CodeStar 專案連線至 JIRA 執行個體時，會顯示下列訊息：「URL 不是有效的 JIRA URL。請確認 URL 是否正確。」

可能的修正：

- 確定 JIRA URL 是正確的，然後再試一次連線。
- 您的自我託管 JIRA 執行個體可能無法從公有網際網路存取。聯絡您的網路管理員，確保您的 JIRA 執行個體可從公有網際網路存取，然後再次嘗試連線。

GitHub：無法存取儲存庫的遞交歷史記錄、問題或程式碼

問題：在將程式碼儲存在 GitHub 的專案儀表板中，遞交歷史記錄和 GitHub 連線錯誤圖磚會顯示連線錯誤，或是在這些圖磚中選擇 Open in GitHub (在 GitHub 中開啟) 或 Create issue (建立問題) 時顯示錯誤。

可能原因：

- AWS CodeStar 專案可能無法再存取 GitHub 儲存庫。
- 在 GitHub 中，儲存庫可能已遭到刪除或重新命名。

AWS CloudFormation：遺失許可的回復建立堆疊

在新增資源到 `template.yml` 檔案後，檢視任何錯誤訊息的 AWS CloudFormation 堆疊更新。如果未達特定條件 (例如，當必要的資源許可遺失時)，堆疊更新失敗。

> **Note**
>
> 自 2019 年 5 月 2 日起，我們已更新所有現有專案的 AWS CloudFormation 工作者角色政策。此更新會減少授予您專案管道的存取範圍，來改善您專案中的安全性。

若要疑難排解，請在專案管道的 AWS CodeStar 儀表板檢視中檢視失敗狀態。

接下來，選擇管道部署階段中的 CloudFormation 連結，在 AWS CloudFormation 主控台中排除故障。若要檢視堆疊建立詳細資訊，請展開專案的事件清單，並檢視任何失敗訊息。訊息會指出缺少哪些許可。修正 AWS CloudFormation 工作者角色政策，然後再次執行您的管道。

AWS CloudFormation 無權在 Lambda 執行角色上執行 iam：PassRole

如果您已在 2018 年 12 月 6 日之前建立建立 Lambda 函數的專案，您可能會看到如下 AWS CloudFormation 錯誤：

```
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)
```

發生此錯誤是因為您的 AWS CloudFormation 工作者角色沒有許可傳遞角色來佈建新的 Lambda 函數。

若要修正此錯誤，您需要使用下列程式碼片段更新您的 AWS CloudFormation 工作者角色政策。

```
{
        "Action":[ "iam:PassRole" ],
        "Resource": [
           "arn:aws:iam::account-id:role/CodeStarWorker-project-id-Lambda",
           ],

        "Effect": "Allow"
}
```

更新政策後，請再次執行管道。

或者，您也可以透過將許可界限新增至您的專案來使用 Lambda 函數的自訂角色，如 中所述 新增 IAM 許可界限至現有的專案

無法為 GitHub 儲存庫建立連線

問題：

由於 GitHub 儲存庫的連線使用 AWS Connector for GitHub，因此您需要儲存庫的組織擁有者許可或管理員許可，才能建立連線。

可能的修正方式：如需 GitHub 儲存庫許可層級的詳細資訊，請參閱 https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-organizations-and-teams/permission-levels-for-an-organization。

AWS CodeStar 使用者指南版本備註

下表說明 AWS CodeStar 使用者指南每個版本的重要變更。如需有關此文件更新的通知，您可以訂閱 RSS 訂閱源。

變更	描述	日期
存取政策更新	已更新 AWS CodeStar 存取角色政策。政策的結果相同，但除了已經需要的 DescribeStacks 之外， cloudformation 還需要 ListStacks。 DescribeStacks 若要參考更新的政策，請參閱 AWSCodeStarFullAccess 政策。	2023 年 3 月 24 日
服務角色政策更新	AWS CodeStar 服務角色政策已更新。若要參考更新的政策，請參閱 AWSCodeStarServiceRole 政策。	2021 年 9 月 23 日
使用具有 GitHub 來源儲存庫之專案的連線資源	當您使用主控台在 AWS CodeStar 中使用 GitHub 儲存庫建立專案時，會使用連線資源來管理您的 GitHub 動作。連線使用 GitHub 應用程式，而先前的 GitHub 授權使用 OAuth。如需說明如何建立使用 GitHub 連線的專案的教學課程，請參閱教學課程：使用 GitHub 來源儲存庫建立專案。本教學課程也說明如何建立、檢閱和合併專案來源儲存庫的提取請求。	2021 年 4 月 27 日

AWS CodeStar 支援美國西部 AWS Cloud9（加利佛尼亞北部）區域	AWS CodeStar 現在支援 AWS Cloud9 在美國西部（加利佛尼亞北部）區域使用 。如需詳細資訊，請參閱設定 Cloud9。	2021 年 2 月 16 日
更新文件以反映新的主控台體驗	在 2020 年 8 月 12 日，該 AWS CodeStar 服務已移至 AWS 主控台中的新使用者體驗。使用者指南已更新，以符合新的主控台體驗。	2020 年 8 月 12 日
AWS CodeStar 您可以使用 AWS CodeStar CLI 建立 專案	AWS CodeStar 您可以使用 CLI 命令來建立 專案。 會使用原始程式碼和您提供的工具鏈範本來 AWS CodeStar 建立您的專案和基礎設施。請參閱在 AWS CodeStar 中建立專案 (AWS CLI)。	2018 年 10 月 24 日
所有 AWS CodeStar 專案範本現在都包含基礎設施更新 AWS CloudFormation 的檔案	AWS CodeStar 與 AWS CloudFormation 搭配使用，可讓您使用程式碼在雲端建立支援服務和伺服器或無伺服器平台。 AWS CloudFormation 檔案現在適用於所有 AWS CodeStar 專案範本類型（使用 Lambda、EC2 或 Elastic Beanstalk 運算平台的範本）。此檔案存放於來源儲存庫中的 `template.yml` 。您可以檢視和修改檔案，新增資源到您的專案。請參閱專案範本。	2018 年 8 月 3 日

AWS CodeStar 使用者指南更新通知現在可透過 RSS 取得	AWS CodeStar 使用者指南的 HTML 版本現在支援 文件更新版本備註頁面中記錄的更新 RSS 摘要。RSS 摘要包含 2018 年 6 月 30 日以後所做的更新。先前發佈的更新仍可在 Documentation Update Release Notes (文件更新版本備註) 頁面中取得。使用頂部選單面板中的 RSS 按鈕來訂閱摘要。	2018 年 6 月 30 日

下表說明 AWS CodeStar 使用者指南 2018 年 6 月 30 日之前每個版本的重要變更。

變更	描述	變更日期
AWS CodeStar 使用者指南現已在 GitHub 上提供	本指南現已在 GitHub 上可用。您也可以使用 GitHub 提交意見回饋和變更本指南內容的請求。如需詳細資訊，請選擇 Edit on GitHub (在 GitHub 上編輯) 圖示，或參閱 GitHub 網站上的 awsdocs/aws-codestar-user-guide 儲存庫。	2018 年 2 月 22 日
AWS CodeStar 現可於亞太區域（首爾）使用	AWS CodeStar 現可於亞太區域（首爾）區域使用。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 AWS CodeStar。	2018 年 2 月 14 日
AWS CodeStar 現可於亞太區域（東京）和加拿大（中部）使用	AWS CodeStar 現可於亞太區域（東京）和加拿大（中部）區域使用。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 AWS CodeStar。	2017 年 12 月 20 日
AWS CodeStar 現在支援 AWS Cloud9	AWS CodeStar 現在支援使用 AWS Cloud9 Web 瀏覽器型線上 IDE 來使用專案程式碼。如需詳細資訊，請參閱AWS Cloud9 搭配 使用 AWS CodeStar。 如需支援的 AWS 區域清單，請參閱 AWS Cloud9 中的 Amazon Web Services 一般參考。	2017 年 11 月 30 日

變更	描述	變更日期
AWS CodeStar 現在支援 GitHub	AWS CodeStar 現在支援在 GitHub 中存放專案程式碼。如需詳細資訊，請參閱建立專案。	2017 年 10 月 12 日
AWS CodeStar 現已在美國西部（加利佛尼亞北部）和歐洲（倫敦）推出	AWS CodeStar 現已在美國西部（加利佛尼亞北部）和歐洲（倫敦）區域提供。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 AWS CodeStar。	2017 年 8 月 17 日
AWS CodeStar 現可於亞太區域（雪梨）、亞太區域（新加坡）和歐洲（法蘭克福）使用	AWS CodeStar 現可於亞太區域（雪梨）、亞太區域（新加坡）和歐洲（法蘭克福）區域使用。如需詳細資訊，請參閱 Amazon Web Services 一般參考 中的 AWS CodeStar。	2017 年 7 月 25 日
AWS CloudTrail 現在支援 AWS CodeStar	AWS CodeStar 現在已與 CloudTrail 整合，這項服務會擷取 AWS CodeStar 您 AWS 帳戶中由 或代表 發出的 API 呼叫，並將日誌檔案交付至您指定的 Amazon S3 儲存貯體。如需詳細資訊，請參閱使用 記錄 AWS CodeStar API 呼叫 AWS CloudTrail。	2017 年 6 月 14 日
初始版本	此為AWS CodeStar 使用者指南的第一版。	2017 年 4 月 19 日

AWS 詞彙表

如需最新的 AWS 術語，請參閱 AWS 詞彙表 參考中的AWS 詞彙表。