
AWS CodeStar

User Guide



AWS CodeStar: User Guide

Copyright © 2017 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What Is AWS CodeStar?	1
What Can I Do with AWS CodeStar?	1
A Quick Look at AWS CodeStar	2
How Do I Get Started with AWS CodeStar?	2
Setting Up	3
Step 1: Create an AWS Account	3
Step 2: Create the AWS CodeStar Service Role	3
Step 3: Create or Use an IAM User	4
Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects	4
Step 5: Open the AWS CodeStar Console	4
Next Steps	4
Getting Started with AWS CodeStar	5
Step 1: Create an AWS CodeStar Project	6
Step 2: Add Display Information for Your AWS CodeStar User Profile	10
Step 3: View Your Project	11
Step 4: Customize the Team Wiki Tile and the Project Dashboard	14
Step 5: Commit a Change	15
Step 6: Add More Team Members	17
Step 7: Clean Up	20
Next Steps	20
Serverless Project Tutorial	20
Overview	21
Step 1: Create the Project	22
Step 2: Explore Project Resources	24
Step 3: Test the Web Service	28
Step 4: Set Up Your Local Workstation to Edit Project Code	30
Step 5: Add Logic to the Web Service	32
Step 6: Test the Enhanced Web Service	33
Step 7: Add a Unit Test to the Web Service	34
Step 8: View Unit Test Results	35
Step 9: Clean Up	38
Next Steps	39
Project Templates	40
How Do I Choose the Right Template?	40
Web Application	42
Web Service	42
Amazon Alexa Skill	43
Working with Projects	45
Create a Project	46
Create a Project in AWS CodeStar	46
Use an IDE with AWS CodeStar	50
Use Eclipse with AWS CodeStar	51
Use Visual Studio with AWS CodeStar	55
Customize a Project Dashboard	55
Add, Remove, or Move Tiles on Your Dashboard	56
Add a Project Extension to Your Dashboard	56
Customize the Team Wiki Tile	57
Change Project Resources	57
Supported Resource Changes	57
Add a Stage to AWS CodePipeline	58
Change AWS Elastic Beanstalk Environment Settings	59
Change an AWS Lambda Project	59
Working with Project Tags	59
Add a Tag to a Project	59

Remove a Tag from a Project	59
Get a List of Tags for a Project	60
Delete a Project	60
Delete a Project in AWS CodeStar Using the Console	61
Delete a Project in AWS CodeStar Using the AWS CLI	61
Working with Teams	62
Add Team Members to a Project	63
Add a Team Member Using the AWS CodeStar Console	64
Add and View Team Members Using the AWS CLI	66
Manage Team Permissions	67
Manage Team Permissions Using the AWS CodeStar Console	67
Manage Team Permissions Using the AWS CLI	69
Remove Team Members from a Project	70
Remove Team Members Using the Console	70
Remove Team Members Using the AWS CLI	70
Working with Your AWS CodeStar User Profile	72
Manage Display Information	72
Manage Your User Profile Using the AWS CodeStar Console	73
Manage User Profiles Using the AWS CLI	74
Add a Public Key to Your User Profile	76
Manage Your Public Key Using the AWS CodeStar Console	76
Manage Your Public Key Using the AWS CLI	78
Connect to Amazon EC2 Instance with Your Private Key	78
Access Permissions Reference	80
AWS CodeStar Project-Level Policies and Permissions	81
AWS CodeStar Owner Role Policy	81
AWS CodeStar Contributor Role Policy	82
AWS CodeStar Viewer Role Policy	82
AWS CodeStar Service Role Policy and Permissions	83
Attach a Policy to an IAM User	85
Action and Resource Syntax	86
Resource Scoping in AWS CodeStar	87
Projects	87
Resources	87
Teams	88
Users	88
Logging API Calls with CloudTrail	90
AWS CodeStar Information in CloudTrail	90
Understanding AWS CodeStar Log File Entries	91
Limits	93
Troubleshooting AWS CodeStar	95
Project creation failure: A project was not created	95
Project creation: I see an error when I try to edit Amazon EC2 configuration when creating a project	96
Project deletion: An AWS CodeStar project was deleted, but resources still exist	96
Team management failure: An IAM user could not be added to a team in an AWS CodeStar project	97
Service role issue: The service role could not be created	98
Service role issue: The service role is not valid or missing	98
Project role issue: AWS Elastic Beanstalk health status checks fail for instances in an AWS CodeStar project	98
Project role issue: A project role is not valid or missing	99
Project extensions: Can't connect to JIRA	99
GitHub: Can't access a repository's commit history, issues, or code	99
Document History	100
AWS Glossary	101

What Is AWS CodeStar?

AWS CodeStar is a cloud-based service for creating, managing, and working with software development projects on AWS. You can quickly develop, build, and deploy applications on AWS with an AWS CodeStar project. An AWS CodeStar project creates and integrates AWS services for your project development toolchain. Depending on your choice of AWS CodeStar project template, that toolchain might include source control, build, deployment, virtual servers or serverless resources, and more. AWS CodeStar also manages the permissions required for project users (called team members). By adding users as team members to an AWS CodeStar project, project owners can quickly and simply grant each team member role-appropriate access to a project and its resources.

The following video provides a brief introduction to AWS CodeStar: [An Introduction to AWS CodeStar](#)

Topics

- [What Can I Do with AWS CodeStar? \(p. 1\)](#)
- [A Quick Look at AWS CodeStar \(p. 2\)](#)
- [How Do I Get Started with AWS CodeStar? \(p. 2\)](#)

What Can I Do with AWS CodeStar?

You can use AWS CodeStar to help you set up your application development in the cloud and manage your development from a single, centralized dashboard. Specifically, you can:

- **Start new software projects on AWS in minutes using templates for web applications, web services and more:** AWS CodeStar includes AWS CodeStar project templates for various project types and programming languages. Because AWS CodeStar takes care of the setup, all your project resources are configured to work together.
- **Manage project access for your team:** AWS CodeStar provides a central console where you can assign project team members the roles they need to access tools and resources. These permissions are applied automatically across all AWS services used in your project, so you don't need to create or manage complex IAM policies.
- **Visualize, operate, and collaborate on your projects in one place:** AWS CodeStar includes a project dashboard that provides an overall view of the project, its toolchain, and important events. You can monitor the latest project activity, like recent code commits, and track the status of your code changes, build results, and deployments, all from the same web page. You can monitor what's going on in the project from a single dashboard and drill into problems to investigate.
- **Iterate quickly with all the tools you need:** AWS CodeStar includes an integrated development toolchain for your project. Team members push code, and changes are automatically deployed.

Integration with issue tracking allows team members to keep track of what needs to be done next. You and your team can work together more quickly and efficiently across all phases of code delivery.

A Quick Look at AWS CodeStar

The following shows an example dashboard for a software project in AWS CodeStar.



In this example, a project pipeline in AWS CodePipeline ensures that each code push made to an AWS CodeCommit repository is built and deployed to an Amazon EC2 instance using AWS CodeDeploy. The most recent commit messages, deployment statistics, and deployment status are displayed. You can drill down on the specifics of any commits, builds, or deployments.

How Do I Get Started with AWS CodeStar?

To get started with AWS CodeStar:

1. **Prepare** to use AWS CodeStar by following the steps in [Setting Up AWS CodeStar \(p. 3\)](#).
2. **Experiment** with AWS CodeStar by following the steps in the [Getting Started with AWS CodeStar \(p. 5\)](#) walkthrough.
3. **Share** your project with other developers by following the steps in [Add Team Members to an AWS CodeStar Project \(p. 63\)](#).
4. **Expand** your AWS CodeStar project by adding integration with your favorite IDE by following the steps in [Use an IDE with AWS CodeStar \(p. 50\)](#).

Setting Up AWS CodeStar

Before you can start using AWS CodeStar, you must complete the following steps. The account you use to sign in to AWS must be configured to allow the following actions:

Topics

- [Step 1: Create an AWS Account \(p. 3\)](#)
- [Step 2: Create the AWS CodeStar Service Role \(p. 3\)](#)
- [Step 3: Create or Use an IAM User \(p. 4\)](#)
- [Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects \(p. 4\)](#)
- [Step 5: Open the AWS CodeStar Console \(p. 4\)](#)
- [Next Steps \(p. 4\)](#)

Step 1: Create an AWS Account

Create an AWS account by going to <https://aws.amazon.com/> and choosing **Sign Up**.

Step 2: Create the AWS CodeStar Service Role

AWS CodeStar requires the creation of a [service role \(p. 83\)](#) in order to create and manage AWS resources and IAM permissions. You only need to create the service role once.

Important

You must be signed in as an IAM administrative user (or root account) in order to create this service role. For more information about administrative users, see [Creating Your First IAM User and Group](#).

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
2. Choose **Start project**. (If you do not see **Start project** but instead are directed to the projects list page, the service role has been created. You can jump ahead to [Step 3: Create or Use an IAM User \(p. 4\)](#).)
3. In the **Create service role** dialog, choose **Yes, create role**.
4. Exit project creation. You'll come back to this later.

Step 3: Create or Use an IAM User

To use AWS CodeStar, create an [IAM user](#) (or use an existing one in your AWS account), and then sign in to the console with that IAM user. You need an AWS access key ID and an AWS secret access key associated with your IAM user.

Important

To create IAM users, you must be logged in as an administrative user. AWS CodeStar does not support federated users. Using AWS CodeStar with a root account is not recommended.

After you have an IAM user, do one of the following:

- If you want to create projects in AWS CodeStar, apply the **AWSCodeStarFullAccess** managed policy to your IAM user. For more information, see [Working with Managed Policies](#).

Alternatively, you can create projects if you have an IAM administrative user with full permissions for all AWS services.

- If your IAM user has already been added to one or more AWS CodeStar projects, it already has the policies and permissions required to access the service and resources for the projects you belong to. To set up your local computer for working with AWS CodeStar projects, follow the steps in [Getting Started \(p. 15\)](#). You can also [sign in to the AWS CodeStar console](#) and configure your user profile. For more information, see [Manage Display Information for Your AWS CodeStar User Profile \(p. 72\)](#) and [Add a Public Key to Your AWS CodeStar User Profile \(p. 76\)](#).

Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects

Many AWS CodeStar projects use AWS CodeDeploy or AWS Elastic Beanstalk to deploy code to Amazon Elastic Compute Cloud (Amazon EC2) instances. To access Amazon EC2 instances associated with your project, create an Amazon EC2 key pair for your IAM user. Your IAM user must have permissions to create and manage Amazon EC2 keys (for example, permission to take the actions `ec2:CreateKeyPair` and `ec2:ImportKeyPair`). For more information, see [Amazon EC2 Key Pairs](#).

Step 5: Open the AWS CodeStar Console

Sign in to the AWS Management Console, and then open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Next Steps

Congratulations, you have completed the setup! To start working with AWS CodeStar, see [Getting Started with AWS CodeStar \(p. 5\)](#).

Getting Started with AWS CodeStar

In this walkthrough, you'll use AWS CodeStar to create a web application. This project includes sample code in a source repository, a continuous deployment toolchain, and a project dashboard where you can view and monitor your project.

By following the steps, you will:

- Create a project in AWS CodeStar.
- Explore the project.
- Commit a code change.
- See your code change deployed automatically.
- Add other people to work on your project.
- Clean up project resources when they're no longer needed.

Note

If you haven't already, complete the steps in [Setting Up AWS CodeStar \(p. 3\)](#) before you start this walkthrough, including [Step 2: Create the AWS CodeStar Service Role \(p. 3\)](#) while logged in with an account that is an administrative user in IAM. To create a project, you must sign in to the AWS Management Console using an IAM user that has the **AWSCodeStarFullAccess** policy.

Topics

- [Step 1: Create an AWS CodeStar Project \(p. 6\)](#)
- [Step 2: Add Display Information for Your AWS CodeStar User Profile \(p. 10\)](#)
- [Step 3: View Your Project \(p. 11\)](#)
- [Step 4: Customize the Team Wiki Tile and the Project Dashboard \(p. 14\)](#)
- [Step 5: Commit a Change \(p. 15\)](#)
- [Step 6: Add More Team Members \(p. 17\)](#)
- [Step 7: Clean Up \(p. 20\)](#)
- [Next Steps \(p. 20\)](#)

- [Tutorial: Creating and Managing a Serverless Project in AWS CodeStar \(p. 20\)](#)

Step 1: Create an AWS CodeStar Project

In this step, you'll create a JavaScript (Node.js) software development project for a web application. You'll use an AWS CodeStar project template to create the project.

Note

This walkthrough uses an AWS CodeStar project template with the following options:

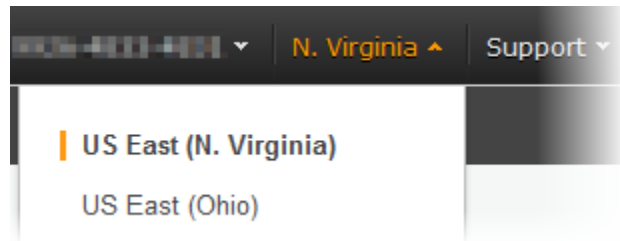
- **Application category:** Web application
- **Programming language:** Node.js
- **AWS Service:** Amazon EC2

If you choose other options, your experience might not match what's documented in this walkthrough.

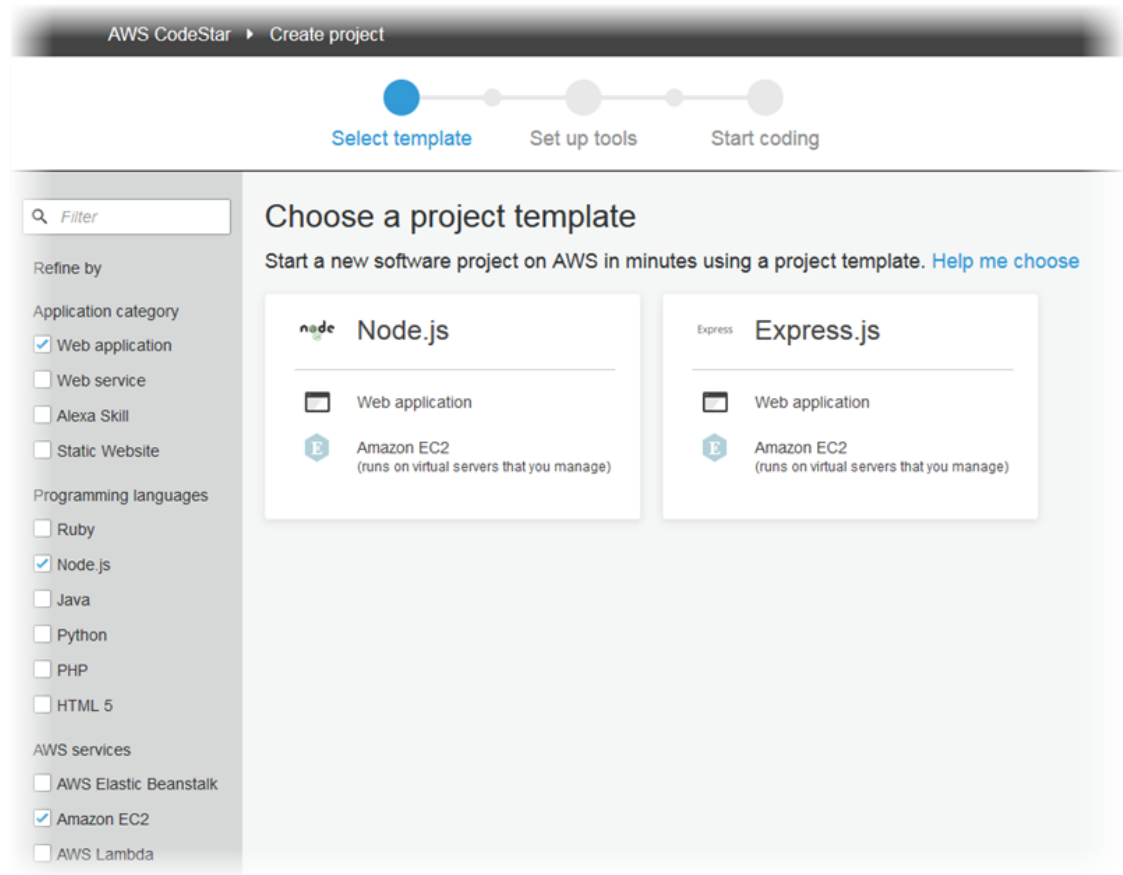
To create a project in AWS CodeStar

1. Sign in to the AWS Management Console, and then open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Make sure that you are signed in to the AWS region where you want to create the project and its resources. For example, to create a project in US East (Ohio), make sure you have that region selected. For information about AWS regions where AWS CodeStar is available, see [Regions and Endpoints](#) in the *AWS General Reference*.



2. On the **AWS CodeStar** page, choose **Create a new project**. (If you are the first user to create a project, choose **Start a project**.)
3. On the **Choose a project template** page, choose the project type from the list of AWS CodeStar project templates. You can use the filter bar to narrow your choices. For example, for a web application project written in Node.js that will be deployed to Amazon EC2 instances, select the **Web application**, **Node.js**, and **Amazon EC2** check boxes. Then choose from the templates available for that set of options.



For more information, see [AWS CodeStar Project Templates \(p. 40\)](#).

4. In **Project name**, type a name for the project, such as *My First Project*. The ID for the project is derived from this project name, but is limited to 15 characters and allowed characters.

For example, the default ID for a project named *My First Project* is *my-first-projec*. This project ID is the basis for the names of all resources associated with the project. For example, AWS CodeStar uses this project ID as part of the URL for your code repository as well as the names of related security access roles and policies in IAM. After the project is created, the project ID cannot be changed, so make sure you are okay with this project ID. To edit the project ID before you create the project, choose **Edit**.

For information about the limits on project names and project IDs, see [Limits in AWS CodeStar \(p. 93\)](#).

Note

Project IDs must be unique for your AWS account within an AWS region.

Project details

Project name
My First Project

Project ID ⓘ Edit
my-first-project

Which repository do you want to use?
AWS CodeStar will store the project's source code with the service you choose here.

AWS CodeCommit
Highly available Git source control from AWS. Includes encryption, IAM integration, and more.

GitHub
Creates a GitHub source repository for this project. Requires an existing GitHub account.

Repository name*
My-First-Project

Previous Next

5. Choose the repository provider to store this project's source code with: **AWS CodeCommit** or **GitHub**.
6. If you chose **AWS CodeCommit**, for **Repository name**, accept the default AWS CodeCommit repository name that AWS CodeStar suggests, or type a different AWS CodeCommit repository name of your choice. Then skip ahead to step 8 in this procedure.
7. If you chose **GitHub**, then choose **Connect with GitHub**.

1. If the **Sign in to GitHub** page is displayed, type your GitHub username or email address and password, and then choose **Sign in**.

Note

To complete this page, you must have a GitHub account. For more information, see [Join GitHub](#) on the GitHub website.

2. If the **Two-factor authentication** page displays, for **Authentication code**, type the code that GitHub sends you. Then choose **Verify**.
3. On the **Authorize AWS CodeStar** page, choose **Authorize**.

Note

When you choose **Authorize**, you allow AWS CodeStar to create a GitHub repository for your personal GitHub account, or for any GitHub organization where you have permissions (which is marked with a green check icon in **Organization access**).

To add a GitHub organization to the **Organization access** list, ask one of the organization's owners to invite you to the organization by following the instructions in [Inviting users to join your organization](#) on the GitHub Help website. After you join the organization, refresh the **Authorize AWS CodeStar** page to see the organization in the list.

To get permissions to authorize a GitHub organization that is in the list but does not have a green check icon, choose **Grant**. If you see **Request** instead, choose it, and then ask one of the organization's owners to allow AWS CodeStar to create a GitHub repository in the organization by following the instructions in [Approving OAuth Apps](#)

for [your organization](#) on the GitHub Help website. After the owner does this, refresh the **Authorize AWS CodeStar** page to see the **Grant** button.

4. For **Owner**, choose the GitHub organization or your personal GitHub account that you want AWS CodeStar to create the GitHub repository for.
5. For **Repository name**, accept the default GitHub repository name that AWS CodeStar suggests, or type a different GitHub repository name of your choice.
6. Choose **Public repository** or **Private repository** to make the GitHub repository public or private.

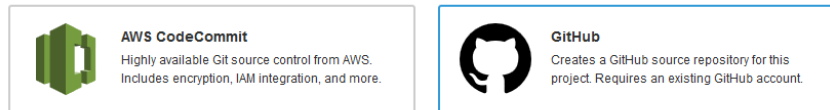
Note

Depending on your GitHub account type, GitHub may not allow you to create a private repository. For more information, see [GitHub Pricing](#) on the GitHub website.

7. For **Repository description**, provide an optional description for the GitHub repository.

Which repository do you want to use?

AWS CodeStar will store the project's source code with the service you choose here.



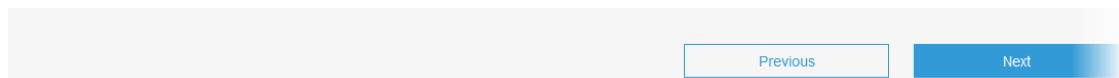
Connect to GitHub

You must give AWS CodeStar permission to create a repository in your GitHub account.

✓ AWS CodeStar is now connected to GitHub

Owner: / Repository name:

Repository description (optional):



8. Choose **Next**.
9. Review the resources and configuration details. Choose **Edit Amazon EC2 Configuration** (where available) if your project will deploy to Amazon EC2 instances and you want to make changes. For example, you can choose from available instance types for your project.

Note

Different Amazon EC2 instance types provide different levels of computing power and might have different associated costs. For more information, see [Amazon EC2 Instance Types](#) and [Amazon EC2 Pricing](#).

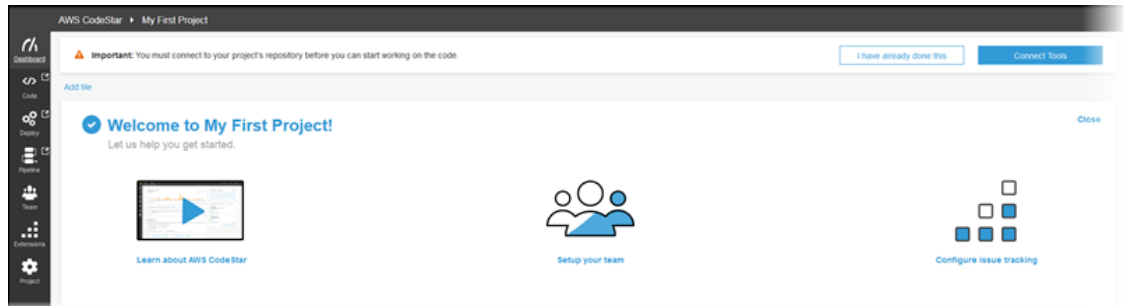
If you have more than one virtual private cloud (VPC) or multiple subnets created in Amazon Virtual Private Cloud, you can also choose the VPC and subnet to use. However, if you choose an Amazon EC2 instance type that is not supported on dedicated instances, you cannot choose a VPC whose instance tenancy is set to **Dedicated**.

For more information, see [What Is Amazon VPC?](#) and [Dedicated Instance Basics](#).

10. Leave the **AWS CodeStar would like permission to administer AWS resources on your behalf** check box selected. If this box is not selected, you will not be able to create a project. For more information about the service role, the policy, and its permissions, see [AWS CodeStar Service Role Policy and Permissions](#) (p. 83).

Choose **Next** or **Create project**. (The displayed choice depends on your project template.)

11. In **Choose an Amazon EC2 Key Pair**, choose the Amazon EC2 key pair you created in [Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects \(p. 4\)](#) in *Setting Up*. Select **I acknowledge that I have access to the private key file for this key pair**, and then choose **Create project**.
12. It might take a few minutes to create the project (including the repository). After your project has a repository, you can use the **Choose how you want to edit your project code** page to configure access to it, or you can choose **Skip** and configure access later. After your project has been created, you will see a **Welcome** tile that contains useful links. You can use these links to optionally configure other items, such as your [user profile in AWS CodeStar \(p. 72\)](#).



Step 2: Add Display Information for Your AWS CodeStar User Profile

When you create a project, you're added to the project team as an owner. If this is the first time you've used AWS CodeStar, you'll be asked to provide additional information, such as:

- The display name to show to other users.
- The email address to show to other users.

This information is used in your AWS CodeStar user profile. User profiles are not project-specific, but are limited to an AWS region. If you belong to projects in more than one region, you'll have to create a user profile in each region. Each regional profile can contain different information, if you prefer.

Provide a user name and email address when prompted, and then choose **Next**.

AWS CodeStar > Create project

Select template Set up tools Start coding Provisioning 0% complete

It looks like this is your first time here.
Set up your AWS CodeStar user settings

IAM user name ⓘ

Display Name
Mary Major

Email
mary.major@example.com

Previous Next

Note

This user name and email address is used in your AWS CodeStar user profile. If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resource providers may use separate user profiles, which may have different user names and email addresses. For more information, see the resource provider's documentation.

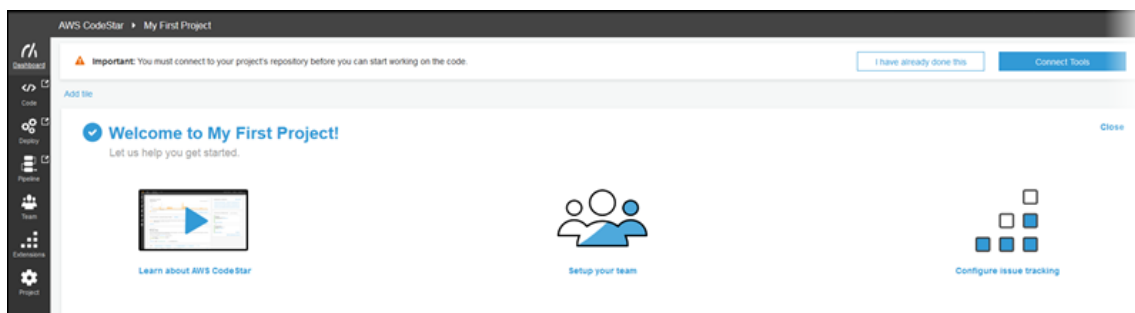
Step 3: View Your Project

Your AWS CodeStar project dashboard is where you and your team view the status of your project resources, including the latest commits to your project, the state of your continuous delivery pipeline, and the performance of your instances. This information is displayed on tiles that are dedicated to a particular resource. To see more information about any of these resources, choose the details link on the tile. The console for that AWS service will open on the details page for that resource.

You can change where each tile appears on your dashboard by dragging and dropping it to a new location. You can also use the ellipsis menu on each tile to remove that tile from the display. To add a tile, choose **Add tile** in the dashboard, and choose any tile that is not already present.

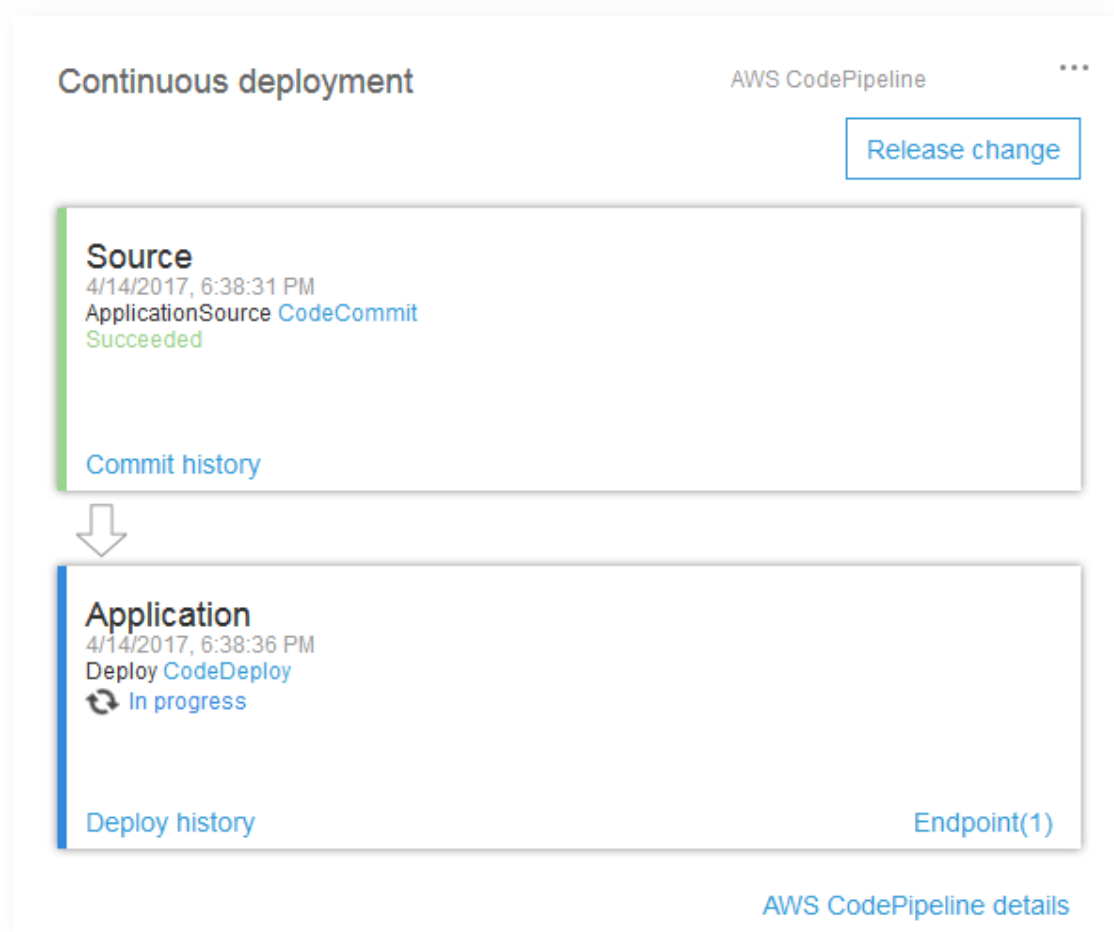
In your new project, you'll see the following tiles:

- The **Welcome** tile contains links to actions you might want to perform. Unlike other tiles, you cannot move this tile to another location, or add it back after closing it.

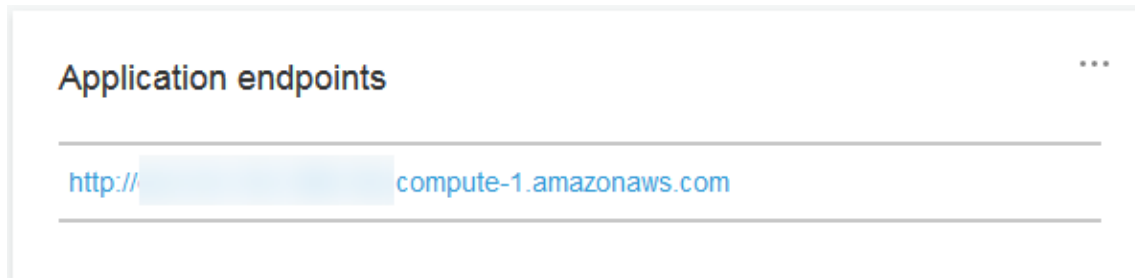


- The **Continuous deployment** tile displays a summary view of the continuous delivery pipeline for your project. The pipeline deploys the web application code when there is a change in your repository. Because your project is new, the pipeline immediately starts deploying the sample code. You can see the processing and completion of each stage as your web application is deployed. For a deployment stage, choose **Endpoint** to view a link to that endpoint (if you chose the project template suggested at the start of this walkthrough, it's an Amazon EC2 instance where your sample web application is deployed and running).

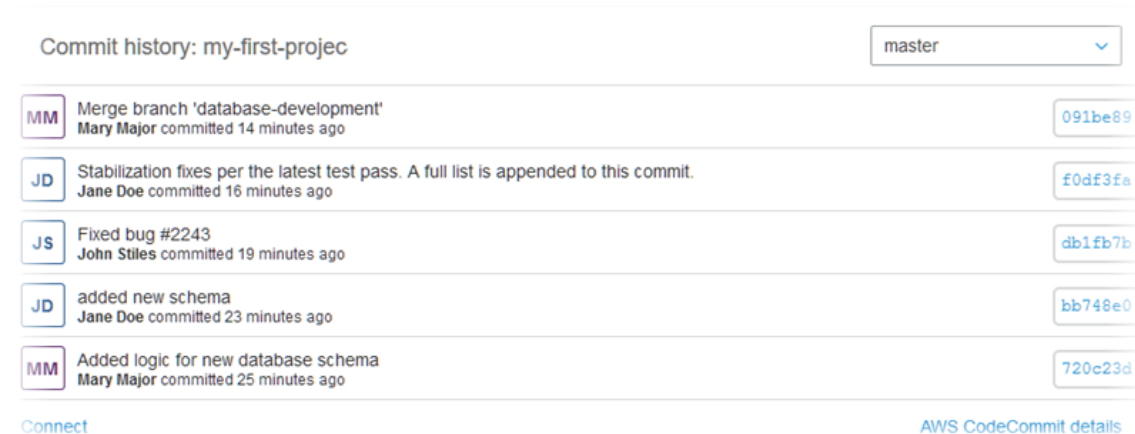
You can also see if a stage has a problem or requires approval. To see details about the state of the pipeline, its stages, and its actions, or to make changes such as adding or editing a stage, choose **AWS CodePipeline details**.



- The **Application endpoints** tile displays links to the endpoints where you can view your software. This is the same link you saw in the **Continuous deployment** tile. Choose the link to view your application or service.



- The **Commit history** tile displays the recent commit history of the repository. When you first create a project, the most recent commit is the one made by AWS CodeStar. This commit started running the sample code through the pipeline. When you make another commit, that will appear in the history, too. That code change will start running through the pipeline automatically. To view the commits of a different branch, use the branch selector button. To view all commits or other details about the commits or the repository, choose **AWS CodeCommit details** (if the code is stored in AWS CodeCommit) or **Open in GitHub** (if the source code is stored in GitHub).



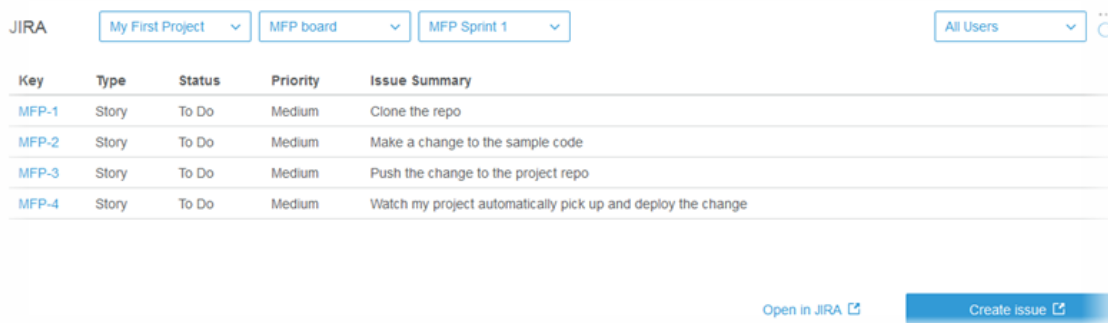
- The **Application activity** tile displays Amazon CloudWatch metrics for your project. For example, it displays the CPU utilization of any Amazon EC2 instances deployed to by AWS Elastic Beanstalk or AWS CodeDeploy resources in your pipeline. In projects that use AWS Lambda, it displays invocation and error metrics for the Lambda function. This information is displayed by the hour. If you used the suggested AWS CodeStar project template for this walkthrough, you should see a noticeable spike in activity as your application is first deployed to those instances. You can refresh monitoring to see changes in your instance health, which can help you identify problems or the need for more resources.



Note

If your AWS CodeStar project includes more than one metric, you can filter the display by choosing a particular metric in the tile.

- The **JIRA** tile is for integrating your AWS CodeStar project with an existing Atlassian JIRA project. Configuring this tile will enable you and your project team to track JIRA issues from the project dashboard. To configure this tile, choose **Connect** and follow the instructions.



- There is also a **Team wiki tile**. You can customize the contents of this tile to store team notes, link to useful resources for your team project, provide samples, and so on. You'll customize this tile in the next step.

Step 4: Customize the Team Wiki Tile and the Project Dashboard

Each AWS CodeStar project includes a customizable team wiki tile that can be used for any purpose (for example, adding links to team resources or showing code snippets for a preferred development style). This tile supports both plain text and formatted content. In this step, you'll customize this tile to include a link to the AWS DevOps blog.

To customize the team wiki tile

1. In the project dashboard, on the team wiki tile, choose the ellipsis menu, and then choose **Edit**.
2. In **Widget title**, type **Team links**. In **Markdown content**, add an item to the list and paste the following:

```
[AWS DevOps Blog](https://aws.amazon.com/blogs/devops/)
```

Choose **Save**.

3. Choose the link on the tile to test it.

To customize your dashboard appearance

1. Choose one of the tiles on the dashboard. Drag and drop it to a new position. You can rearrange dashboard tiles to help ensure the information most important to you is in the most visible positions.
2. To remove a tile, choose the ellipsis menu (...) on that tile, and then choose **Remove from Dashboard**.
3. To add a tile, choose **Add tile** at the top of the dashboard, and then choose the tile to add. You can only have one of each kind of tile on your dashboard.

Step 5: Commit a Change

First, take a look at the sample code that was included in your project, and see what the application looks like. On the **Application endpoints** tile, choose the link to your endpoint. Your sample web application is displayed in a new window or browser tab. This is the project sample code that was deployed.

If the project's code is stored in an AWS CodeCommit repository, and you'd like to look at the code itself, in the navigation bar, choose **Code**. Your project's repository opens in a new tab or window. Read the contents of the repository's readme file (`README.md`), and browse the content of those files.

In this step, you will make a change to the code and then push that change to your repository. You can do this in one of several ways:

- If the project's code is stored in an AWS CodeCommit repository, and you have Visual Studio or Eclipse installed, you can use the AWS Toolkit for Visual Studio or AWS Toolkit for Eclipse to more easily connect to the code. For more information, see [Use an IDE with AWS CodeStar \(p. 50\)](#). If you don't have Visual Studio or Eclipse installed, then install a Git client, and follow the instructions later in this step.
- If the project's code is stored in a GitHub repository, you can use your IDE's tools for connecting to GitHub. For example:
 - For Visual Studio, you can use a tools such as the GitHub Extension for Visual Studio. For more information, see the [Overview](#) page on the GitHub Extension for Visual Studio website and [Getting Started with GitHub for Visual Studio](#) on the GitHub website.
 - For Eclipse, you can use a tool such as EGit for Eclipse. For more information, see the [EGit](#) website and [EGit Documentation](#) on the EGit website.
 - For other IDEs, consult your IDE's documentation.
- For other types of code repositories, see the repository provider's documentation.

The following instructions show how to make a basic change to the sample.

To set up your computer to commit changes

Note

This procedure assumes that your project's code is stored in an AWS CodeCommit repository. For other types of code repositories, see the repository provider's documentation, and then skip ahead to the next procedure, "[To clone the project repository and make a change \(p. 16\)](#)."

If the code is stored in AWS CodeCommit, and you are already using AWS CodeCommit, you don't need more configuration. Skip ahead to the next procedure, "[To clone the project repository and make a change \(p. 16\)](#)."

1. [Install Git](#) on your local computer.
2. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.

Sign in as the IAM user who will use Git credentials for connections to your AWS CodeStar project repository in AWS CodeCommit.

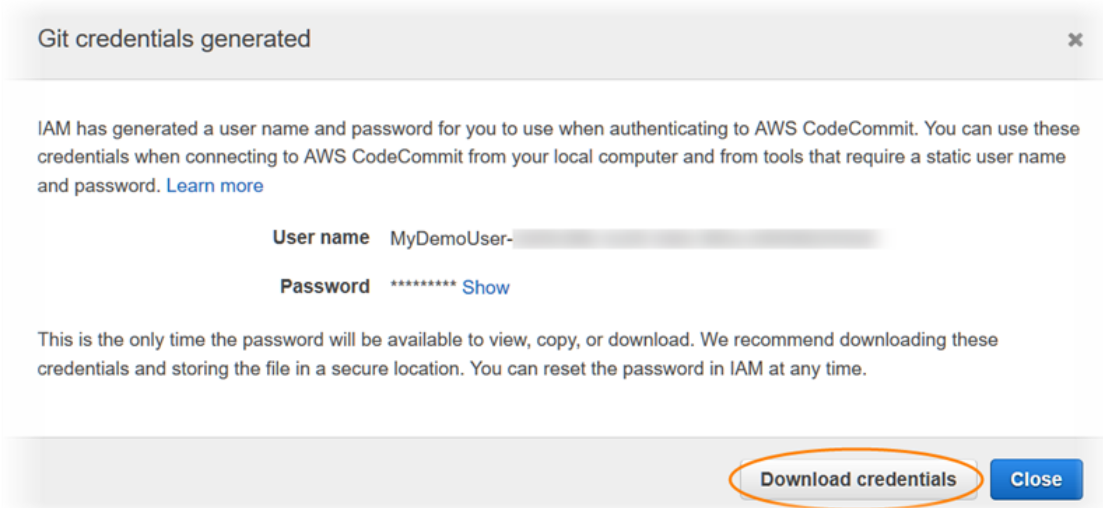
3. In the IAM console, in the navigation pane, choose **Users**, and from the list of users, choose your IAM user.
4. On the user details page, choose the **Security Credentials** tab, and in **HTTPS Git credentials for AWS CodeCommit**, choose **Generate**.



Note

You cannot choose your own user name or password for Git credentials. For more information, see [Use Git Credentials and HTTPS with AWS CodeCommit](#).

5. Copy the user name and password that IAM generated for you. You can choose **Show** and then copy and paste this information into a secure file on your local computer, or you can choose **Download credentials** to download this information as a .CSV file. You will need this information to connect to AWS CodeCommit.



After you have saved your credentials, choose **Close**.

Important

This is your only chance to save the user name and password. If you do not save them, you can copy the user name from the IAM console, but you cannot look up the password. You must reset the password and then save it.

To clone the project repository and make a change

Note

This procedure shows how to clone the project's code repository to your computer, make a change to the project's `index.html` file, and then push your change to the remote repository. This procedure assumes that your project's code is stored in an AWS CodeCommit repository and that you're using a Git client from the command line. For other types of code repositories

or tools, see the provider's documentation for how to clone the repository, change the file, and then push the code.

1. With your project open in the AWS CodeStar console, on the navigation bar, choose the **Project** gear icon, and then choose the **Connect tools** button. In the drop-down list next to **Clone repository URL**, choose the protocol for the connection type you have set up for AWS CodeCommit, and then copy the link. For example, if you followed the steps in the previous procedure to set up Git credentials for AWS CodeCommit, choose **HTTPS**.
2. On your local computer, open a terminal or command line window and change directories to a temporary directory. Run the **git clone** command to clone the repository to your computer. Paste the link you copied. For example, for AWS CodeCommit using HTTPS:

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

The first time you connect, you will be prompted for the user name and password for the repository. For AWS CodeCommit, type the Git credentials user name and password you downloaded in the previous procedure.

3. Navigate to the clone directory on your computer and browse the contents.
4. Open the `index.html` file in an editor and make a change to the file. For example, you could change the header text from `<H1>Congratulations!</H1>` to the following:

```
<H1>I did it!</H1>
```

Save the file.

5. At the terminal or command prompt, add your changed file, and then commit and push your change:

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

6. On your project dashboard, view the changes in progress. You'll see that the commit history for the repository is updated with your commit, including the commit message. You can also see the pipeline pick up your change to the repository and start building and deploying it. You can use the links you added to the project information tile to view your change to the web application after it is deployed.

Step 6: Add More Team Members

One of the benefits of an AWS CodeStar project is the simplified process for adding users and giving them access to project resources. Every AWS CodeStar project comes preconfigured with three different AWS CodeStar roles. Each role provides its own level of access to the project and its resources:

- **Owner:** Can add and remove team members, change the project dashboard (including adding or removing project extensions), and delete the project.
- **Contributor:** Can change the project dashboard and contribute code if the code is stored in AWS CodeCommit, but cannot add or remove team members or delete the project. This is the role you should choose for most team members in an AWS CodeStar project.
- **Viewer:** Can view the project dashboard, project code if the code is stored in AWS CodeCommit, and the state of the project, but cannot move, add, or remove tiles from the project dashboard.

Important

If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, access to those resources are controlled by the resource provider, not AWS CodeStar. For more information, see the resource provider's documentation.

Anyone who has access to an AWS CodeStar project may be able to use the AWS CodeStar console to access resources that are outside of AWS but are related to that project.

For more information about teams and project roles, see [Working with AWS CodeStar Teams \(p. 62\)](#).

To add a team member to an AWS CodeStar project (console)

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Choose the project.

2. In the navigation bar for the project, choose **Team**.
3. On the **Team members** page, choose **Add team member**.
4. In **Choose user**, do one of the following:
 - If an IAM user already exists for the person you want to add, choose the IAM user name from the list.

Note

Users who have already been added to another AWS CodeStar project will appear in the **AWS CodeStar users from other projects** list.

On the **Add team member** tab, in **Project role**, choose the AWS CodeStar role (Owner, Contributor, or Viewer) for this user. This is an AWS CodeStar project-level role that can only be changed by an owner of the project. When applied to an IAM user, the role provides all appropriate permissions required to access AWS CodeStar project resources. It applies policies required for creating and managing Git credentials for code stored in AWS CodeCommit in IAM or uploading Amazon EC2 SSH keys for the user in IAM.

Important

You cannot provide or change the display name or email information for an IAM user unless you are signed in to the console as that user. For more information, see [Manage Display Information for Your AWS CodeStar User Profile \(p. 72\)](#).

Choose **Add**.

The screenshot shows the AWS CodeStar console interface for a project named 'My First Project'. The 'Team' tab is selected, displaying the 'Project Team' section with the heading 'Manage users and permissions in your project.' Below this, the 'Add team member' tab is active, showing a form to add a new team member. The form includes a search bar with 'Jane Doe' entered, and fields for 'Display Name' (Jane Doe), 'Email Address' (jane.doe@example.com), 'Project Role' (Contributor), and 'Remote Access' (checked). An 'Add' button is at the bottom of the form. Below the form, the 'Team member list (4)' is displayed, showing a table of existing team members.

Name	Email	Role	Remote Access
MM Mary Major ([You])	mary.major@example.com	Owner	Change Public SSH key Remove Edit
JD John Doe ()	john.doe@example.com	Contributor	Granted Remove Edit
JS John Stiles ()	john.stiles@example.com	Contributor	Granted Remove Edit

- If an IAM user does not exist for the person you want to add to the project, choose **Create new IAM user**. Fill in the IAM user name, AWS CodeStar display name, email address, and project role you want to apply to this new user, and choose **Create**.

Create IAM user ✕

Create an IAM user to add to your project.

IAM user name

Display Name

Email

Project role **Remote access**

☐

You will be redirected to the IAM console to confirm user creation. Choose **Create user**, save the password information for that new user, and then choose **Close** to return to the AWS CodeStar console. The user will be automatically added to the project with the role you chose.

Note

For ease of management, at least one user should have the **Owner** role for the project.

5. Send the new team member the following information:
 - Connection information for your AWS CodeStar project.
 - If the source code is stored in AWS CodeCommit, [Instructions for setting up access with Git credentials](#) to the AWS CodeCommit repository from their local computers.
 - Information about how the user can manage their display name, email address, and public Amazon EC2 SSH key, as described in [Working with Your AWS CodeStar User Profile](#) (p. 72).
 - One-time password and connection information, if the user is new to AWS and you created an IAM user for that person. The password will expire the first time the user logs on. The user must choose a new password.

Step 7: Clean Up

Congratulations! You've finished the Getting Started walkthrough for AWS CodeStar. If you don't want to continue to use this project and its resources, you should delete it so you can avoid any possible ongoing charges to your AWS account.

To delete a project in AWS CodeStar

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
2. Find the project in the list, and from the ellipsis menu, choose **Delete**.

Alternatively, open the project, and in the navigation pane, choose **Project**. On the project details page, choose **Delete project**.

3. In the box next to **Type the following project ID to confirm**, type the ID of the project, and then choose **Delete**.

Deleting a project can take several minutes. After it's deleted, the project no longer appears in the list of projects in the AWS CodeStar console.

Important

By default, when you delete a project, all resources listed under **Project resources** are deleted. If you clear the check box, the project resources will be retained. For more information, go [here \(p. 60\)](#).

If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resources are not deleted, even if the check box is selected.

Next Steps

We suggest continuing to learn about AWS CodeStar by using the following resources:

- The [Tutorial: Creating and Managing a Serverless Project in AWS CodeStar \(p. 20\)](#) uses a project that creates and deploys a web service using logic in AWS Lambda and can be called by an API in Amazon API Gateway.
- [AWS CodeStar Project Templates \(p. 40\)](#) describes other types of projects you can create.
- [Customize an AWS CodeStar Dashboard \(p. 55\)](#) provides more information about customizing your projects' dashboards, integrating with JIRA, and more.
- [Working with AWS CodeStar Teams \(p. 62\)](#) provides more information about enabling others to help you work on your projects.

Tutorial: Creating and Managing a Serverless Project in AWS CodeStar

In this tutorial, you will use AWS CodeStar to create a project that uses the AWS Serverless Application Model (AWS SAM) to create and manage AWS resources for a web service hosted in AWS Lambda.

AWS CodeStar uses AWS SAM, which relies on AWS CloudFormation, to provide a simplified way of creating and managing supported AWS resources, including Amazon API Gateway APIs, AWS Lambda functions, and Amazon DynamoDB tables. (This project does not use any Amazon DynamoDB tables.)

For more information about AWS SAM, see [AWS Serverless Application Model \(AWS SAM\)](#) on GitHub.

Prerequisite: Complete the steps in [Setting Up AWS CodeStar \(p. 3\)](#).

Note

Your AWS account may be charged for costs related to this tutorial, including costs for AWS services used by AWS CodeStar. For more information, see [AWS CodeStar Pricing](#).

Topics

- [Overview \(p. 21\)](#)
- [Step 1: Create the Project \(p. 22\)](#)
- [Step 2: Explore Project Resources \(p. 24\)](#)
- [Step 3: Test the Web Service \(p. 28\)](#)
- [Step 4: Set Up Your Local Workstation to Edit Project Code \(p. 30\)](#)
- [Step 5: Add Logic to the Web Service \(p. 32\)](#)
- [Step 6: Test the Enhanced Web Service \(p. 33\)](#)
- [Step 7: Add a Unit Test to the Web Service \(p. 34\)](#)
- [Step 8: View Unit Test Results \(p. 35\)](#)
- [Step 9: Clean Up \(p. 38\)](#)
- [Next Steps \(p. 39\)](#)

Overview

In this tutorial, you will do the following:

1. Use AWS CodeStar to create a project that uses AWS SAM to build and deploy a Python-based web service. This web service is hosted in AWS Lambda and can be accessed through Amazon API Gateway.
2. Explore the project's main resources, which include:
 - The AWS CodeCommit repository where the project's source code is stored. This source code includes the web service's logic and defines related AWS resources.
 - The AWS CodePipeline pipeline that automates the building of the source code. This pipeline uses AWS SAM to create and deploy a function to AWS Lambda, create a related API in Amazon API Gateway, and connect the API to the function.
 - The function that is deployed to AWS Lambda.
 - The API that is created in Amazon API Gateway.
3. Manually test the web service to confirm that AWS CodeStar built and deployed the web service as expected.
4. Set up your local workstation to work with the project's source code.
5. Change the project's source code using your local workstation. You add a function to the project and then push your changes to the source code, which instructs AWS CodeStar to rebuild and redeploy the web service.
6. Manually test the web service again to confirm that AWS CodeStar rebuilt and redeployed as expected.
7. Write a unit test using your local workstation to replace some of your manual testing with an automated test. Push the unit test, which instructs AWS CodeStar to rebuild and redeploy the web service and run the unit test automatically.
8. View the results of the unit tests.
9. Clean up the project. This step is to keep AWS from charging your AWS account for costs related to this tutorial.

Step 1: Create the Project

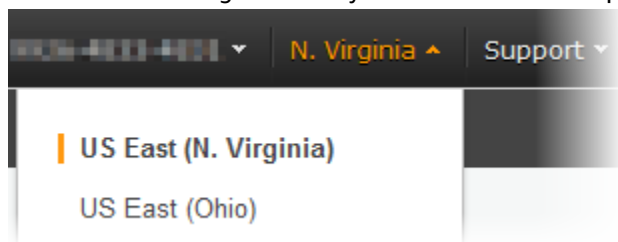
In this step, you use the AWS CodeStar console to create a project. This project uses AWS SAM to create and deploy a Python-based web service that is hosted in AWS Lambda and can be accessed through Amazon API Gateway.

1. Sign in to the AWS Management Console and open the AWS CodeStar console, at <https://console.aws.amazon.com/codestar/>.

Note

You must sign in to the AWS Management Console using credentials associated with the IAM user you created or identified in [Setting Up AWS CodeStar \(p. 3\)](#). This user must have the AWS managed policy named **AWSCodeStarFullAccess** attached.

2. Choose the AWS region where you want to create the project and its resources.



For information about AWS regions where AWS CodeStar is available, see [Regions and Endpoints](#) in the *AWS General Reference*.

3. Choose **Create a new project** (or, if **Create a new project** is not displayed, **Start a new project**).
4. On the **Choose a project template** page:
 - For **Application category**, select **Web service**.
 - For **Programming languages**, select **Python**.
 - For **AWS services**, select **AWS Lambda**.
5. Choose the box that contains your selections.

The screenshot shows the 'Choose a project template' interface in AWS CodeStar. On the left, a sidebar contains filter sections: 'Application category' with 'Web service' selected, 'Programming languages' with 'Python' selected, and 'AWS services' with 'AWS Lambda' selected. The main area on the right displays the chosen template: 'Python' with the Python logo, followed by 'Web service' with a server icon, and 'AWS Lambda (running serverless)' with the Lambda icon. Red rectangular boxes are drawn around the 'Web service' checkbox in the filters, the 'Python' checkbox, and the entire template selection card on the right.

6. For **Project name**, type a name for the project (for example, **my SAM Project**). If you type a name different from the example, be sure to use it throughout this tutorial.

For **Project ID**, AWS CodeStar chooses a related identifier for this project (for example, **my-sam-project**). If you see a different project ID, be sure to use it throughout this tutorial.

Leave **AWS CodeCommit** selected, and do not change the **Repository name** value.

7. Choose **Next**.
8. Leave the **AWS CodeStar would like permission to administer AWS resources on your behalf** box selected, and then choose **Create Project**.

If this is your first time using AWS CodeStar in this AWS region, then for **Display Name** and **Email**, type the display name and email address you want AWS CodeStar to use for your IAM user. Choose **Next**.

9. On the **Choose how you want to edit your project code** page, choose **Skip**. You set up your local workstation to edit the project's code in a later step.

10. Wait while AWS CodeStar creates the project. This might take several minutes. Do not proceed until you see **Welcome to My SAM Project!**.

⚠ Important: You must connect to your project's repository before you can start working on

Add tile



[Learn about AWS CodeStar](#)

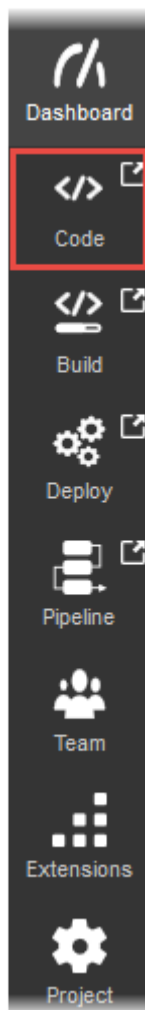
Step 2: Explore Project Resources

In this step, you explore four of the project's AWS resources to understand how the project works.

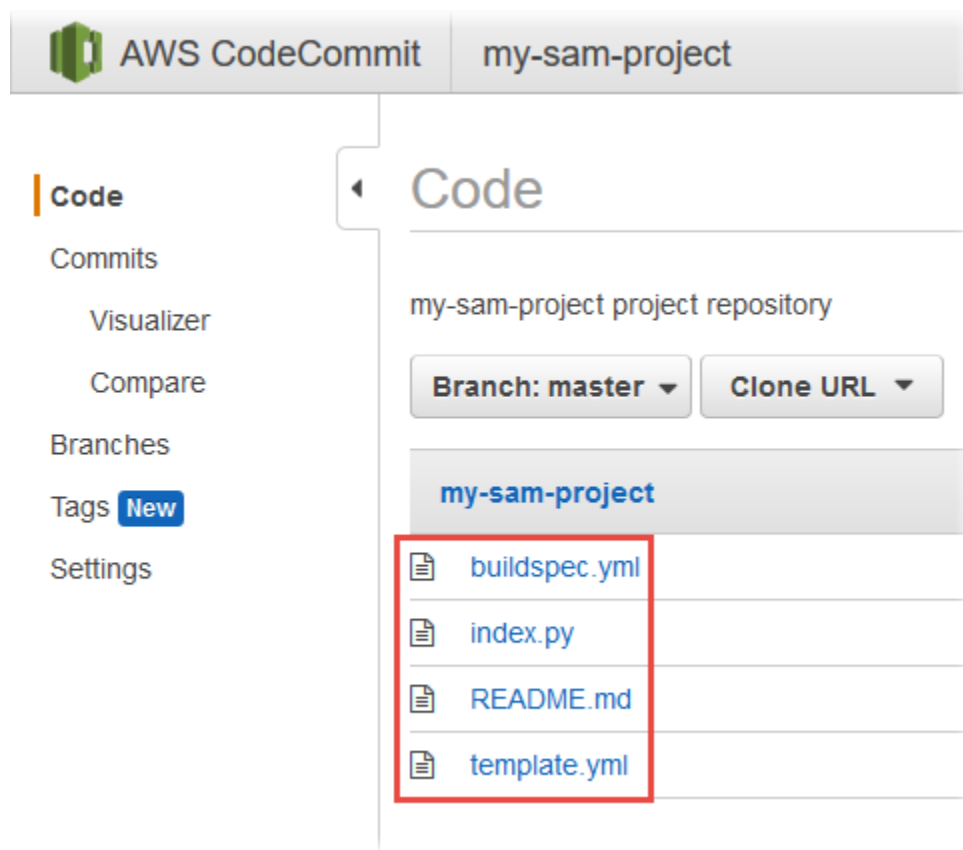
- The AWS CodeCommit repository where the project's source code is stored. AWS CodeStar gives the repository the name **my-sam-project**, where **my-sam-project** is the name of the project.
- The AWS CodePipeline pipeline that uses AWS CodeBuild and AWS SAM to automate building and deploying the web service's Lambda function and API in API Gateway. AWS CodeStar gives the pipeline the name **my-sam-project--Pipeline**, where **my-sam-project** is the ID of the project.
- The Lambda function that contains the logic of the web service. AWS CodeStar gives the function the name **awscodestar-my-sam-project-lambda-HelloWorld-*RANDOM_ID***, where:
 - **my-sam-project** is the ID of the project.
 - **HelloWorld** is the function ID as specified in the `template.yaml` file in the AWS CodeCommit repository. You explore this file later.
 - ***RANDOM_ID*** is a random ID that AWS SAM assigns to the function to help ensure uniqueness.
- The API in API Gateway that makes it easier to call the Lambda function. AWS CodeStar gives the API the name **awscodestar-my-sam-project--lambda**, where **my-sam-project** is the ID of the project.

To explore the source code repository in AWS CodeCommit

1. With your project open in the AWS CodeStar console, on the side navigation bar, choose **Code**.



2. In the AWS CodeCommit console, on the **Code** page, the source code files for the project are displayed.
 - `buildspec.yml`, which AWS CodePipeline instructs AWS CodeBuild to use during the build phase, to package the web service using AWS SAM.
 - `index.py`, which contains the logic for the Lambda function. This function simply outputs the string `Hello World`, along with a timestamp in ISO format.
 - `README.md`, which contains general information about the repository.
 - `template.yml`, which AWS SAM uses to package the web service and create the API in API Gateway.

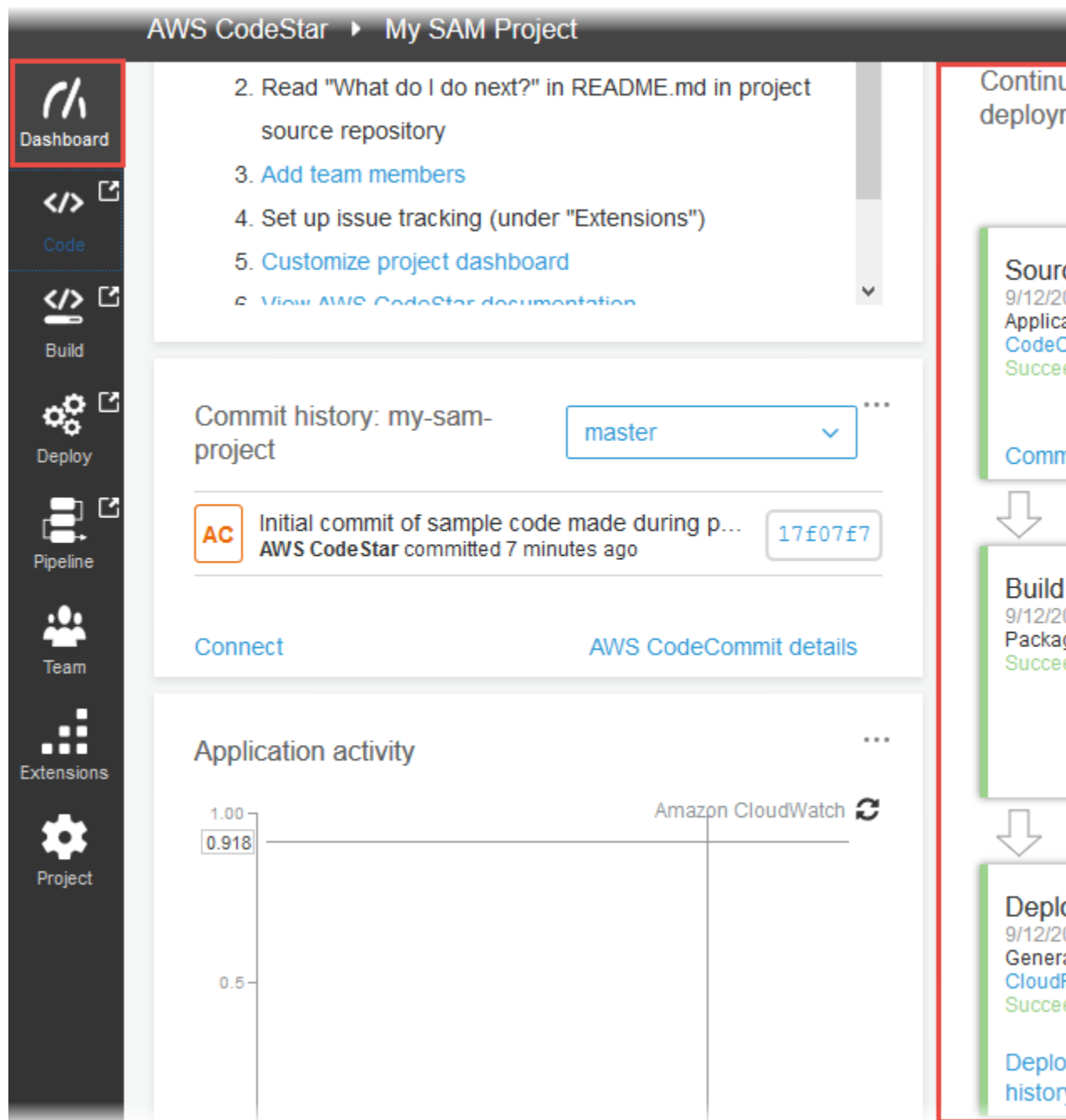


To view the contents of a file, simply choose it from the list.

For more information about using the AWS CodeCommit console, see the [AWS CodeCommit User Guide](#).

To explore the pipeline in AWS CodePipeline

1. To view information about the pipeline, with your project open in the AWS CodeStar console, on the side navigation bar, choose **Dashboard**. On the **Continuous deployment** tile, you see the pipeline contains:
 - A **Source** stage for getting the source code from AWS CodeCommit.
 - A **Build** stage for building the source code with AWS CodeBuild.
 - A **Deploy** stage for deploying the built source code and AWS resources with AWS SAM.



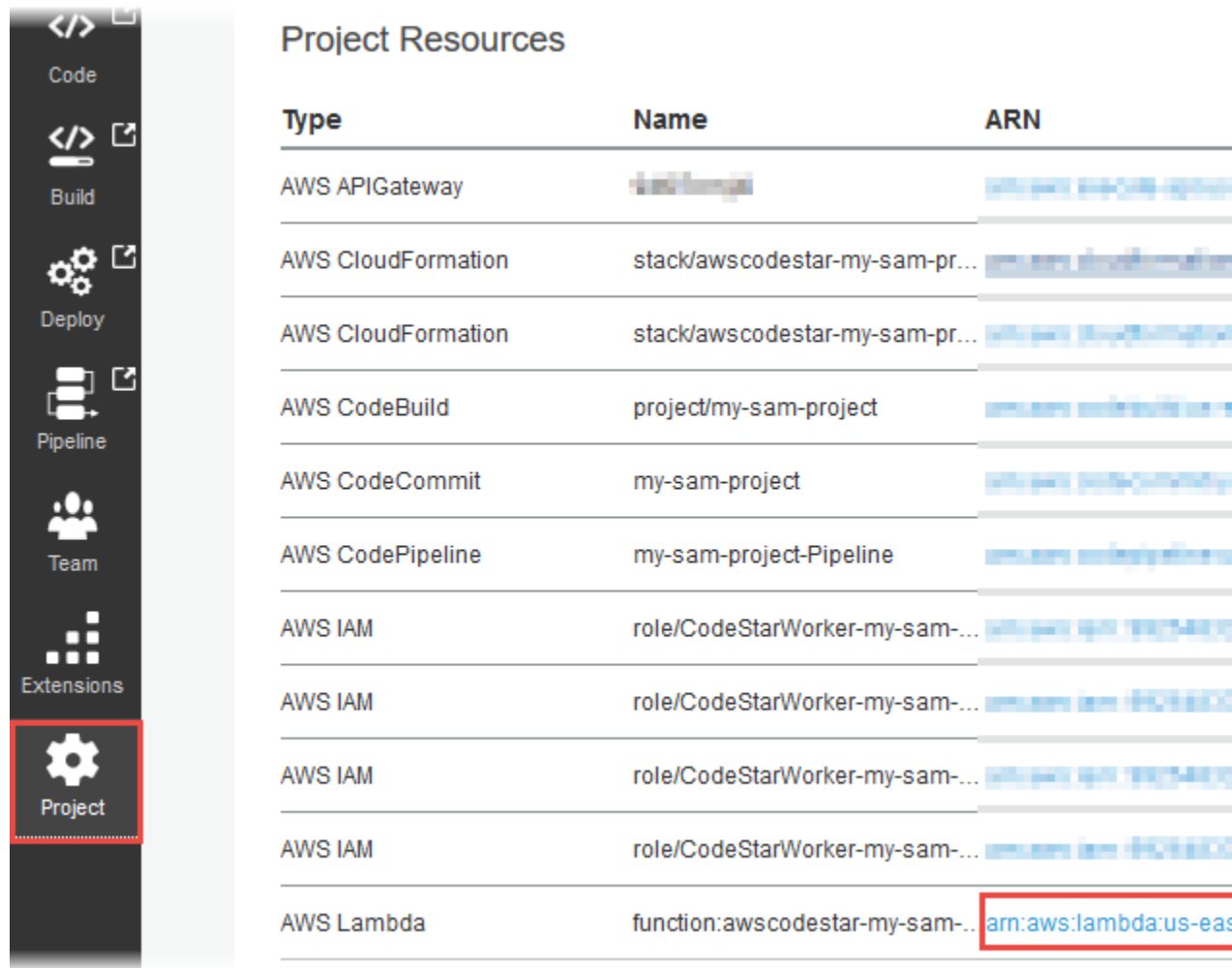
2. To view detailed information about the pipeline, on the **Continuous deployment** tile, choose the **AWS CodePipeline details** link or, on the side navigation bar, choose **Pipeline** to open the pipeline in the AWS CodePipeline console.

For information about using the AWS CodePipeline console, see the [AWS CodePipeline User Guide](#).

To explore the function in Lambda

1. With your project open in the AWS CodeStar console, on the side navigation bar, choose **Project**.

2. In the **Project Resources** list, choose the link in the **ARN** column for the Lambda function.



The screenshot shows the AWS CodeStar console interface. On the left is a dark navigation bar with icons for Code, Build, Deploy, Pipeline, Team, Extensions, and Project. The 'Project' icon, which is a gear, is highlighted with a red rectangular box. To the right of the navigation bar is the 'Project Resources' section. It contains a table with three columns: 'Type', 'Name', and 'ARN'. The table lists several AWS resources. The last row, 'AWS Lambda', has its 'ARN' value, 'arn:aws:lambda:us-east-1:123456789012:function:my-lambda-function', highlighted with a red rectangular box.

Type	Name	ARN
AWS APIGateway	api-gateway	arn:aws:apigateway:us-east-1:123456789012:api-gateway
AWS CloudFormation	stack/awscodestar-my-sam-pr...	arn:aws:cloudformation:us-east-1:123456789012:stack/awscodestar-my-sam-pr...
AWS CloudFormation	stack/awscodestar-my-sam-pr...	arn:aws:cloudformation:us-east-1:123456789012:stack/awscodestar-my-sam-pr...
AWS CodeBuild	project/my-sam-project	arn:aws:codebuild:us-east-1:123456789012:project/my-sam-project
AWS CodeCommit	my-sam-project	arn:aws:codecommit:us-east-1:123456789012:my-sam-project
AWS CodePipeline	my-sam-project-Pipeline	arn:aws:codepipeline:us-east-1:123456789012:my-sam-project-Pipeline
AWS IAM	role/CodeStarWorker-my-sam-...	arn:aws:iam:us-east-1:123456789012:role/CodeStarWorker-my-sam-...
AWS IAM	role/CodeStarWorker-my-sam-...	arn:aws:iam:us-east-1:123456789012:role/CodeStarWorker-my-sam-...
AWS IAM	role/CodeStarWorker-my-sam-...	arn:aws:iam:us-east-1:123456789012:role/CodeStarWorker-my-sam-...
AWS IAM	role/CodeStarWorker-my-sam-...	arn:aws:iam:us-east-1:123456789012:role/CodeStarWorker-my-sam-...
AWS Lambda	function:aws-codestar-my-sam-...	arn:aws:lambda:us-east-1:123456789012:function:aws-codestar-my-sam-...

The function's code is displayed in the Lambda console.

For information about using the Lambda console, see the [AWS Lambda Developer Guide](#).

To explore the API in API Gateway

1. With your project open in the AWS CodeStar console, on the side navigation bar, choose **Project**.
2. In the **Project Resources** list, choose the link in the **ARN** column for the Amazon API Gateway API.

Settings for the API are displayed in the API Gateway console.

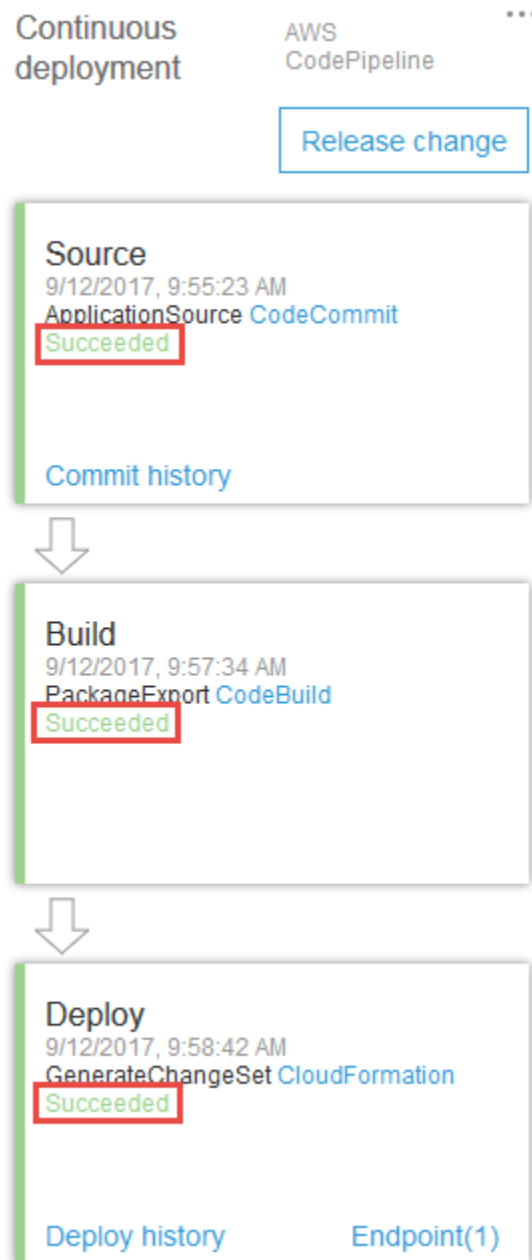
For information about using the API Gateway console, see the [API Gateway Developer Guide](#).

Step 3: Test the Web Service

In this step, you will test the web service that AWS CodeStar just built and deployed.

1. With your project still open from the previous step, on the side navigation bar, choose **Dashboard**.

2. On the **Continuous deployment** tile, make sure **Succeeded** is displayed for the **Source**, **Build**, and **Deploy** stages before you continue. This might take several minutes.



Note

If **Failed** is displayed for any of the stages, see the following for troubleshooting help:

- For the **Source** stage, see [Troubleshooting AWS CodeCommit](#) in the *AWS CodeCommit User Guide*.
- For the **Build** stage, see [Troubleshooting AWS CodeBuild](#) in the *AWS CodeBuild User Guide*.

- For the **Deploy** stage, see [Troubleshooting AWS CloudFormation](#) in the *AWS CloudFormation User Guide*.
 - For other issues, see [Troubleshooting AWS CodeStar \(p. 95\)](#).
3. Choose the link on the **Application endpoints** tile. It should look like `https://API_ID.execute-api.REGION_ID.amazonaws.com/Prod/`, where:
 - **API_ID** represents the ID that API Gateway assigned to the API.
 - **REGION_ID** is the ID of the related AWS region.
 - **Prod** is the name of the related API deployment stage in API Gateway.

On the new tab that opens in your web browser, the web service displays the following response output:

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

Step 4: Set Up Your Local Workstation to Edit Project Code

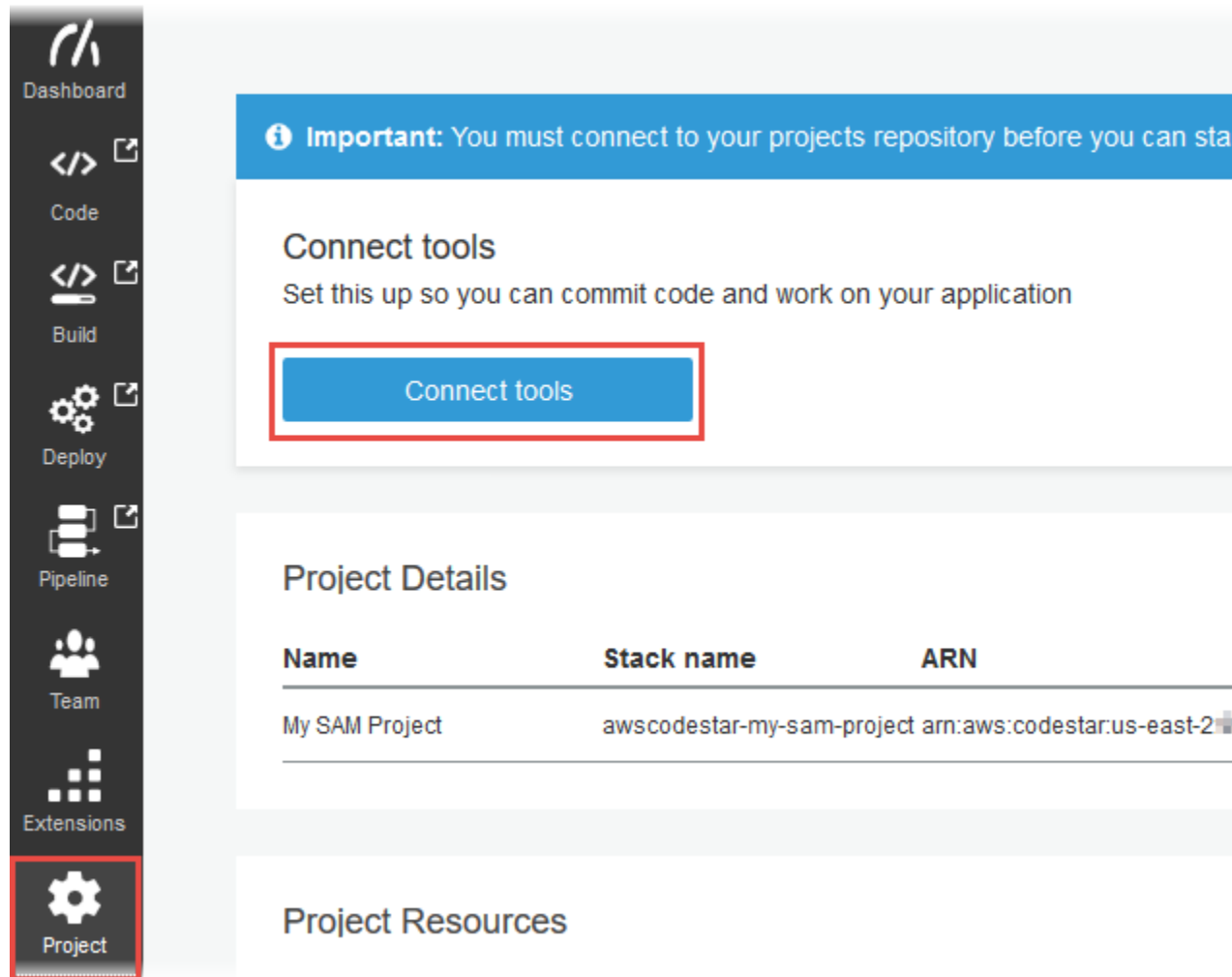
In this step, you will set up your local workstation to edit the source code in the AWS CodeStar project. Your local workstation can be a physical or virtual computer running macOS, Windows, or Linux.

1. With your project still open from the previous step, do one of the following:
 - If **You must connect to your project's repository before you can start working on the code** is displayed, choose the **Connect Tools** button.
 - In the side navigation pane, choose **Project**, and then choose the **Connect tools** button.



Important: You must connect to your project's repository before you can start working on the code.

[I have already done](#)



2. Choose the **Command line tools** tile.

If you have Visual Studio or Eclipse installed, choose the **Visual Studio** or **Eclipse** tile instead, follow the instructions, and then skip to [Step 5: Add Logic to the Web Service \(p. 32\)](#).

3. On the **Connect to your tools** page, for **Operating System**, choose the operating system running on your local workstation.
4. For **Connection Method**, choose **HTTPS**.

We recommend that you choose **HTTPS** instead of **SSH** because HTTPS has fewer setup tasks. If you must use SSH, choose **SSH**, follow the instructions, and then skip to [Step 5: Add Logic to the Web Service \(p. 32\)](#).

5. Follow the instructions to complete the following tasks:
 1. Set up Git on your local workstation.
 2. Use the IAM console to generate Git credentials for your IAM user.
 3. Clone the project's AWS CodeCommit repository onto your local workstation.

Step 5: Add Logic to the Web Service

In this step, you use your local workstation to add logic to the web service. Specifically, you add a Lambda function and then connect it to the API in API Gateway.

1. On your local workstation, go to the directory that contains the cloned source code repository.
2. In that directory, create a file named `hello.py`. Add the following code to the file, and then save the file:

```
import json

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

The preceding code simply outputs the string `Hello` along with whatever string the caller sends to the function.

3. In the same directory, open the `template.yml` file. Add the following code to the end of the file, and then save the file:

```
Hello:
  Type: AWS::Serverless::Function
  Properties:
    Handler: hello.handler
    Runtime: python2.7
    Role:
      Fn::ImportValue:
        !Join ['-', [!Ref 'ProjectId', !Ref 'AWS::Region', 'LambdaTrustRole']]
  Events:
    GetEvent:
      Type: Api
      Properties:
        Path: /hello/{name}
        Method: get
```

AWS SAM will use the preceding code to create a function in Lambda, add a new method and path to the API in API Gateway, and then connect this method and path to the new function.

Note

The indentation of the preceding code is important. If you don't add the code exactly it's shown, the project might not build correctly.

4. Use Git to add your file changes to the cloned repository's staging area by running the command `git add .`. Do not forget the dot (`.`), which adds all changed files.

Note

If you are using Visual Studio or Eclipse instead of the command line, the instructions for using Git might be different. Consult the Visual Studio or Eclipse documentation.

5. Use Git to commit your staged files in the cloned repository by running the command `git commit -m "Added hello.py and updated template.yaml."`
6. Use Git to push your commit to the remote repository by running the command `git push`.

Note

You might be prompted for the user name and password IAM generated for you earlier. To keep from being prompted each time you interact with the remote repository,

consider installing and configuring a Git credential manager. For example, on macOS or Linux you can run `git config credential.helper 'cache --timeout 900'` in the terminal to be prompted no sooner than every 15 minutes. Or you can run `git config credential.helper 'store --file ~/.git-credentials'` to never be prompted again. Git will store your credentials in clear text in a plain file in your home directory. For more information, see [Git Tools - Credential Storage](#) on the Git website.

After AWS CodeStar detects the push, it instructs AWS CodePipeline to use AWS CodeBuild and AWS SAM to rebuild and redeploy the web service automatically.

AWS SAM gives the new function the name `awscodestar-my-sam-project-lambda-Hello-RANDOM_ID`, where:

- **my-sam-project** is the ID of the project.
- **Hello** is the function ID as specified in the `template.yaml` file.
- ***RANDOM_ID*** is a random ID that AWS SAM assigns to the function for uniqueness.

Step 6: Test the Enhanced Web Service

In this step, you test the enhanced web service that AWS CodeStar built and deployed, based on the logic you added in the previous step.

1. With your project still open in the AWS CodeStar console, on the side navigation bar, choose **Dashboard**.
2. On the **Continuous deployment** tile, make sure the pipeline has run again and that **Succeeded** is displayed for the **Source**, **Build**, and **Deploy** stages before you continue. This might take several minutes.

Note

If **Failed** is displayed for any of the stages, see the following for troubleshooting help:

- For the **Source** stage, see [Troubleshooting AWS CodeCommit](#) in the *AWS CodeCommit User Guide*.
 - For the **Build** stage, see [Troubleshooting AWS CodeBuild](#) in the *AWS CodeBuild User Guide*.
 - For the **Deploy** stage, see [Troubleshooting AWS CloudFormation](#) in the *AWS CloudFormation User Guide*.
 - For other issues, see [Troubleshooting AWS CodeStar \(p. 95\)](#).
3. Choose the link on the **Application endpoints** tile. It should look like `https://API_ID.execute-api.REGION_ID.amazonaws.com/Prod/`, where:
 - ***API_ID*** represents the ID that API Gateway assigned to the API.
 - ***REGION_ID*** is the ID of the related AWS region.
 - **Prod** is the name of the related API deployment stage in API Gateway.

On the new tab that opens in your web browser, the web service displays the following response output:

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

4. In the tab's address box, add the path `/hello/` and your first name to the end of the URL and then press **Enter**. For example, **`https://API_ID.execute-api.REGION_ID.amazonaws.com/Prod/hello/YOUR_FIRST_NAME`**.

If your first name is Mary, the web service displays the following response output:

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

Step 7: Add a Unit Test to the Web Service

In this step, you use your local workstation to add a test that AWS CodeStar runs automatically on the web service. This test replaces the manual testing you did earlier.

1. On your local workstation, go to the directory that contains the cloned source code repository.
2. In that directory, create a file named `hello_test.py`. Add the following code to the file, and then save the file:

```
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
```

This test checks whether the output of the Lambda function is in the expected format. If so, the test succeeds. Otherwise, the test fails.

3. In the same directory, open the `buildspec.yml` file. Replace the file's contents with the following code, and then save the file:

```
version: 0.2

phases:

  install:
    commands:
      - pip install pytest

  pre_build:
    commands:
      - pytest
```

```
build:
  commands:
    - aws cloudformation package --template template.yml --s3-bucket $S3_BUCKET --
      output-template template-export.json

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

This build specification instructs AWS CodeBuild to install pytest, the Python test framework, into its build environment. AWS CodeBuild uses pytest to run the unit test. The rest of the build specification is the same as before.

4. Use Git to push these changes to the remote repository.

```
git add .

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

git push
```

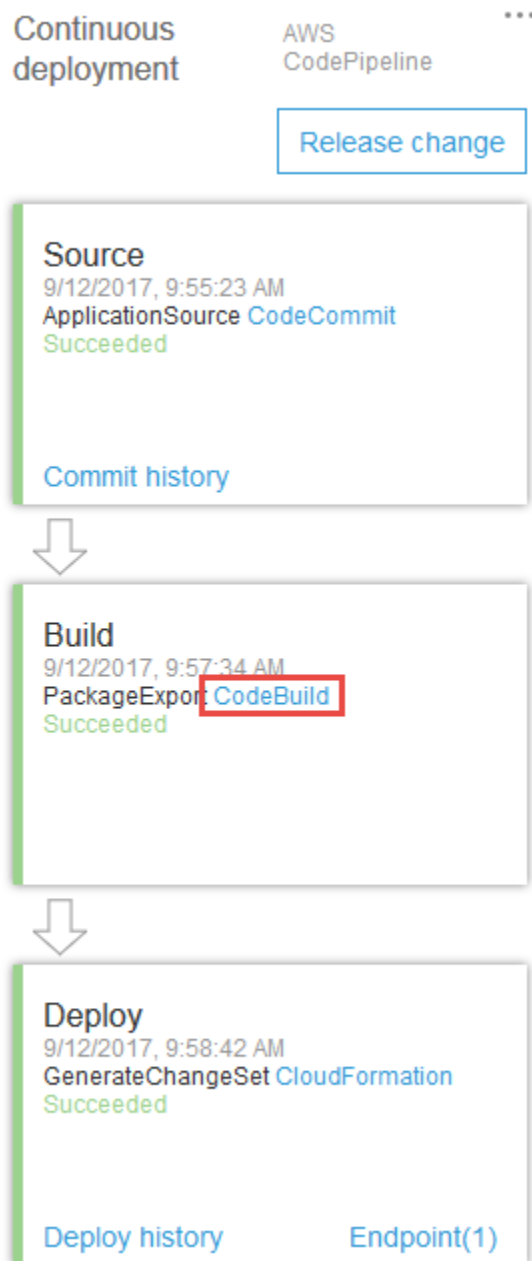
Step 8: View Unit Test Results

In this step, you see whether the unit test succeeded or failed.

1. With your project still open in the AWS CodeStar console, on the side navigation bar, choose **Dashboard**.
2. On the **Continuous deployment** tile, make sure the pipeline has run again before you continue. This might take several minutes.

If the unit test was successful, **Succeeded** is displayed for the **Build** stage. If the unit test failed, **Failed** is displayed.

3. To view the unit test result details, on the **Continuous deployment** tile, choose the **CodeBuild** link in the **Build** stage.



4. In the AWS CodeBuild console, on the **Build Project: my-sam-project** page, in **Build history**, choose the link in the **Build run** column of the table.

AWS CodeBuild

Build projects

Build history

Build Project: my-sam-project

Back

Delete

Edit project

Project

Project name my-sam-project

Description AWS CodeStar create

Source provider AWS CodePipeline

Repository

Artifacts upload location

► Project details

Build history

Retry

Stop

Start build

Build run		Submitter
<input type="checkbox"/>	► my-sam-project:a1f...	my-sam-project-Pip...

5. On the **my-sam-project:BUILD_ID** page, in **Build logs**, choose the **View entire log** link.

Phase details

	Name	Status
▶	SUBMITTED	Success
▶	PROVISIONING	Success
▶	DOWNLOAD_SOURCE	Success
▶	INSTALL	Success
▶	PRE_BUILD	Success
▶	BUILD	Success
▶	POST_BUILD	Success
▶	UPLOAD_ARTIFACTS	Success
▶	FINALIZING	Success
▶	COMPLETED	Success

Build logs

Showing the last 20 lines of build log below. [View entire log](#)

```
[Container] 2017/09/12 16:56:20 Phase context status code: Message:
[Container] 2017/09/12 16:56:20 Entering phase BUILD
```

6. In the Amazon CloudWatch Logs console, look in the log output for a test result similar to the following. In the following test result, the test passed:

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

If the test failed, there should be details in the log output to help you troubleshoot the failure.

Step 9: Clean Up

In this step, you clean up the project to avoid ongoing charges related to this project.

If you want to keep using this project, you can skip this step, but your AWS account might continue to be charged.

1. With your project still open in the AWS CodeStar console, on the side navigation bar, choose **Project**.
2. Choose **Delete project**.
3. Type the name of the project, keep the **Delete associated resources along with AWS CodeStar project** box selected, and then choose **Delete**.

Important

If you clear this box, the project record will be deleted from AWS CodeStar, but many of the project's AWS resources will remain, and your AWS account might be charged for ongoing related costs.

An Amazon S3 bucket that AWS CodeStar created for this project might still remain. To delete this remaining bucket:

1. Open the AWS CodeCommit console, at <https://console.aws.amazon.com/s3/>.
2. In the list of buckets, choose the icon next to **aws-codestar-*REGION_ID*-*ACCOUNT_ID*-my-sam-project--pipe**, where:
 - *REGION_ID* is the ID of the AWS region for the project you just deleted.
 - *ACCOUNT_ID* is your AWS account ID.
 - **my-sam-project** is the ID of the project you just deleted.
3. Choose **Empty Bucket**. Type the name of the bucket, and then choose **Confirm**.
4. Choose **Delete Bucket**. Type the name of the bucket, and then choose **Confirm**.

Next Steps

Now that you have completed this tutorial, we suggest you review the following resources:

- The [Getting Started with AWS CodeStar \(p. 5\)](#) walkthrough uses a project that creates and deploys a Node.js-based web application running on an Amazon EC2 instance.
- [AWS CodeStar Project Templates \(p. 40\)](#) describes other types of projects you can create.
- [Customize an AWS CodeStar Dashboard \(p. 55\)](#) shows you how to customize your projects' dashboards, integrate with JIRA, and more.
- [Working with AWS CodeStar Teams \(p. 62\)](#) shows you how others can help you work on your projects.

AWS CodeStar Project Templates

You can use a AWS CodeStar project template to quickly configure AWS CodeStar to support your development project. These preconfigured AWS CloudFormation templates create projects based on your choices. They include support for development projects like websites, web services, microservices, Alexa Skills, and more. You can use the search box or the filter bar to find a template.

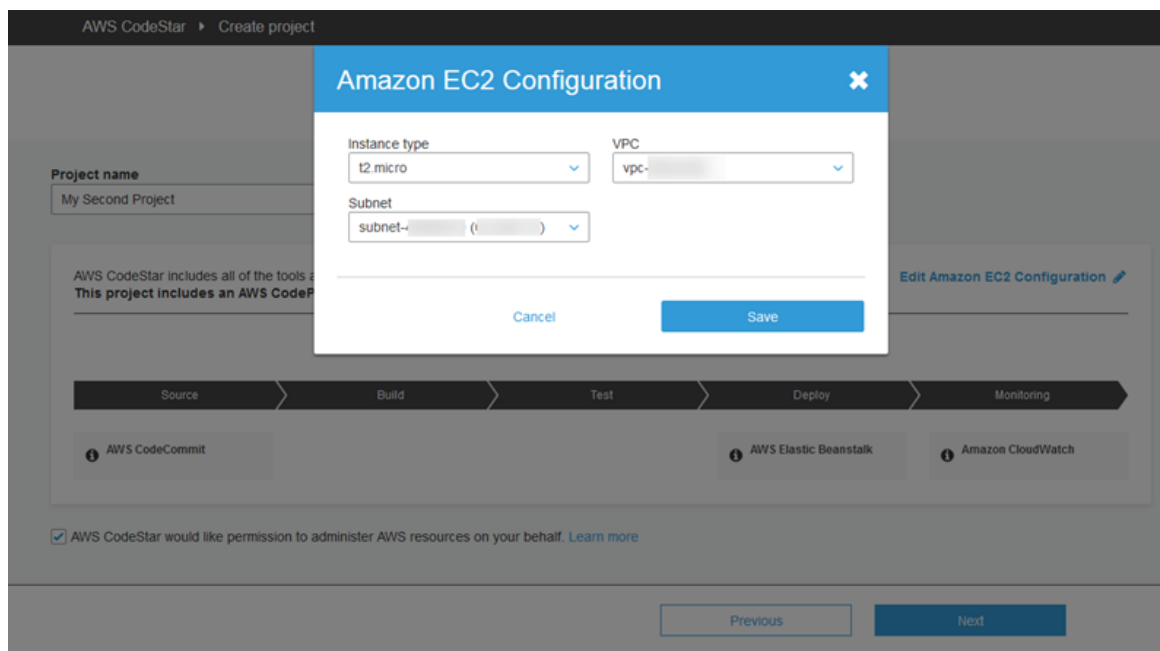
Topics

- [How Do I Choose the Right Template? \(p. 40\)](#)
- [Web Application \(p. 42\)](#)
- [Web Service \(p. 42\)](#)
- [Amazon Alexa Skill \(p. 43\)](#)

How Do I Choose the Right Template?

Each AWS CodeStar project template includes the supported programming language in its title and description. The template name also indicates whether your project is hosted on servers in the cloud (Amazon EC2, either in a managed application environment (AWS Elastic Beanstalk) or that you manage yourself) or run serverless (without Amazon EC2 instances, for example on AWS Lambda). If you see two AWS CodeStar project templates that look the same, check the description and the service bar to distinguish the differences.

After you make your choice of an AWS CodeStar project template, the page will display a list of resources that will be created for the project. All of these resources will be configured for you. You do not have to perform any manual configuration to get started with your project. If your project template includes Amazon EC2 instances, you can choose **Edit Amazon EC2 Configuration** to modify your configuration. Some of these choices, such as instance type, might affect the cost of your project. For more information, see [Create a Project in AWS CodeStar \(p. 46\)](#) and [Pricing](#).



Many of the AWS CodeStar project templates allow you to choose from a variety of options to host your web application or service in the cloud. All options offer high availability and scaling. The option you choose is configured when the project is created. You don't have to worry about configuring interoperation or setting permissions. The following table can help you determine the best fit for your software project.

Which hosting option is right for my AWS CodeStar project?

AWS Elastic Beanstalk	Amazon EC2 with AWS CodeDeploy	AWS Lambda
Automated deployments to Amazon EC2 instances.	Automated deployments to Amazon EC2 instances.	Serverless (no servers or instances to manage or administer).
Automated management of capacity and load balancing.	Flexible deployment to any instance.	AWS CodeBuild configured to build your artifacts automatically.
Team member access to Amazon EC2 instances (if your project owner allows it).	Team member access to Amazon EC2 instances (if your project owner allows it).	Amazon API Gateway configured automatically as a Lambda proxy for GET and POST calls.
End-to-end application management solution.	Building block service focused on deploying and updating software.	Code executed in response to events.

After your project is created, you can view the sample source code included in your project, including a readme file that provides details of files and directories. This readme file also includes suggestions for how to get started with the sample code.

Web Application

Choose this category if you want a project that will help you develop and deploy a web application. The AWS CodeStar project templates in this category support development in Ruby, Java, ASP.NET, PHP, and more. A source repository and continuous delivery pipeline is configured for you automatically, along with a sample application that you can use to evaluate the AWS CodeStar project. You can choose AWS services to use for your application.

All web application projects include the following resources:

- A source code repository in AWS CodeCommit or GitHub.
- A continuous deployment pipeline in AWS CodePipeline.
- A CPU utilization monitor for Amazon EC2 instances (Amazon EC2 and AWS Elastic Beanstalk projects) or an Invocations and Errors monitor (AWS Lambda projects) in Amazon CloudWatch.
- Project roles and associated policies in IAM. Policies are applied automatically to IAM users when you add those users to your project team.
- Sample code for your project, including a README.md with details of the sample.

If you choose an AWS CodeStar project template that uses Lambda, your project will also include the following resources:

- A build server and environment in AWS CodeBuild.
- A sample function in Lambda.
- A RESTful API that exposes the Lambda function in Amazon API Gateway.
- Roles for job workers in IAM.

If you chose to create a project with AWS Lambda, you can add resources to your AWS CodeStar project by editing the template.yaml file that is included in the sample code for Lambda projects. Configurable resources include:

- Applications and deployment groups in AWS CodeDeploy.
- Applications and environments in AWS Elastic Beanstalk.
- Stages and actions in a pipeline in AWS CodePipeline.
- Events in Amazon CloudWatch.
- Build projects in AWS CodeBuild.

Web Service

Choose this category if you want a project that will help you develop and deploy a RESTful web service. This template supports development in Ruby, Java, ASP.NET, PHP, and more. A source repository, build server, and continuous delivery pipeline will be configured for you automatically, along with CloudWatch metrics. This template also includes some sample code you can use to help evaluate the AWS CodeStar project and its resources.

All web service projects include the following resources:

- A source code repository in AWS CodeCommit or GitHub.
- A continuous deployment pipeline in AWS CodePipeline.
- A CPU utilization monitor for Amazon EC2 instances (Amazon EC2 and AWS Elastic Beanstalk projects) or an Invocations and Errors monitor (AWS Lambda projects) in Amazon CloudWatch.

- Sample code for your project, including a README.md with details of the sample.

If you choose a AWS CodeStar project template that uses Lambda, your project will also include the following resources:

- A build server and environment in AWS CodeBuild .
- A sample function in Lambda.
- A RESTful API that exposes the Lambda function in Amazon API Gateway.
- Roles for job workers in IAM.

To view application activity in an AWS CodeStar project template that uses Lambda, you must first invoke the function by choosing to visit the host. The host link appears on the **Continuous deployment** tile of your project.

If you chose to create a project with AWS Lambda, you can add resources to your AWS CodeStar project by editing the template.yaml file that is included in the sample code for Lambda projects. Configurable resources include:

- Applications and deployment groups in AWS CodeDeploy.
- Applications and environments in AWS Elastic Beanstalk.
- Stages and actions in a pipeline in AWS CodePipeline.
- Events in Amazon CloudWatch.
- Build projects in AWS CodeBuild.

Amazon Alexa Skill

Choose this template if you want a project for a AWS Lambda function based on an Alexa Skills blueprint for [Amazon Alexa](#). The function returns an Amazon Resource Name (ARN) that you can use as a service endpoint for your Alexa Skill when you configure it in the Alexa Developer Portal. For more information, see [Creating an AWS Lambda Function for a Custom Skill](#).

Note

Lambda functions for Alexa Skills are only supported in the US East (N. Virginia) and EU (Ireland) regions.

All Alexa Skill projects include the following resources:

- A source code repository in AWS CodeCommit or GitHub.
- A continuous deployment pipeline in AWS CodePipeline.
- An Invocations and Errors monitor in Amazon CloudWatch.
- Sample code for your project, including a README.md with details of the sample.
- A build server and environment in AWS CodeBuild .
- A sample function in Lambda.
- A RESTful API that exposes the Lambda function in Amazon API Gateway.
- Roles for job workers in IAM.

To view application activity in an AWS CodeStar project template that uses Lambda, you must first invoke the function by choosing to visit the host. The host link appears on the **Continuous deployment** tile of your project.

You can add resources to your AWS CodeStar project by editing the template.yaml file that is included in the sample code for Lambda projects. Configurable resources include:

- Applications and deployment groups in AWS CodeDeploy.
- Applications and environments in AWS Elastic Beanstalk.
- Stages and actions in a pipeline in AWS CodePipeline.
- Events in Amazon CloudWatch.
- Build projects in AWS CodeBuild.

Working with Projects in AWS CodeStar

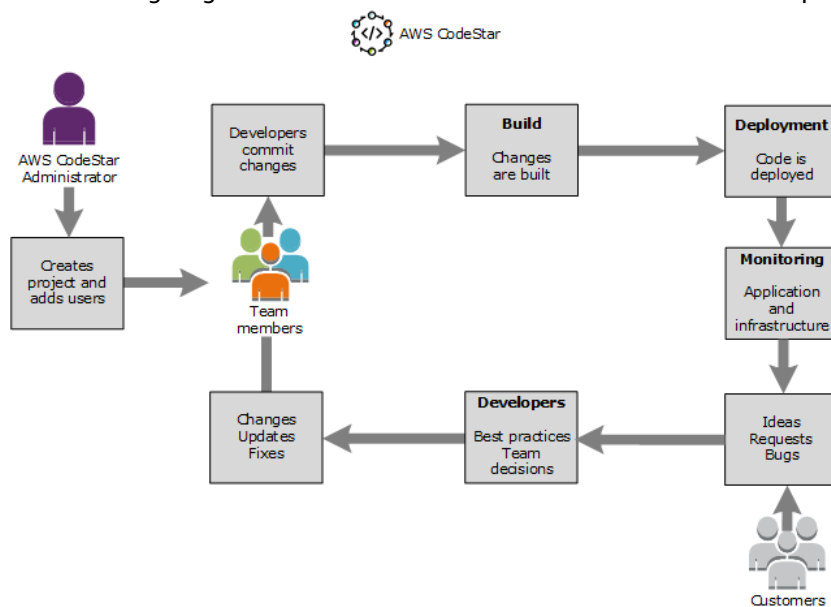
When you use an AWS CodeStar project template, you can quickly create a project that is already configured with the resources you need, including:

- Source repository.
- Build environment.
- Deployment and hosting resources.
- Programming language.

The template even includes sample source code so you can start working with your project right away.

After you have a project, you can start working with it by adding or removing resources, customizing your project dashboard, and monitoring progress.

The following diagram illustrates a basic workflow in an AWS CodeStar project.



A developer with the `AWSCodeStarFullAccess` policy applied creates a project and adds team members to it. Together they write, build, test, and deploy code. The project dashboard provides tools that can be used in real time to view application activity and monitor builds, the flow of code through the deployment pipeline, and more. The team uses the team wiki tile to share information, best practices, and links. They integrate their issue-tracking software to help them track progress and tasks. As customers provide requests, information, and ideas, the team adds this information to the project and integrates it into their project planning and development. As the project grows, the team adds more team members to support their growing code base.

Create a Project in AWS CodeStar

You use the AWS CodeStar console to create a project. If you use a project template, it sets up the required resources. It also includes sample code you can use to start coding and to understand how the project resources work together.

To create a project, sign in to the AWS Management Console with an IAM user that has the `AWSCodeStarFullAccess` policy or equivalent permissions. For more information, see [Setting Up AWS CodeStar \(p. 3\)](#).

Note

You must complete the steps in [Setting Up AWS CodeStar \(p. 3\)](#) before you can complete the procedures in this topic.

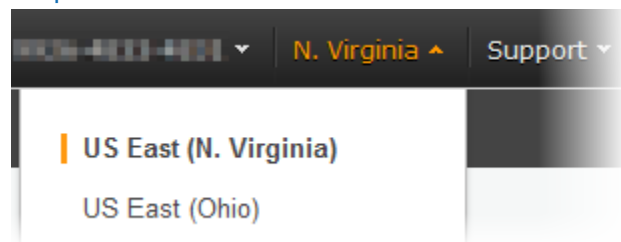
Create a Project in AWS CodeStar

Use the AWS CodeStar console to create a project.

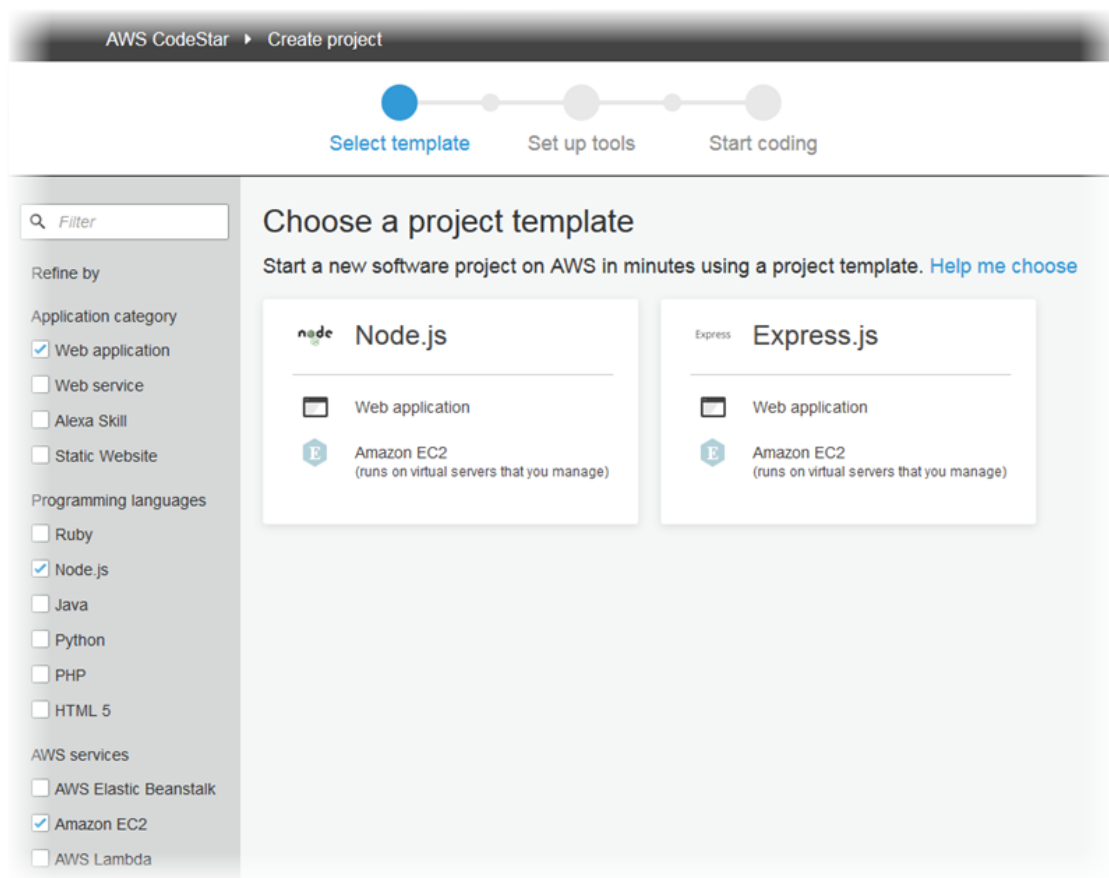
To create a project in AWS CodeStar

1. Sign in to the AWS Management Console, and then open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Make sure that you are signed in to the AWS region where you want to create the project and its resources. For example, to create a project in US East (Ohio), make sure you have that region selected. For information about AWS regions where AWS CodeStar is available, see [Regions and Endpoints](#) in the *AWS General Reference*.



2. On the **AWS CodeStar** page, choose **Create a new project**. (If you are the first user to create a project, choose **Start a project**.)
3. On the **Choose a project template** page, choose the project type from the list of AWS CodeStar project templates. You can use the filter bar to narrow your choices. For example, for a web application project written in Node.js that will be deployed to Amazon EC2 instances, select the **Web application**, **Node.js**, and **Amazon EC2** check boxes. Then choose from the templates available for that set of options.



For more information, see [AWS CodeStar Project Templates \(p. 40\)](#).

4. In **Project name**, type a name for the project, such as *My First Project*. The ID for the project is derived from this project name, but is limited to 15 characters and allowed characters.

For example, the default ID for a project named *My First Project* is *my-first-projec*. This project ID is the basis for the names of all resources associated with the project. For example, AWS CodeStar uses this project ID as part of the URL for your code repository as well as the names of related security access roles and policies in IAM. After the project is created, the project ID cannot be changed, so make sure you are okay with this project ID. To edit the project ID before you create the project, choose **Edit**.

For information about the limits on project names and project IDs, see [Limits in AWS CodeStar \(p. 93\)](#).

Note

Project IDs must be unique for your AWS account within an AWS region.

Project details

Project name
My First Project

Project ID [Edit](#)
my-first-project

Which repository do you want to use?
AWS CodeStar will store the project's source code with the service you choose here.

AWS CodeCommit
Highly available Git source control from AWS. Includes encryption, IAM integration, and more.

GitHub
Creates a GitHub source repository for this project. Requires an existing GitHub account.

Repository name*
My-First-Project

[Previous](#) [Next](#)

5. Choose the repository provider to store this project's source code with: **AWS CodeCommit** or **GitHub**.
6. If you chose **AWS CodeCommit**, for **Repository name**, accept the default AWS CodeCommit repository name that AWS CodeStar suggests, or type a different AWS CodeCommit repository name of your choice. Then skip ahead to step 8 in this procedure.
7. If you chose **GitHub**, then choose **Connect with GitHub**.

1. If the **Sign in to GitHub** page is displayed, type your GitHub username or email address and password, and then choose **Sign in**.

Note

To complete this page, you must have a GitHub account. For more information, see [Join GitHub](#) on the GitHub website.

2. If the **Two-factor authentication** page displays, for **Authentication code**, type the code that GitHub sends you. Then choose **Verify**.
3. On the **Authorize AWS CodeStar** page, choose **Authorize**.

Note

When you choose **Authorize**, you allow AWS CodeStar to create a GitHub repository for your personal GitHub account, or for any GitHub organization where you have permissions (which is marked with a green check icon in **Organization access**).

To add a GitHub organization to the **Organization access** list, ask one of the organization's owners to invite you to the organization by following the instructions in [Inviting users to join your organization](#) on the GitHub Help website. After you join the organization, refresh the **Authorize AWS CodeStar** page to see the organization in the list.

To get permissions to authorize a GitHub organization that is in the list but does not have a green check icon, choose **Grant**. If you see **Request** instead, choose it, and then ask one of the organization's owners to allow AWS CodeStar to create a GitHub repository in the organization by following the instructions in [Approving OAuth Apps](#)

for [your organization](#) on the GitHub Help website. After the owner does this, refresh the **Authorize AWS CodeStar** page to see the **Grant** button.

4. For **Owner**, choose the GitHub organization or your personal GitHub account that you want AWS CodeStar to create the GitHub repository for.
5. For **Repository name**, accept the default GitHub repository name that AWS CodeStar suggests, or type a different GitHub repository name of your choice.
6. Choose **Public repository** or **Private repository** to make the GitHub repository public or private.

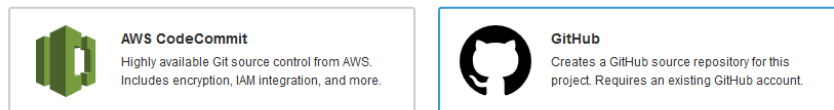
Note

Depending on your GitHub account type, GitHub may not allow you to create a private repository. For more information, see [GitHub Pricing](#) on the GitHub website.

7. For **Repository description**, provide an optional description for the GitHub repository.

Which repository do you want to use?

AWS CodeStar will store the project's source code with the service you choose here.



Connect to GitHub

You must give AWS CodeStar permission to create a repository in your GitHub account.

✓ AWS CodeStar is now connected to GitHub

Owner: / Repository name:

Repository description (optional):

Previous

Next

8. Choose **Next**.
9. Review the resources and configuration details. Choose **Edit Amazon EC2 Configuration** (where available) if your project will deploy to Amazon EC2 instances and you want to make changes. For example, you can choose from available instance types for your project.

Note

Different Amazon EC2 instance types provide different levels of computing power and might have different associated costs. For more information, see [Amazon EC2 Instance Types](#) and [Amazon EC2 Pricing](#).

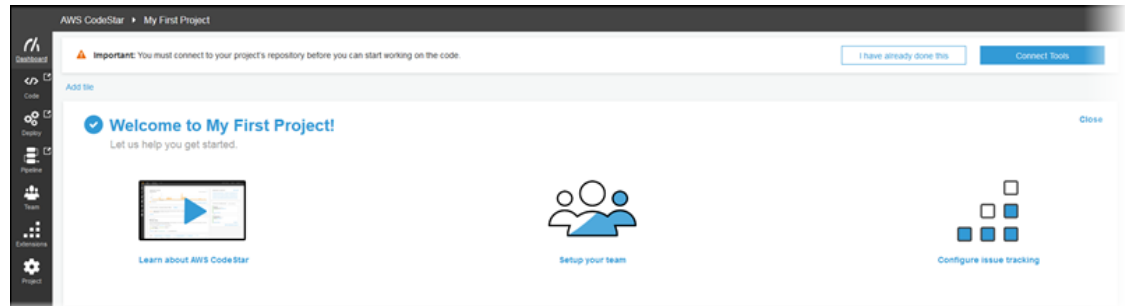
If you have more than one virtual private cloud (VPC) or multiple subnets created in Amazon Virtual Private Cloud, you can also choose the VPC and subnet to use. However, if you choose an Amazon EC2 instance type that is not supported on dedicated instances, you cannot choose a VPC whose instance tenancy is set to **Dedicated**.

For more information, see [What Is Amazon VPC?](#) and [Dedicated Instance Basics](#).

10. Leave the **AWS CodeStar would like permission to administer AWS resources on your behalf** check box selected. If this box is not selected, you will not be able to create a project. For more information about the service role, the policy, and its permissions, see [AWS CodeStar Service Role Policy and Permissions](#) (p. 83).

Choose **Next** or **Create project**. (The displayed choice depends on your project template.)

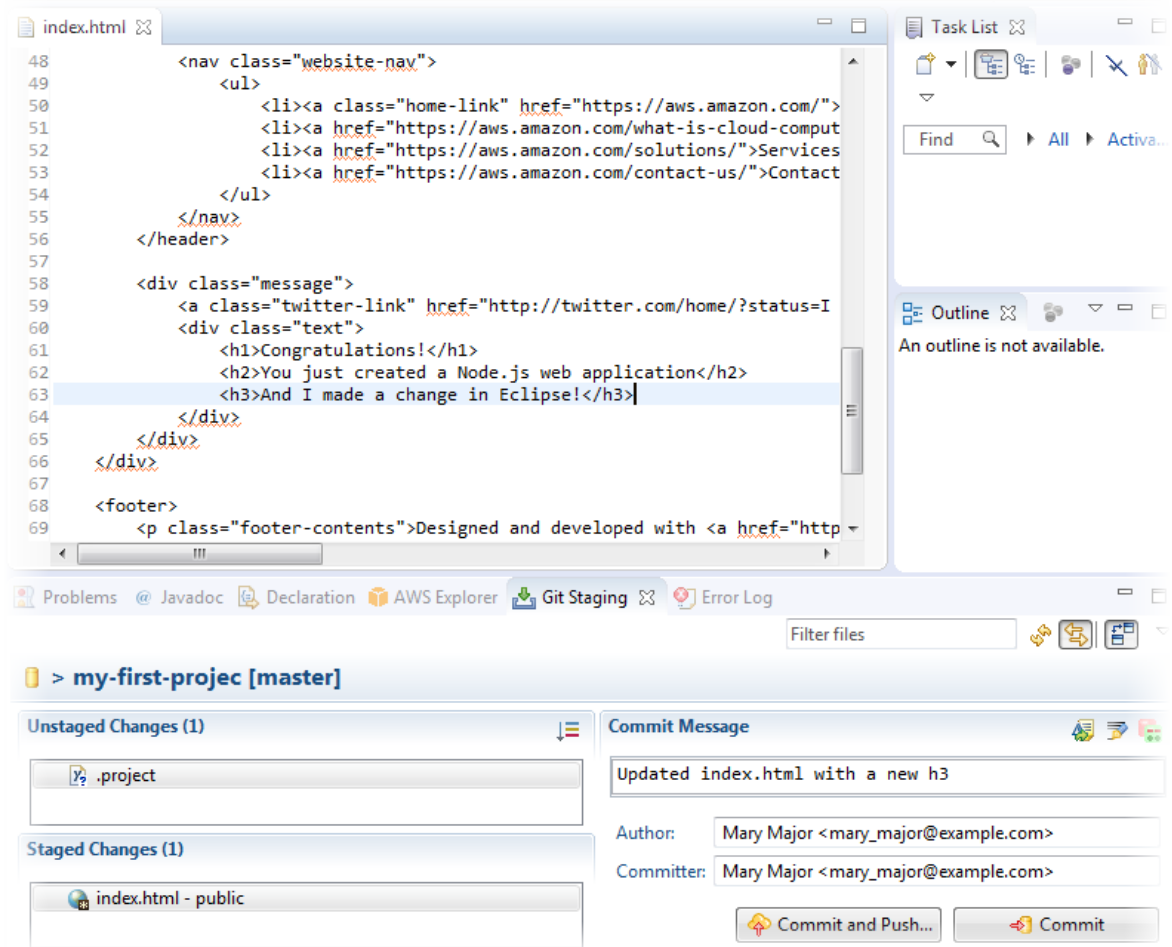
11. In **Choose an Amazon EC2 Key Pair**, choose the Amazon EC2 key pair you created in [Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects \(p. 4\)](#) in *Setting Up*. Select **I acknowledge that I have access to the private key file for this key pair**, and then choose **Create project**.
12. It might take a few minutes to create the project (including the repository). After your project has a repository, you can use the **Choose how you want to edit your project code** page to configure access to it, or you can choose **Skip** and configure access later. After your project has been created, you will see a **Welcome** tile that contains useful links. You can use these links to optionally configure other items, such as your [user profile in AWS CodeStar \(p. 72\)](#).



While your project is being created, you can [add team members \(p. 63\)](#) or [configure access \(p. 50\)](#) to your project repository from the command line or your favorite IDE.

Use an IDE with AWS CodeStar

When you integrate an integrated development environment (IDE) with AWS CodeStar, you can continue to write and develop code in your preferred environment. The changes you make will be included in the AWS CodeStar project each time you commit and push your code.



Topics

- [Use Eclipse with AWS CodeStar \(p. 51\)](#)
- [Use Visual Studio with AWS CodeStar \(p. 55\)](#)

Use Eclipse with AWS CodeStar

You can use Eclipse to make code changes and develop software in an AWS CodeStar project. You can edit your AWS CodeStar project code with Eclipse and then commit and push your changes to the source repository for the AWS CodeStar project.

Note

The information in this topic applies only to AWS CodeStar projects that store their source code in AWS CodeCommit. If your AWS CodeStar project stores its source code in GitHub, you can use a tool such as EGit for Eclipse. For more information, see the [EGit website](#) and [EGit Documentation](#) on the EGit website.

If the AWS CodeStar project stores its source code in AWS CodeCommit, you must install a version of the AWS Toolkit for Eclipse that supports AWS CodeStar. You must also be a member of the AWS CodeStar project team with the owner or contributor role.

To use Eclipse, you'll also need:

- An IAM user that has been added to an AWS CodeStar project as a team member.

- If the AWS CodeStar project stores its source code in AWS CodeCommit, [Git credentials \(p. 15\)](#) (user name and password) for the IAM user.
- Sufficient permissions to install Eclipse and the AWS Toolkit for Eclipse on your local computer.

Topics

- [Step 1: Install AWS Toolkit for Eclipse \(p. 52\)](#)
- [Step 2: Import Your AWS CodeStar Project to Eclipse \(p. 52\)](#)
- [Step 3: Edit AWS CodeStar Project Code in Eclipse \(p. 54\)](#)

Step 1: Install AWS Toolkit for Eclipse

The Toolkit for Eclipse is a software package you can add to Eclipse. It is installed and managed in the same way as other software packages in Eclipse. The AWS CodeStar toolkit is included as part of the Toolkit for Eclipse.

To install the Toolkit for Eclipse with the AWS CodeStar module

1. Install Eclipse on your local computer if you don't have a supported version already installed. Supported versions of Eclipse include Luna, Mars, and Neon.
2. Download and install the Toolkit for Eclipse. For more information, see the [AWS Toolkit for Eclipse Getting Started Guide](#).
3. In Eclipse, choose **Help**, and then choose **Install New Software**.
4. In **Available Software**, choose **Add**.
5. In **Add Repository**, choose **Archive**, browse to the location where you saved the .zip file, and open the file. Leave **Name** blank, and then choose **OK**.
6. In **Available Software**, choose **Select all** to select both **AWS Core Management Tools** and **Developer Tools**, and then choose **Next**.
7. In **Install Details**, choose **Next**.
8. In **Review Licenses**, review the license agreements. If you agree, choose **I accept the terms of the license agreement** and choose **Finish**. Restart Eclipse.

Step 2: Import Your AWS CodeStar Project to Eclipse

After you have installed the Toolkit for Eclipse, you can import AWS CodeStar projects and edit, commit, and push code from the IDE.

Note

You can add multiple AWS CodeStar projects to a single workspace in Eclipse, but if you do, you must update your project credentials when you change from one project to another.

To import an AWS CodeStar project

1. Open the drop-down menu on the AWS toolbar icon and choose **Import AWS CodeStar Project**. Alternatively, choose **File**, and then choose **Import**. In **Select**, expand **AWS**, and then choose **AWS CodeStar Project**.

Choose **Next**.

2. In **AWS CodeStar Project Selection**, choose your AWS profile and the region where the AWS CodeStar project is hosted. If you do not have an AWS profile configured with an access key and secret key on your computer, choose **Configure AWS accounts** and follow the instructions.

In **Select AWS CodeStar project and repository**, choose your AWS CodeStar project from the list. In **Configure Git credentials**, provide the user name and password you generated for access to the project's repository. (If you don't have Git credentials, see [Getting Started \(p. 15\)](#).) Choose **Next**.

AWS CodeStar Project Selection
Select the AWS CodeStar project you want to checkout from the remote host.

Select AWS account and region:

Select Account: **default** [Configure AWS accounts...](#)

Select Region: **US**

Select AWS CodeStar project and repository:

Project Name	Project ID	Project Description
My First Project	my-first-projec	AWS CodeStar created project

Select repository: **my-first-projec**

Configure Git credentials:

You can manually copy and paste Git credentials for AWS CodeCommit below. Alternately, you can import them from a downloaded .csv file. To learn how to generate Git credentials, see [Create Git Credentials for HTTPS Connections to AWS CodeCommit](#).

User name:

Password:

☐ Show password

3. All branches of the project's repository are selected by default. If you don't want to import one or more branches, clear the boxes, and then choose **Next**.
4. In **Local Destination**, choose a destination where the import wizard will create the local repo on your computer, and then choose **Finish**.
5. In **Project Explorer**, expand the project tree to browse the files in the AWS CodeStar project.

Step 3: Edit AWS CodeStar Project Code in Eclipse

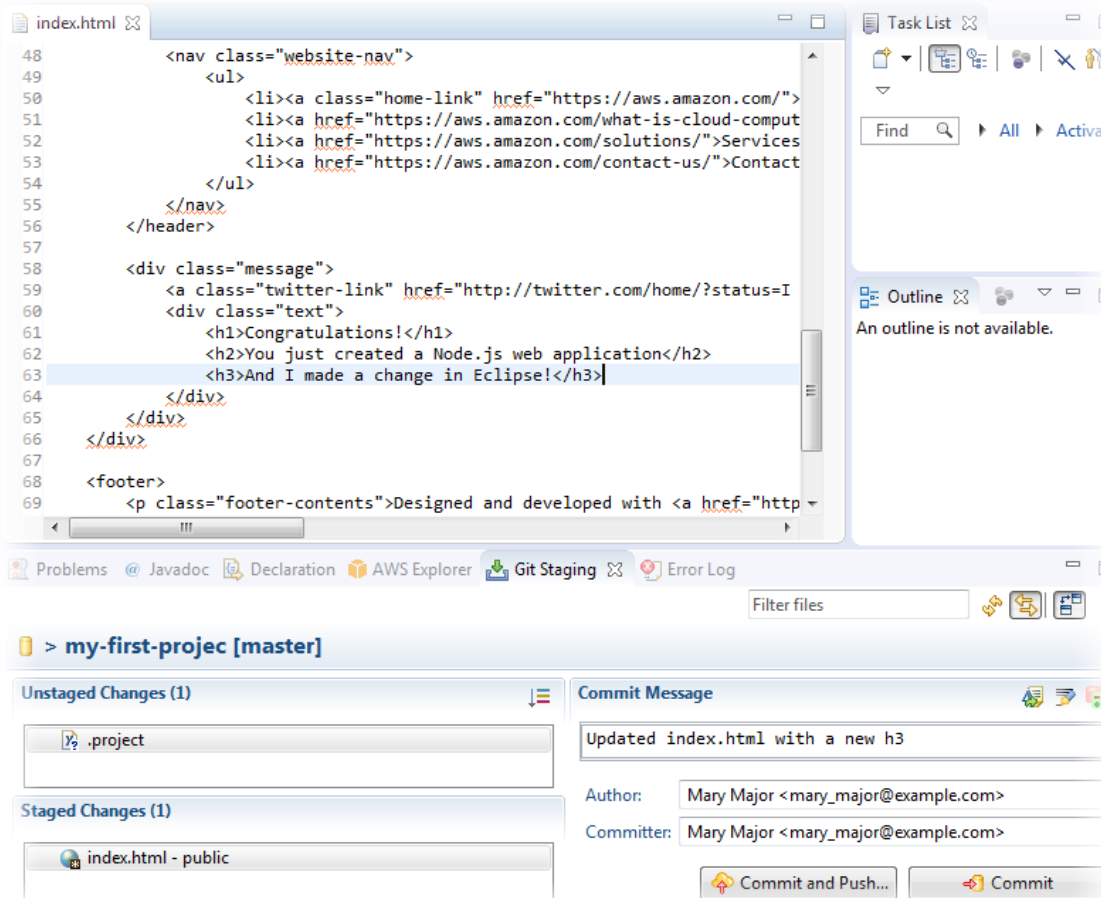
After you have imported an AWS CodeStar project into an Eclipse workspace, you can edit the code for the project, save your changes, and commit and push your code to the source repository for the project. This is the same process you follow for any Git repository using the EGit plugin for Eclipse. For more information, see the [EGit User Guide](#).

To edit project code and make your first commit to the source repository for an AWS CodeStar project

1. In **Project Explorer**, expand the project tree to browse the files in the AWS CodeStar project.
2. Edit one or more code files and save your changes.
3. When you are ready to commit your changes, open the context menu for that file, choose **Team**, and then choose **Commit**.

You can skip this step if you have the **Git Staging** window already open in your project view.

4. In the **Git Staging** window, stage your changes by moving modified files into **Staged Changes**. Type a commit message in **Commit Message**, and then choose **Commit and Push**.



To view the deployment of your code changes, return to the dashboard for your project. For more information, see [Step 3: View Your Project \(p. 11\)](#).

Use Visual Studio with AWS CodeStar

You can use Visual Studio to make code changes and develop software in an AWS CodeStar project.

Note

The information in this topic applies only to AWS CodeStar projects that store their source code in AWS CodeCommit. If your AWS CodeStar project stores its source code in GitHub, you can use a tool such as the GitHub Extension for Visual Studio. For more information, see the [Overview](#) page on the GitHub Extension for Visual Studio website and [Getting Started with GitHub for Visual Studio](#) on the GitHub website.

If the AWS CodeStar project stores its source code in AWS CodeCommit, to use Visual Studio to edit code in the source repository for an AWS CodeStar project, you must install a version of the AWS Toolkit for Visual Studio that supports AWS CodeStar. You must be a member of the AWS CodeStar project team with the Owner or Contributor role.

To use Visual Studio, you'll also need:

- An IAM user that has been added to an AWS CodeStar project as a team member.
- If the AWS CodeStar project stores its source code in AWS CodeCommit, AWS credentials for your IAM user, for example your access key and secret key.
- Sufficient permissions to install Visual Studio and the AWS Toolkit for Visual Studio on your local computer.

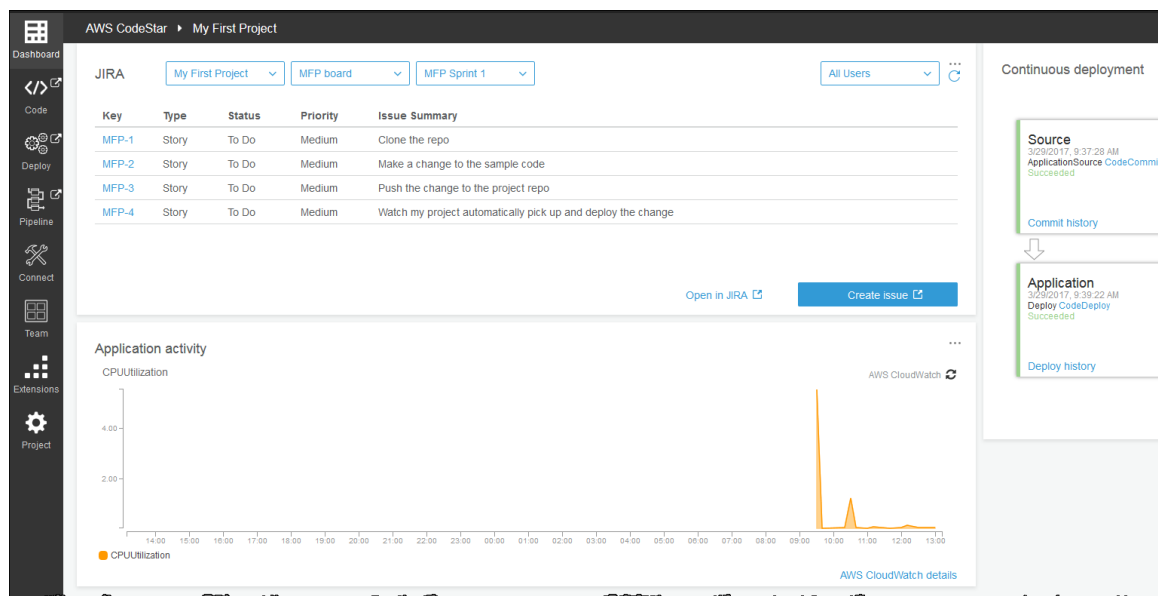
The Toolkit for Visual Studio is a software package you can add to Visual Studio. It is installed and managed in the same way as other software packages in Visual Studio.

To install the Toolkit for Visual Studio with the AWS CodeStar module and configure access to your project repository

1. Install Visual Studio on your local computer if you don't have a supported version already installed.
2. Download and install the Toolkit for Visual Studio and save the .zip file to a local folder or directory. When prompted on the **Getting Started with the AWS Toolkit for Visual Studio** page, type or import your AWS credentials, and then choose **Save and Close**.
3. In **Visual Studio**, open **Team Explorer**. In **Hosted Service Providers**, find **AWS CodeCommit**, and choose **Connect**.
4. In **Manage Connections**, choose **Clone**. Choose your project's repository and the folder you want to clone the repository into on your local computer, and then choose **OK**.
5. If the AWS CodeStar project stores its source code in AWS CodeCommit, and if prompted to create Git credentials, choose **Yes**. The toolkit will attempt to create credentials on your behalf. Save the credentials file when prompted in a secure location. This is the only opportunity you will have to save these credentials. If the toolkit cannot create credentials on your behalf, or if you chose **No**, you must create and provide your own Git credentials. For more information, see [To set up your computer to commit changes \(p. 15\)](#), or follow the on-screen directions.
6. When you have finished cloning the project, you're ready to start editing your code in Visual Studio and committing and pushing your changes to your project's repository in AWS CodeCommit.

Customize an AWS CodeStar Dashboard

You can customize your project dashboard by adding, removing, and moving tiles. You can also customize the team wiki tile to show information about your project.



Topics

- [Add, Remove, or Move Tiles on Your Dashboard \(p. 56\)](#)
- [Add a Project Extension to Your Dashboard \(p. 56\)](#)
- [Customize the Team Wiki Tile \(p. 57\)](#)

Add, Remove, or Move Tiles on Your Dashboard

You can change the appearance of your project dashboard by adding tiles, removing tiles, or changing the order and position of tiles on your dashboard.

To change the appearance of your project dashboard, do one or more of the following:

- To add a tile, on the project dashboard, choose **Add tile** and choose the tile from the list. You can only add one of each type of tile.
- To remove a tile, on the project dashboard, choose the ellipsis menu on the tile, and then choose **Remove from dashboard**.
- To change the position of a tile on the dashboard, drag it to the position where you want it to appear.

Add a Project Extension to Your Dashboard

AWS CodeStar includes extensions that add tiles and functionality to your dashboard. For example, you can configure the JIRA extension to add and configure a JIRA tile for your project dashboard.

To add a project extension, in the navigation bar for your project, choose **Extensions**. Choose **Add extension**, and then follow the instructions for setting up the extension. Depending on the extension and the configuration of your project dashboard tiles, you might need to choose the ellipsis menu and choose **Add tile** after you have configured the extension. Configuring an extension does not add the tile to the dashboard automatically.

Note

After you have added the extension, you can manage it by returning to the **Extensions** page and choosing the ellipsis menu for that extension.

Customize the Team Wiki Tile

Each AWS CodeStar project includes a customizable team wiki tile. You can change the name of this tile as well as its contents. You can use this customizable tile to share links to team resources or highlight code samples. Every project team member can view this tile, but only team members who are assigned a Contributor or Owner role can modify its contents. This tile supports both plain text and [CommonMark](#) content, with the following differences:

- You can highlight programming language syntax in code blocks. To do this, specify the language, followed by the code. For example, for JavaScript:

```
```JavaScript
var hello = function() {
 console.log("hello world");
}
```
```

- Inline embedding of images is not supported.

Note

Do not use this tile to store confidential data.

To customize a team wiki tile in a project

- Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
Choose the project from the list of projects.
- In the project dashboard, choose the ellipsis button for the project information tile, and then choose **Edit**.
- In **Markdown Editor**, enter the new tile name in **Title**. In **Markdown content**, add plain text or [CommonMark](#) content. For example, you could add a link to your team project wiki or other content.
Choose **Save**.

For a step-by-step example, see [Step 4: Customize the Team Wiki Tile and the Project Dashboard \(p. 14\)](#).

Change AWS Resources in an AWS CodeStar Project

After you create a project in AWS CodeStar, you may want to change the default set of AWS resources that AWS CodeStar adds to the project. The following information describes the types of changes AWS CodeStar supports.

Supported Resource Changes

The following table lists the types of changes you may want to make to default AWS resources in an existing AWS CodeStar project and whether they are supported.

| Change | Supported | Notes |
|---------------------------------|-----------|---|
| Add a stage to AWS CodePipeline | Yes | See Add a Stage to AWS CodePipeline (p. 58) |

| Change | Supported | Notes |
|---|-----------|---|
| Change Elastic Beanstalk environment settings | Yes | See Change AWS Elastic Beanstalk Environment Settings (p. 59) |
| Change an AWS Lambda function's code or settings, its related IAM role, or its related API in Amazon API Gateway | Yes | See Change an AWS Lambda Project (p. 59) |
| Add AWS X-Ray support | No | |
| Switch to a different deployment target (for example, deploy to AWS Elastic Beanstalk instead of AWS CodeDeploy) | No | |
| Change an IAM role definition | No | |
| Add a friendly web endpoint name | No | |
| Change the AWS CodeCommit repository name (for an AWS CodeStar project connected to AWS CodeCommit) | No | |
| Disconnect the GitHub repository (for an AWS CodeStar project connected to GitHub) and then reconnect the repository to that project, or connect any other repository to that project | No | <p>You can use the AWS CodePipeline console (not the AWS CodeStar console) to disconnect and reconnect to GitHub in a pipeline's Source stage. However, if you reconnect the Source stage to a different GitHub repository, then in the AWS CodeStar dashboard for the project, the information in the Application endpoints, Commit history, and GitHub Issues tiles may be wrong or out of date.</p> <p>Disconnecting the GitHub repository does not remove that repository's information from the commit history and GitHub issues tiles in the AWS CodeStar dashboard for the project. To remove this information, use the GitHub website to disable access to GitHub from the AWS CodeStar project. To do this, on the GitHub website, revoke access using the Authorized OAuth Apps section of the settings page for your GitHub account profile.</p> |
| Disconnect the AWS CodeCommit repository (for an AWS CodeStar project connected to AWS CodeCommit) and then reconnect the repository to that project, or connect any other repository to that project | No | |

Add a Stage to AWS CodePipeline

You can add a new stage to a pipeline that AWS CodeStar creates in a project. For more information, see [Edit a Pipeline in AWS CodePipeline](#) in the *AWS CodePipeline User Guide*.

Note

If the new stage depends on any AWS resources that AWS CodeStar did not create, then the pipeline may break. This is because the IAM role that AWS CodeStar created for AWS CodePipeline may not have access to those resources by default.

To attempt to give AWS CodePipeline access to AWS resources that AWS CodeStar did not create, you may want to change the IAM role that AWS CodeStar created. However, note that this is not supported. This is because AWS CodeStar may remove your IAM role changes whenever it performs regular update checks on the project.

Change AWS Elastic Beanstalk Environment Settings

You can change the settings of an Elastic Beanstalk environment that AWS CodeStar creates in a project. For more information, see [The AWS Elastic Beanstalk Environment Management Console](#) in the *AWS Elastic Beanstalk Developer Guide*.

Change an AWS Lambda Project

You can change the code or settings of a Lambda function, or its related IAM role or API Gateway API, that AWS CodeStar creates in a project. To do this, we recommend you use the AWS Serverless Application Model (AWS SAM) along with the `template.yaml` file in your project's AWS CodeCommit repository. This `template.yaml` file defines your function's name, handler, runtime, related IAM role, and related API in API Gateway. For more information, see [How to create serverless applications using AWS SAM](#) on the GitHub website.

Working with Project Tags in AWS CodeStar

You can associate tags with projects in AWS CodeStar. Tags can help you manage your projects. For example, you could add a tag with a key of `Release` and a value of `Beta` to any project your organization is working on for an upcoming beta release.

Add a Tag to a Project

1. With the project open in the AWS CodeStar console, in the navigation pane, choose **Project**.
2. In the **Tags** area, for **Create new tag**, type the tag's name in the **Key** box. Then type the tag's value in the **Value** box.
3. Choose **Add tag**.

Remove a Tag from a Project

1. With the project open in the AWS CodeStar console, in the navigation pane, choose **Project**.
2. In the **Tags** area, for **Current tags**, find the tag you want to remove.
3. In the **Options** column for that tag, choose **Remove**.

Note

You have approximately three seconds to undo the remove operation. To keep the tag, choose **Undo**. Otherwise, the tag is permanently deleted.

Get a List of Tags for a Project

Use the AWS Command Line Interface (AWS CLI) to run the AWS CodeStar `list-tags-for-project` command, specifying the name of the project. For example:

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

If successful, a list of tags appears in the output, similar to the following:

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

Delete an AWS CodeStar Project

If you no longer need a project, you can delete it and its resources so that you do not incur any further charges in AWS. When you delete a project, all team members are removed from that project. Their project roles are removed from their IAM users, but their user profiles in AWS CodeStar are not changed.

Important

Deleting a project in AWS CodeStar cannot be undone. By default all AWS resources for the project are deleted in your AWS account, including:

- The AWS CodeCommit repository for the project along with anything stored in that repository.
- The AWS CodeStar project roles and the associated IAM policies configured for the project and its resources.
- Any Amazon EC2 instances created for the project.
- The deployment application and associated resources, such as:
 - An AWS CodeDeploy application and associated deployment groups.
 - An AWS Lambda function and associated API Gateway APIs.
 - An AWS Elastic Beanstalk application and associated environment.
- The continuous deployment pipeline for the project in AWS CodePipeline.
- The AWS CloudFormation stacks associated with the project.

To delete all project resources along with the project, select the **Delete associated resources along with AWS CodeStar project** check box. If you clear this option, the project will be deleted in AWS CodeStar, and the project roles that enabled access to those resources will be deleted in IAM, but all other resources will be retained. You might continue to incur charges for these resources in AWS. If you decide you no longer want one or more of these resources, you must manually delete them. For more information about manually deleting resources after a project has been deleted, see [Project deletion: An AWS CodeStar project was deleted, but resources still exist \(p. 96\)](#).

If you decide to keep resources when deleting a project, as a best practice, copy the list of resources from the project details page before you delete an AWS CodeStar project. This way, you will have a record of all resources that you have kept, even though the project no longer exists.

Topics

- [Delete a Project in AWS CodeStar Using the Console \(p. 61\)](#)

- [Delete a Project in AWS CodeStar Using the AWS CLI \(p. 61\)](#)

Delete a Project in AWS CodeStar Using the Console

Use the AWS CodeStar console to delete a project.

To delete a project in AWS CodeStar

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
2. Find the project in the list, and from the ellipsis menu, choose **Delete**.

Alternatively, open the project, and in the navigation pane, choose **Project**. On the project details page, choose **Delete project**.

3. In the box next to **Type the following project ID to confirm**, type the ID of the project, and then choose **Delete**.

Deleting a project can take several minutes. After it's deleted, the project no longer appears in the list of projects in the AWS CodeStar console.

Important

By default, when you delete a project, all resources listed under **Project resources** are deleted. If you clear the check box, the project resources will be retained. For more information, go [here \(p. 60\)](#).

If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resources are not deleted, even if the check box is selected.

Delete a Project in AWS CodeStar Using the AWS CLI

You can use the AWS CLI to delete a project in AWS CodeStar.

To delete a project in AWS CodeStar (AWS CLI)

1. >At a terminal (Linux, macOS, or Unix) or command prompt (Windows), run the **delete-project** command, including the name of the project. For example, to delete a project with the ID *my-2nd-project*:

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

This command returns output similar to the following:

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

2. Run the **list-projects** command and verify that the deleted project no longer appears in the list of projects associated with your AWS account.

```
aws codestar list-projects
```

Working with AWS CodeStar Teams

After you create a development project, you'll want to grant access to others so you can work together. In AWS CodeStar, each project has a *project team*. A user can belong to multiple AWS CodeStar projects and have different AWS CodeStar roles (and thus, different permissions) in each. In the AWS CodeStar console, users see all projects associated with your AWS account, but they will only be able to view and work on those projects in which they are team members.

Team members can choose a friendly name for themselves. They can also add an email address so other team members can contact them. Team members who are not owners cannot change their AWS CodeStar role for the project.

Each project in AWS CodeStar has three roles:

Roles and Permissions in an AWS CodeStar Project

| Role Name | View Project Dashboard and Status | Add/Remove/Access Project Resources | Add/Remove Team Members | Delete Project |
|-------------|-----------------------------------|-------------------------------------|-------------------------|----------------|
| Owner | x | x | x | x |
| Contributor | x | x | | |
| Viewer | x | | | |

- **Owner:** Can add and remove other team members, contribute code to a project repository if the code is stored in AWS CodeCommit, grant or deny other team members remote access to any Amazon EC2 instances running Linux associated with the project, configure the project dashboard, and delete the project.
- **Contributor:** Can add and remove dashboard resources such as a JIRA tile, contribute code to the project repository if the code is stored in AWS CodeCommit, and interact fully with the dashboard. Cannot add or remove team members, grant or deny remote access to resources, or delete the project. This is the role you should choose for most team members.
- **Viewer:** Can view the project dashboard, the code if it is stored in AWS CodeCommit, and, on the dashboard tiles, the state of the project and its resources.

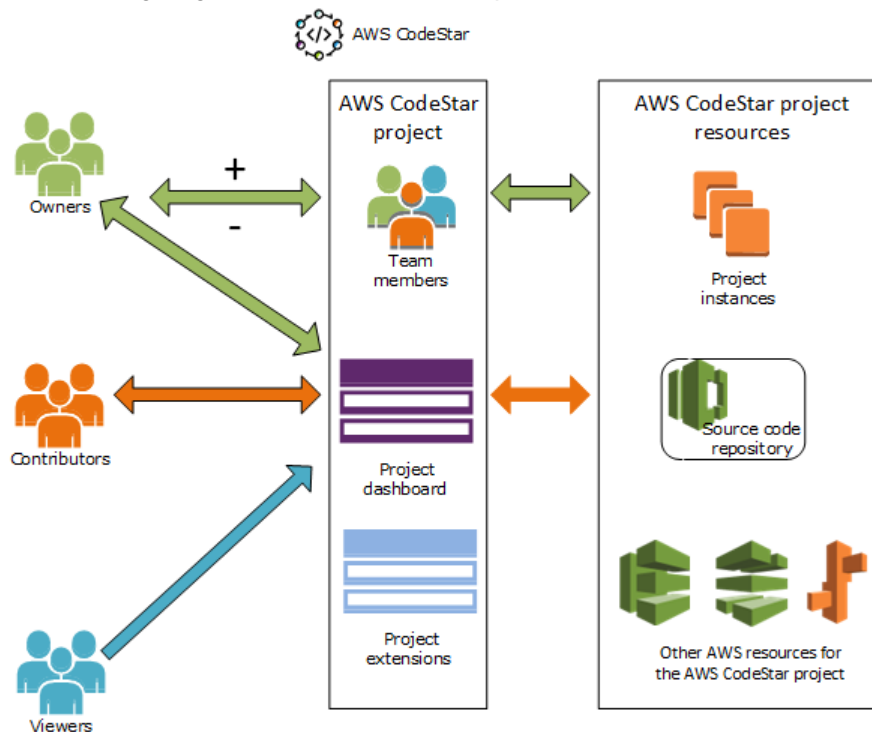
Important

If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, access to those resources are controlled by the resource provider, not AWS CodeStar. For more information, see the resource provider's documentation.

Anyone who has access to an AWS CodeStar project may be able to use the AWS CodeStar console to access resources that are outside of AWS but are related to that project.

An IAM policy is associated with each project role. This policy is customized for your project to reflect its resources. For more information about these policies, see [AWS CodeStar Access Permissions Reference](#) (p. 80).

The following diagram shows the relationship between each role and an AWS CodeStar project.



Topics

- [Add Team Members to an AWS CodeStar Project](#) (p. 63)
- [Manage Permissions for AWS CodeStar Team Members](#) (p. 67)
- [Remove Team Members from an AWS CodeStar Project](#) (p. 70)

Add Team Members to an AWS CodeStar Project

If you have the Owner role in an AWS CodeStar project or have the `AWSCodeStarFullAccess` policy applied to your IAM user, you can add other IAM users to the project team. This is a simple process that applies an AWS CodeStar role (Owner, Contributor, and Viewer) to the user. These roles are per-project and customized. For example, a Contributor team member in ProjectA might have permissions to resources that are different from those of a Contributor team member in ProjectB. A team member can have only one role in a project. After you've added a team member, he or she can interact immediately with your project at the level defined by the role.

Benefits of AWS CodeStar roles and team membership include:

- You do not have to manually configure permissions in IAM for your team members.
- You can easily change a team member's level of access to a project.
- Users can access project dashboards in the AWS CodeStar console only if they are team members.

- User access to a project is defined by the role in that project. A user can have a different role in another project.

For more information about teams and AWS CodeStar roles, see [Working with AWS CodeStar Teams \(p. 62\)](#) and [Working with Your AWS CodeStar User Profile \(p. 72\)](#).

To add a team member to a project, you must either have the AWS CodeStar Owner role for that project or have the **AWSCodeStarFullAccess** policy.

Note

Adding a team member does not affect that team member's access to any resources that are outside of AWS, for example a GitHub repository or issues in Atlassian JIRA. Those access permissions are controlled by the resource provider, not AWS CodeStar. For more information, consult the resource provider's documentation.

Anyone who has access to an AWS CodeStar project may be able to use the AWS CodeStar console to access resources that are outside of AWS but are related to that project.

Topics

- [Add a Team Member Using the AWS CodeStar Console \(p. 64\)](#)
- [Add and View Team Members Using the AWS CLI \(p. 66\)](#)

Add a Team Member Using the AWS CodeStar Console

You can add a team member to your project in the AWS CodeStar console. If an IAM user already exists for the person you want to add, you can add the IAM user directly. If the person does not yet have an IAM user, you can create an IAM user for that person as part of adding them to your project.

To add a team member to an AWS CodeStar project (console)

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Choose the project.

2. In the navigation bar for the project, choose **Team**.
3. On the **Team members** page, choose **Add team member**.
4. In **Choose user**, do one of the following:

- If an IAM user already exists for the person you want to add, choose the IAM user name from the list.

Note

Users who have already been added to another AWS CodeStar project will appear in the **AWS CodeStar users from other projects** list.

On the **Add team member** tab, in **Project role**, choose the AWS CodeStar role (Owner, Contributor, or Viewer) for this user. This is an AWS CodeStar project-level role that can only be changed by an owner of the project. When applied to an IAM user, the role provides all appropriate permissions required to access AWS CodeStar project resources. It applies policies required for creating and managing Git credentials for code stored in AWS CodeCommit in IAM or uploading Amazon EC2 SSH keys for the user in IAM.

Important

You cannot provide or change the display name or email information for an IAM user unless you are signed in to the console as that user. For more information, see [Manage Display Information for Your AWS CodeStar User Profile \(p. 72\)](#).

Choose **Add**.

AWS CodeStar > My First Project > Team

Project Team

Manage users and permissions in your project.

Add team member

Jane Doe ()

Display Name: Jane Doe Email Address: jane.doe@example.com Project Role: Contributor Remote Access: ☒

Add

Team member list (4)

| | Name | Email | Role | Remote Access |
|----|----------------------|-------------------------|-------------|---|
| MM | Mary Major () [You] | mary.major@example.com | Owner | Change Public SSH key Remove Edit |
| JD | John Doe () | john.doe@example.com | Contributor | Remove Edit |
| JS | John Stiles () | john.stiles@example.com | Contributor | Remove Edit |

- If an IAM user does not exist for the person you want to add to the project, choose **Create new IAM user**. Fill in the IAM user name, AWS CodeStar display name, email address, and project role you want to apply to this new user, and choose **Create**.

Create IAM user

Create an IAM user to add to your project.

IAM user name

DemoUser

Display Name

John Doe

Email

john.doe@example.com

Project role

Viewer

Remote access

☐

Cancel

Create

You will be redirected to the IAM console to confirm user creation. Choose **Create user**, save the password information for that new user, and then choose **Close** to return to the AWS CodeStar console. The user will be automatically added to the project with the role you chose.

Note

For ease of management, at least one user should have the **Owner** role for the project.

5. Send the new team member the following information:
 - Connection information for your AWS CodeStar project.
 - If the source code is stored in AWS CodeCommit, [Instructions for setting up access with Git credentials](#) to the AWS CodeCommit repository from their local computers.
 - Information about how the user can manage their display name, email address, and public Amazon EC2 SSH key, as described in [Working with Your AWS CodeStar User Profile](#) (p. 72).
 - One-time password and connection information, if the user is new to AWS and you created an IAM user for that person. The password will expire the first time the user logs on. The user must choose a new password.

Add and View Team Members Using the AWS CLI

You can add team members to your project team using the AWS CLI. You can also view information about all of the team members in your project.

To add a team member (AWS CLI)

1. Open a terminal or command window.
2. Run the **associate-team-member** command, including the `--project-id`, `--user-arn`, and `--project-role` parameters to add a team member to your project with an associated role. You can also specify whether the user has remote access to project instances by including the `--remote-access-allowed` or `--no-remote-access-allowed` parameters. For example:

```
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
```

This command returns no output.

To view all team members (AWS CLI)

1. Open a terminal or command window.
2. Run the **list-team-members** command, including the `--project-id`. For example:

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

This command returns output similar to the following:

```
{
  "teamMembers":[
    {
      "projectRole":"Owner",
      "remoteAccessAllowed":true,
      "userArn":"arn:aws:iam:111111111111:user/
      Mary_Major"},
  ]
}
```

```
{ "projectRole": "Contributor", "remoteAccessAllowed": true, "userArn": "arn:aws:iam::111111111111:user/Jane_Doe" },  
  
{ "projectRole": "Contributor", "remoteAccessAllowed": true, "userArn": "arn:aws:iam::111111111111:user/John_Doe" },  
  
{ "projectRole": "Viewer", "remoteAccessAllowed": false, "userArn": "arn:aws:iam::111111111111:user/John_Stiles" }  
]  
}
```

Manage Permissions for AWS CodeStar Team Members

You change permissions for team members by changing their AWS CodeStar role. Each team member can be assigned to only one role in an AWS CodeStar project, but many users can be assigned to the same role.

Important

To change a role for a user, you must either have the AWS CodeStar Owner role for that project or have the **AWSCodeStarFullAccess** policy applied.

You can also grant permissions for users to remotely access any Amazon EC2 instances running Linux associated with the project. After you grant this permission, the user must upload an SSH public key that will be associated with their AWS CodeStar user profile across all team projects. To successfully connect to the Linux instances associated with the project, the user must have SSH configured and the private key on the local computer.

Note

Changing a team member's permissions does not affect that team member's access to any resources that are outside of AWS, for example a GitHub repository or issues in Atlassian JIRA. Those access permissions are controlled by the resource provider, not AWS CodeStar. For more information, consult the resource provider's documentation.

Anyone who has access to an AWS CodeStar project may be able to use the AWS CodeStar console to access resources that are outside of AWS but are related to that project.

Topics

- [Manage Team Permissions Using the AWS CodeStar Console \(p. 67\)](#)
- [Manage Team Permissions Using the AWS CLI \(p. 69\)](#)

Manage Team Permissions Using the AWS CodeStar Console

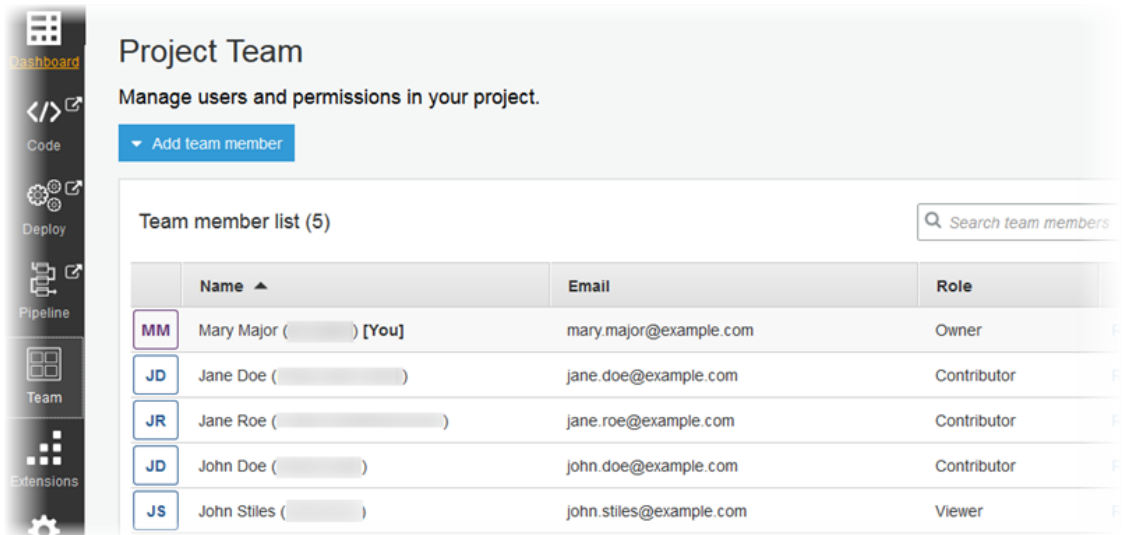
You can manage the roles of team members in the AWS CodeStar console. You can also manage whether team members have remote access to the Amazon EC2 instances associated with your project.

To change the role of a team member (console)

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Choose the project.

2. In the navigation bar for the project, choose **Team**.



3. On the **Team members** page, find the name of the team member, and then choose **Edit**.
4. In **Role**, choose the AWS CodeStar role (Owner, Contributor, or Viewer) you want to apply to this user.

For more information about AWS CodeStar roles and their permissions, see [Working with AWS CodeStar Teams \(p. 62\)](#).

Display name:

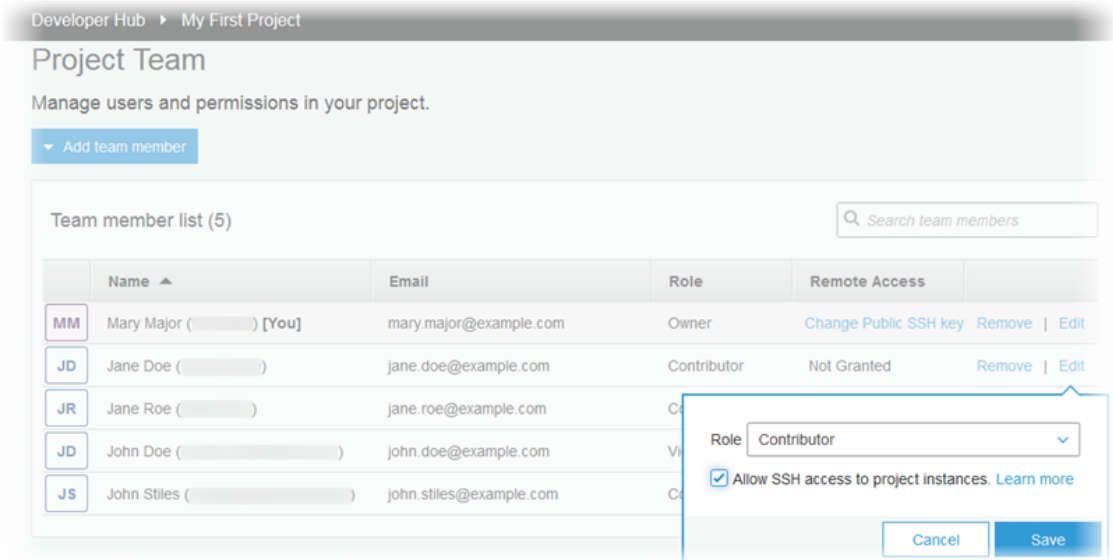
Email:

Role: (dropdown menu showing Owner, Viewer, Contributor, Owner)

Choose **Save**.

To grant a team member remote access permissions to Amazon EC2 instances (console)

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
- Choose the project.
2. In the navigation bar for the project, choose **Team**.
3. On the **Project team** page, find the name of the team member, and then choose **Edit**.
4. Select the **Allow SSH access to project instances** check box, and then choose **Save**.



5. (Optional) Notify the team members that they should upload an SSH public key for their AWS CodeStar users, if they have not already done so. For more information, see [Add a Public Key to Your AWS CodeStar User Profile](#) (p. 76).

Manage Team Permissions Using the AWS CLI

You can use the AWS CLI to manage the project role assigned to a team member. You can use the same AWS CLI commands to manage whether that team member has remote access to Amazon EC2 instances associated with your project.

To manage the permissions for a team member (AWS CLI)

1. Open a terminal or command window.
2. Run the **update-team-member** command, including the `--project-id`, `--user-arn`, and `--project-role` parameters to add a team member to your project with an associated role. You can also specify whether the user has remote access to project instances by including the `--remote-access-allowed` or `--no-remote-access-allowed` parameters. For example, to update the project role of an IAM user named John_Doe and change his permissions to those of a Viewer with no remote access to project Amazon EC2 instances:

```
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
```

This command returns output similar to the following:

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

Remove Team Members from an AWS CodeStar Project

After you remove from an AWS CodeStar project, the user will still appear in the commit history for the project repository, but will no longer have access to the AWS CodeCommit repository or any other project resources, such as the project pipeline. (The exception to this rule is an IAM user who has other policies applied that grant access to those resources.) The user will not be able to access the project dashboard, and the project will no longer appear in the list of projects that user sees on the AWS CodeStar dashboard.

Note

Although removing a team member from a project will deny remote access to project Amazon EC2 instances, it will not close any of the user's active SSH sessions.

Removing a team member does not affect that team member's access to any resources that are outside of AWS, for example a GitHub repository or issues in Atlassian JIRA. Those access permissions are controlled by the resource provider, not AWS CodeStar. For more information, consult the resource provider's documentation.

To remove a team member from a project, you must have the AWS CodeStar Owner role for that project or have the **AWSCodeStarFullAccess** policy applied to your account.

Topics

- [Remove Team Members Using the Console \(p. 70\)](#)
- [Remove Team Members Using the AWS CLI \(p. 70\)](#)

Remove Team Members Using the Console

You can remove team members from your project team using the AWS CodeStar console.

To remove a team member from a project

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.
Choose the project.
2. In the navigation bar for the project, choose **Team**.
3. On the **Team members** page, find the name of the team member you want to remove, and then choose **Remove**.

Remove Team Members Using the AWS CLI

You can remove team members from your project team using the AWS CLI.

To remove a team member (AWS CLI)

1. Open a terminal or command window.
2. Run the **disassociate-team-member** command, including the `--project-id` and `--user-arn` parameters to remove a team member from your project. For example:

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

This command returns output similar to the following:

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

Working with Your AWS CodeStar User Profile

Your AWS CodeStar user profile is associated with your IAM user. This profile contains a display name and email address that is used in all AWS CodeStar projects you belong to. You can upload an SSH public key that will be associated with your profile. This public key is part of the SSH public/private key pair you use when you connect to Amazon EC2 instances associated with AWS CodeStar projects you belong to.

Note

The information in these topics covers only your AWS CodeStar user profile. If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resource providers may use separate user profiles, which may have different settings. For more information, see the resource provider's documentation.

Topics

- [Manage Display Information for Your AWS CodeStar User Profile \(p. 72\)](#)
- [Add a Public Key to Your AWS CodeStar User Profile \(p. 76\)](#)

Manage Display Information for Your AWS CodeStar User Profile

You can change your display name and email information in AWS CodeStar. This information is part of your AWS CodeStar user profile, which is not project-specific, but instead displays in every project you belong to within an AWS region. Because this information is associated with your IAM user, it will be applied across the AWS CodeStar projects you belong to in that region. If you belong to projects in more than one AWS region, you will have a separate user profile in each region.

You can only manage your own user profile in the AWS CodeStar console. If you have the `AWSCodeStarFullAccess` policy, you can view and manage other profiles using the AWS CLI.

Note

The information in this topic covers only your AWS CodeStar user profile. If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resource providers may use separate user profiles, which may have different settings. For more information, see the resource provider's documentation.

Topics

- [Manage Your User Profile Using the AWS CodeStar Console \(p. 73\)](#)
- [Manage User Profiles Using the AWS CLI \(p. 74\)](#)

Manage Your User Profile Using the AWS CodeStar Console

You can manage your user profile in the AWS CodeStar console by navigating to any project where you are a team member and changing your profile information. Because user profiles are user-specific and not project-specific, your user profile changes will appear in every project where you are a team member within an AWS region.

Important

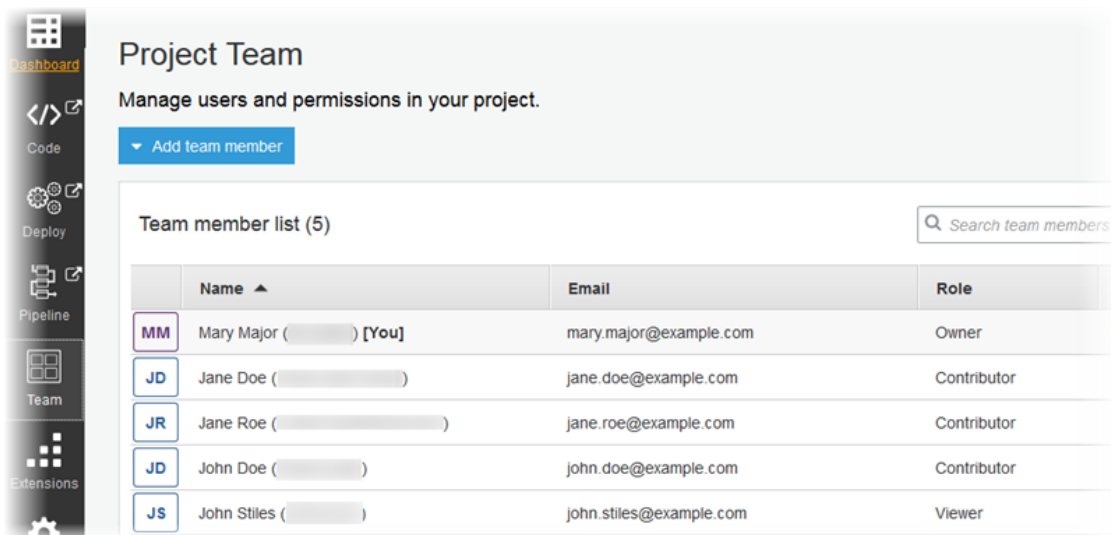
To change the display information for a user in the console, you must be signed in as that IAM user. No other user, even those with AWS CodeStar Owner role for a project or with the `AWSCodeStarFullAccess` policy applied, can change your display information in the console.

To change your display information in all projects within an AWS region (console)

1. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

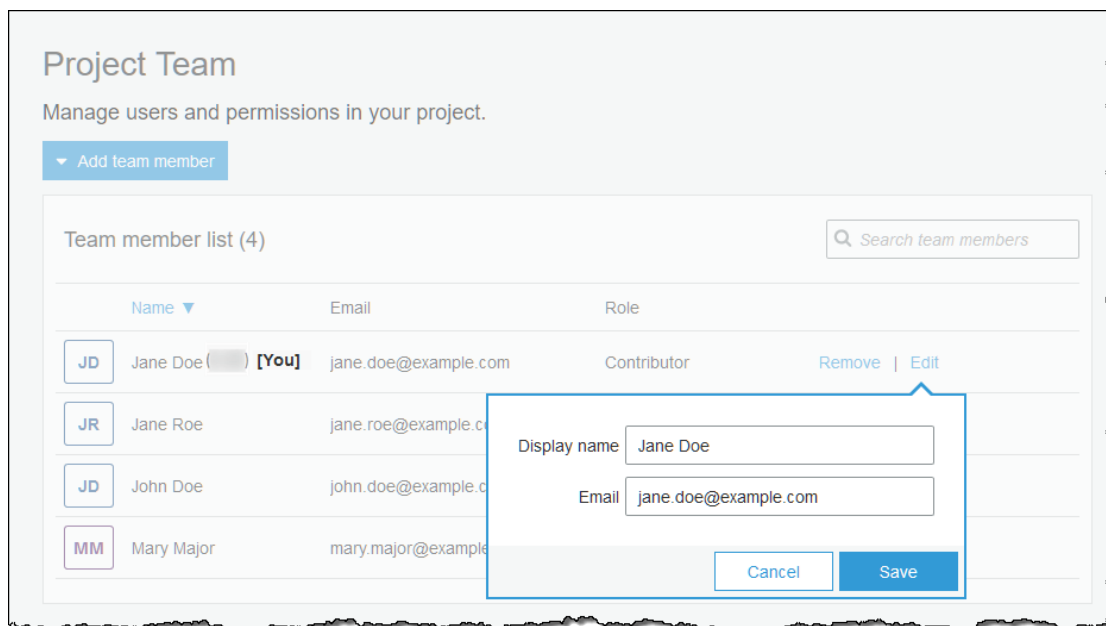
Choose a project where you are a team member.

2. In the navigation bar for the project, choose **Team**.



| | Name ▲ | Email | Role |
|----|----------------------|-------------------------|-------------|
| MM | Mary Major () [You] | mary.major@example.com | Owner |
| JD | Jane Doe () | jane.doe@example.com | Contributor |
| JR | Jane Roe () | jane.roe@example.com | Contributor |
| JD | John Doe () | john.doe@example.com | Contributor |
| JS | John Stiles () | john.stiles@example.com | Viewer |

3. On the **Team members** page, find the name of your IAM user (the team member that has your IAM name in parentheses, and has **[You]** in brackets next to the display name), and then choose **Edit**.



4. Edit the display name, the email address, or both, and then choose **Save**.

Note

Both a display name and an email address are required. For more information, see [Limits in AWS CodeStar \(p. 93\)](#).

Manage User Profiles Using the AWS CLI

You can use the AWS CLI to create and manage your user profile in AWS CodeStar. You can also use the AWS CLI to view your user profile information, and to view all user profiles configured for your AWS account in an AWS region.

Make sure that your AWS profile is configured for the region where you want to create, manage, or view user profiles, as user profiles are region-specific.

To create a user profile (AWS CLI)

1. Open a terminal or command window.
2. Run the **create-user-profile** command, including the **user-arn**, **display-name**, and **email-address** parameters. For example:

```
aws codestar create-user-profile --user-arn arn:aws:iam:11111111111:user/John_Stiles  
--display-name "John Stiles" --email-address "john_stiles@example.com"
```

This command returns output similar to the following:

```
{  
  "createdTimestamp":1.491439687681E9,"  
  displayName:"John Stiles",  
  "emailAddress":"john.stiles@example.com",  
  "lastModifiedTimestamp":1.491439687681E9,  
  "userArn":"arn:aws:iam:11111111111:user/Jane_Doe"  
}
```

To view your display information (AWS CLI)

1. Open a terminal or command window.
2. Run the **describe-user-profile** command, including the `user-arn` parameter. For example:

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

This command returns output similar to the following:

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

To change your display information (AWS CLI)

1. Open a terminal or command window.
2. Run the **update-user-profile** command, including the `user-arn` parameter and the profile parameters you want to change, such as, `display-name` or `email-address` parameters. For example, if a user with the display name "Jane Doe" wanted to change her display name to "Jane Mary Doe":

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

This command returns output similar to the following:

```
{
  "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"
}
```

To list all user profiles in an AWS region in your AWS account (AWS CLI)

1. Open a terminal or command window.
2. Run the **aws codestar list-user-profiles** command. For example:

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

This command returns output similar to the following:

```
{
  "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"
}
]
```

Add a Public Key to Your AWS CodeStar User Profile

You can upload a public SSH key as part of the public/private key pair you create and manage. You use this SSH public/private key pair to access Amazon EC2 instances running Linux. If a project owner has granted you remote access permission, you can access only those instances associated with the project where you were granted access.

Important

To manage an SSH public key associated with your user profile, you must be signed in as that IAM user. No other user, even those with the AWS CodeStar Owner role for a project, can add, remove, or modify your SSH key.

Topics

- [Manage Your Public Key Using the AWS CodeStar Console \(p. 76\)](#)
- [Manage Your Public Key Using the AWS CLI \(p. 78\)](#)
- [Connect to Amazon EC2 Instance with Your Private Key \(p. 78\)](#)

Manage Your Public Key Using the AWS CodeStar Console

Although you cannot generate a public-private key pair in the console, you can create one locally and then add or manage it as part of your user profile through the AWS CodeStar console.

To manage your public SSH key (console)

1. From a terminal or Bash emulator window, run the **ssh-keygen** command to generate an SSH public/private key pair on your local computer. You can generate a key in any format allowed by Amazon EC2. For information about acceptable formats, see [Importing Your Own Public Key to Amazon EC2](#). Ideally, generate a key that is SSH-2 RSA, in OpenSSH format, and contain 2048 bits. The public key is stored in a file with the .pub extension.

2. Open the AWS CodeStar console at <https://console.aws.amazon.com/codestar/>.

Choose a project where you are a team member.

3. In the navigation bar for the project, choose **Team**.
4. On the **Team members** page, find the name of your IAM user (the team member that has your IAM name in parentheses, and has **[You]** in brackets next to the display name), and then choose **Add Public SSH key**.

| | Name ▲ | Email | Role | Remote Access | |
|----|----------------------|-------------------------|-------------|--------------------|---------------|
| MM | Mary Major () [You] | mary.major@example.com | Owner | Add Public SSH key | Remove Edit |
| JD | Jane Doe () | jane.doe@example.com | Contributor | Not Granted | Remove Edit |
| JR | Jane Roe () | jane.roe@example.com | Contributor | Not Granted | Remove Edit |
| JD | John Doe () | john.doe@example.com | Viewer | Not Granted | Remove Edit |
| JS | John Stiles () | john.stiles@example.com | Contributor | Not Granted | Remove Edit |

5. In **Manage your public SSH key**, paste the public key, and then choose **Save**.

Manage your public SSH key

Your membership in this project includes remote access to the Amazon EC2 instances. Before you can access those instances, you must first upload an SSH public key. If you want to change or remove the key, paste in the new key or delete the existing key from the box, and then save your changes. [Learn more](#)

ssh-rsa

Cancel Save

Note

You can change your public key by deleting the old key in this field and pasting in a new one. Similarly, you can delete a public key by deleting the contents of this field and then choosing **Save**.

When you change or delete a public key you are changing your user profile. It is not a per-project change. Because your key is associated with your profile, it will change (or be deleted) in all projects where you have been granted remote access.

Deleting your public key removes your access to Amazon EC2 instances running Linux in all projects where you were granted remote access. However, it does not close any open SSH sessions using that key. Make sure you close any open sessions.

Manage Your Public Key Using the AWS CLI

You can manage your SSH public key as part of your user profile using the AWS CLI. t

To manage your public key (AWS CLI)

1. From a terminal or Bash emulator window, run the **ssh-keygen** command to generate an SSH public/private key pair on your local computer. You can generate a key in any format allowed by Amazon EC2. For information about acceptable formats, see [Importing Your Own Public Key to Amazon EC2](#). Ideally, generate a key that is SSH-2 RSA, in OpenSSH format, and contain 2048 bits. The public key is stored in a file with the .pub extension.
2. Run the **update-user-profile** command with the **--ssh-public-key** parameter to add or change your SSH public key in your AWS CodeStar user profile. For example:

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

This command returns output similar to the following:

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

Connect to Amazon EC2 Instance with Your Private Key

Make sure you have already created an Amazon EC2 key pair. Then add your key pair's public key to your user profile in AWS CodeStar. To create a key pair, see [Step 4: Create an Amazon EC2 Key Pair for AWS CodeStar Projects \(p. 4\)](#) in *Setting Up*. To add your key pair's public key to your user profile, see the instructions earlier in this topic.

To connect to an Amazon EC2 instance running Linux by using your key pair's private key

1. With your project open in the AWS CodeStar console, in the navigation pane, choose **Project**.

2. In **Project Resources**, for the row where **Type** is **Amazon EC2** and **Name** starts with **instance**, choose the **ARN** link.
3. When the Amazon EC2 console displays with the instance details, choose **Connect**.
4. Follow the instructions in the **Connect To Your Instance** dialog.

For the username, use `ubuntu` for projects based on the ASP.NET Core project template, as those instances use Ubuntu. For all other projects, use `ec2-user` for the username, as those instances use Amazon Linux. If you use the wrong username, you will not be able to connect to the instance.

For more information, see the following resources in the *Amazon EC2 User Guide for Linux Instances*.

- [Connecting to Your Linux Instance Using SSH](#)
- [Connecting to Your Linux Instance from Windows Using PuTTY](#)
- [Connecting to Your Linux Instance Using MindTerm](#)

AWS CodeStar Access Permissions Reference

AWS CodeStar uses IAM to control which users have access to AWS CodeStar projects and their resources on your behalf. When you add an IAM user to a project in a project role, policies and roles are applied to that user. These policies and roles grant permissions and access to that user based on the project role. You can use the IAM console to create other policies that assign AWS CodeStar and other permissions to an IAM user.

For example, you might want to allow a user to view but not change an AWS CodeStar project. In this case, you add the IAM user to an AWS CodeStar project with the Viewer role. Every AWS CodeStar project has a set of policies that help you control access to the project. In addition, you can control which users have access to AWS CodeStar.

In the [Setting Up AWS CodeStar \(p. 3\)](#) instructions, you attached a policy named `AWSCodeStarFullAccess` to your IAM user. This policy allows full access to AWS CodeStar. That policy statement looks similar to this:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "CodeStarEC2",
      "Effect": "Allow",
      "Action": [
        "codestar:*",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeSubnets"
      ],
      "Resource": "*"
    },
    {
      "Sid": "CodeStarCF",
      "Effect": "Allow",
      "Action": [
        "cloudformation:DescribeStack*",
        "cloudformation:GetTemplateSummary"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awscodestar-*"
      ]
    }
  ]
}
```

```
}  
  ]  
}
```

This policy statement allows the IAM user to perform all available actions in AWS CodeStar with all available AWS CodeStar resources associated with the AWS account. This includes creating and deleting projects. You might not want to give all IAM users this much access. Instead, you can add IAM users to AWS CodeStar project roles. The roles grant specific levels of access to AWS CodeStar projects.

Topics

- [AWS CodeStar Project-Level Policies and Permissions \(p. 81\)](#)
- [AWS CodeStar Service Role Policy and Permissions \(p. 83\)](#)
- [Attach a Policy to an IAM User \(p. 85\)](#)
- [Action and Resource Syntax \(p. 86\)](#)

AWS CodeStar Project-Level Policies and Permissions

There are three roles in AWS CodeStar projects: Owner, Contributor, and Viewer. Each role is specific to a project and defined by an IAM managed policy, where *project-id* is the ID of the AWS CodeStar project (for example, *my-first-projec*):

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

Important

These policies are subject to change by AWS CodeStar. They should not be modified manually. If you want to add or change permissions, attach additional policies to the IAM user.

When you add a user to a project and choose a role for the user, the appropriate policy is applied automatically to the IAM user. Under most circumstances, you don't need to directly attach or manage policies or permissions in IAM. Manually attaching an AWS CodeStar role policy to an IAM user is not recommended. If absolutely necessary, as a supplement to an AWS CodeStar role policy, you can create your own managed policies to apply your own level of permissions to an IAM user.

Note

The policies for roles in an AWS CodeStar project apply to that project only. This helps ensure that users can only see and interact with the AWS CodeStar projects they have permissions to, at the level determined by their role. Only users who will create AWS CodeStar projects should have a policy applied that allows access to all AWS CodeStar resources, regardless of project.

All AWS CodeStar role policies vary, depending on the AWS resources associated with the project with which the roles are associated. Unlike other AWS services, these policies are customized when the project is created and updated as project resources change. Therefore, there is no one canonical Owner, Contributor, or Viewer managed policy.

AWS CodeStar Owner Role Policy

The CodeStar_*project-id*_Owner managed policy allows a user to perform all actions in the AWS CodeStar project with no restrictions. This is the only policy that allows a user to add or remove team members. Although the contents of the policy vary, depending on the resources associated with the

project, the CodeStar_*project-id*_Owner managed policy contains the following AWS CodeStar permissions. As an AWS managed policy, it is subject to change without notice.

```
"Effect": "Allow",
"Action": [
    ...
    "codestar:*",
    ...
],
"Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project-id"
]
```

An IAM user with this policy can perform all AWS CodeStar actions in the project, but unlike an IAM user with the **AWSCodeStarFullAccess** policy, the user cannot create new projects. The `codestar:*` permission is limited in scope to a specific resource (the AWS CodeStar project associated with that project ID).

AWS CodeStar Contributor Role Policy

The CodeStar_*project-id*_Contributor managed policy allows a user to contribute to the project and change the project dashboard, but does not allow a user to add or remove team members. Although the contents of the policy vary, depending on the resources associated with the project, the CodeStar_*project-id*_Contributor policy contains the following AWS CodeStar permissions. As an AWS managed policy, it is subject to change without notice.

```
"Effect": "Allow",
"Action": [
    ...
    "codestar:DescribeProject",
    "codestar:ListTeamMembers",
    "codestar:*Resource*",
    ...
],
"Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project-id"
]
```

AWS CodeStar Viewer Role Policy

The CodeStar_*project-id*_Viewer managed policy allows a user to view a project in AWS CodeStar, but not change its resources or add or remove team members. Although the contents of the policy vary, depending on the resources associated with the project, the CodeStar_*project-id*_Viewer policy contains the following AWS CodeStar permissions. As an AWS managed policy, it is subject to change without notice.

```
"Effect": "Allow",
"Action": [
    ...
    "codestar:DescribeProject",
    "codestar:ListResources",
    "codestar:ListTeamMembers",
    ...
],
"Resource": [
    "arn:aws:codestar:us-east-2:111111111111:project-id"
]
```

]

AWS CodeStar Service Role Policy and Permissions

AWS CodeStar uses a service role, `aws-codestar-service-role`, when creating and managing the resources for your project.

Important

You must be signed in as an IAM administrative user or root account in order to create this service role.

This role is created the first time you create a project in AWS CodeStar. The service role acts on your behalf to create the resources you choose when creating a project and to display information about those resources in the AWS CodeStar project dashboard. It also acts on your behalf when you manage the resources for a project. It contains the following policy statement:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ProjectStack",
      "Effect": "Allow",
      "Action": [
        "cloudformation:*Stack*",
        "cloudformation:GetTemplate"
      ],
      "Resource": [
        "arn:aws:cloudformation:*:*:stack/awscodestar-*",
        "arn:aws:cloudformation:*:*:stack/awseb-*"
      ]
    },
    {
      "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:::aws-codestar-*/*",
        "arn:aws:s3:::elasticbeanstalk-*",
        "arn:aws:s3:::elasticbeanstalk-*/*"
      ]
    }
  ]
}
```

```
]
},
{
  "Sid": "ProjectServices",
  "Effect": "Allow",
  "Action": [
    "codestar:*Project",
    "codestar:*Resource*",
    "codestar:List*",
    "codestar:Describe*",
    "codestar:Get*",
    "codestar:AssociateTeamMember",
    "codecommit:*",
    "codepipeline:*",
    "codedeploy:*",
    "codebuild:*",
    "ec2:RunInstances",
    "autoscaling:*",
    "cloudwatch:Put*",
    "ec2:*",
    "elasticbeanstalk:*",
    "elasticloadbalancing:*",
    "iam:ListRoles",
    "logs:*",
    "sns:*"
  ],
  "Resource": "*"
},
{
  "Sid": "ProjectWorkerRoles",
  "Effect": "Allow",
  "Action": [
    "iam:AttachRolePolicy",
    "iam:CreateRole",
    "iam>DeleteRole",
    "iam>DeleteRolePolicy",
    "iam:DetachRolePolicy",
    "iam:GetRole",
    "iam:PassRole",
    "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"
  ],
  "Resource": [
    "arn:aws:iam:*:policy/CodeStar_*"
  ]
},
{
  "Sid": "InspectServiceRole",
  "Effect": "Allow",
  "Action": [
    "iam:ListAttachedRolePolicies"
  ],
  "Resource": [
    "arn:aws:iam:*:role/aws-codestar-service-role"
  ]
}
]
```

Attach a Policy to an IAM User

When you add a user to a project, AWS CodeStar automatically attaches the managed policy for the project that matches the user's role. You should not manually attach an AWS CodeStar managed policy for a project to an IAM user. With the exception of `AWSCodestarFullAccess`, we do not recommend that you attach policies that change an IAM user's permissions in an AWS CodeStar project. If you decide to create and attach your own policies, do the following:

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the IAM console, in the navigation pane, choose **Users**, and then choose the user to which you want to attach additional policies.
3. On the **Permissions** tab, choose **Add permissions**. Choose **Attach existing policies directly**, select the policy you want to apply, and then choose **Attach Policy**.

For example, if you want to add your own customized policy to a user, choose the policy name from the list of policies.

4. If you do not want to attach an existing policy but instead want to create your own custom policy, on the **Permissions** tab, choose **Add inline policy**. Choose **Custom Policy**, and then choose **Select**.

In **Policy Name**, type a name for this policy. In the **Policy Document** box, type a policy that follows this format, and then choose **Apply Policy**:

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
```

```
    "Effect" : "Allow",
    "Action" : [
      "action-statement"
    ],
    "Resource" : [
      "resource-statement"
    ]
  },
  {
    "Effect" : "Allow",
    "Action" : [
      "action-statement"
    ],
    "Resource" : [
      "resource-statement"
    ]
  }
]
```

In the preceding statement, for *action-statement* and *resource-statement*, specify the AWS CodeStar actions and resources the IAM user is allowed to perform or access. (By default, the IAM user does not have permissions unless a corresponding `Allow` statement is explicitly stated. If you want to specifically deny a permission granted by another policy, such as the policy for an AWS CodeStar role, choose `Deny` instead of `Allow`.) You can add statements as needed. The following sections describe the format of allowed actions and resources for AWS CodeStar. Syntax examples are provided in these sections.

Action and Resource Syntax

The following sections describe the format for specifying actions and resources.

Actions follow this general format:

```
codestar:action
```

Where *action* is an available AWS CodeStar operation, such as `ListProjects` or `AssociateResource`. To allow an action, use the `"Effect" : "Allow"` clause. To explicitly deny an action, use the `"Effect" : "Deny"` clause. By default, all actions are denied, unless specified otherwise in any other attached policy.

Resources follow this general format:

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

Where *region* is a target region (such as `us-east-2`), *account* is the AWS account ID, and *resource-specifier* is the project ID. Wildcard (*) characters can be used to specify a partial name.

For example, the following specifies the AWS CodeStar project named *my-first-projec* registered to the AWS account `111111111111` in the region `us-east-2`:

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

The following specifies any AWS CodeStar project that begins with the name `my-proj` registered to the AWS account `111111111111` in the region `us-east-2`:

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

Topics

- [Resource Scoping in AWS CodeStar \(p. 87\)](#)
- [Projects \(p. 87\)](#)
- [Resources \(p. 87\)](#)
- [Teams \(p. 88\)](#)
- [Users \(p. 88\)](#)

Resource Scoping in AWS CodeStar

Some of the permissions in AWS CodeStar cannot be scoped to a resource, but instead must be scoped to all, or the action will fail.

The following action cannot be scoped. It must be set to *:

- ListProjects

Projects

Allowed actions include:

- CreateProject to create an AWS CodeStar project.
- DeleteProject to delete an AWS CodeStar project.
- DescribeProject to describe the attributes of an AWS CodeStar project.
- ListProjects to list all the AWS CodeStar projects.
- UpdateProject to update the attributes of an AWS CodeStar project.

The following example allows a specified IAM user to edit the attributes of an AWS CodeStar project, such as its project description:

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

Resources

Allowed actions include:

- ListResources to list all the resources for an AWS CodeStar project.

The following example allows an IAM user who has this policy attached to list resources for a project with the ID *my-first-projec*:

```
{
```

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

Teams

Allowed actions include:

- `AssociateTeamMember` to add a user to an AWS CodeStar project.
- `DisassociateTeamMember` to remove a user from an AWS CodeStar project.
- `ListTeamMembers` to list all the users in an AWS CodeStar project.
- `UpdateTeamMember` to change the team member's attributes in a AWS CodeStar project (for example, the user's project role).

The following example allows an IAM user who has this policy attached to add team members to an AWS CodeStar project with the project ID *my-first-projec*, but explicitly denies that user the ability to remove team members:

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

Users

Allowed actions include:

- `CreateUserProfile` to create a user profile in AWS CodeStar. This profile contains data associated with the user, such as a display name, that appears across all AWS CodeStar projects.
- `DeleteUserProfile` to delete an AWS CodeStar user profile.
- `DescribeUserProfile` to retrieve information about an AWS CodeStar user profile.
- `ListUserProfiles` to list all AWS CodeStar user profiles for an AWS account.

- `UpdateUserProfile` to update an AWS CodeStar profile for a user.

The following example allows an IAM user who has this policy attached to list all AWS CodeStar user profiles associated with an AWS account:

```
{
  "Version": "2012-10-17",
  "Statement" : [
    {
      "Effect" : "Allow",
      "Action" : [
        "codestar:ListUserProfiles",
      ],
      "Resource" : "*"
    }
  ]
}
```

Logging AWS CodeStar API Calls with AWS CloudTrail

AWS CodeStar is integrated with CloudTrail, a service that captures API calls made by or on behalf of AWS CodeStar in your AWS account and delivers the log files to an Amazon S3 bucket you specify. CloudTrail captures API calls from the AWS CodeStar console, the AWS CLI, the AWS SDKs, and the AWS CodeStar HTTP API. Using the information collected by CloudTrail, you can determine which request was made to AWS CodeStar, the source IP address from which the request was made, who made the request, when it was made, and so on. To learn more about CloudTrail, including how to configure and enable it, see the [AWS CloudTrail User Guide](#).

AWS CodeStar Information in CloudTrail

When CloudTrail logging is enabled in your AWS account, calls made to AWS CodeStar actions are tracked in log files. AWS CodeStar records are written together with other AWS service records in a log file. CloudTrail determines when to create and write to a new file based on a time period and file size.

All of the AWS CodeStar actions are logged. These actions are documented in the [AWS CodeStar API Reference](#).

Every log entry contains information about who generated the request. The user identity information in the log helps you determine whether the request was made with root or IAM user credentials, with temporary security credentials for a role or federated user, or by another AWS service. For more information, see the `userIdentity` field in the [CloudTrail Event Reference](#).

You can store your log files in your bucket for as long as you want, but you can also define Amazon S3 lifecycle rules to archive or delete log files automatically. By default, Amazon S3 server-side encryption (SSE) is used to encrypt your log files.

You can have CloudTrail publish Amazon SNS notifications when new log files are delivered. For more information, see [Configuring Amazon SNS Notifications for CloudTrail](#).

You can also aggregate AWS CodeStar log files from multiple AWS regions and multiple AWS accounts into a single Amazon S3 bucket. For more information, see [Receiving CloudTrail Log Files from Multiple Regions](#).

Understanding AWS CodeStar Log File Entries

CloudTrail log files can contain one or more log entries where each entry is made up of multiple JSON-formatted events. A log entry represents a single request from any source and includes information about the requested action, any parameters, the date and time of the action, and so on. The log entries are not guaranteed to be in any particular order. That is, they are not an ordered stack trace of the public calls.

The following example shows a CloudTrail log entry that demonstrates a `CreateProject` operation being called in 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"
}
```

```
}
```


Limits in AWS CodeStar

The following table describes limits in AWS CodeStar. AWS CodeStar depends on other AWS services for project resources. Some of those service limits can be changed. For information about limits that can be changed, see [AWS Service Limits](#).

| | |
|---|---|
| Number of projects | Maximum of 333 projects in an AWS account. Actual limit will vary depending on the level of other service dependencies, for example the maximum number of pipelines in AWS CodePipeline allowed for your AWS account. |
| Number of AWS CodeStar projects to which an IAM user can belong | Maximum of 10 per individual IAM user. |
| Project IDs | <p>Project IDs must be unique within an AWS account. Project IDs must be at least 2 characters and cannot exceed 15 characters. Allowed characters include:</p> <p>Letters a through z, inclusive.</p> <p>Numbers 0 through 9, inclusive.</p> <p>The special character - (minus sign).</p> <p>Any other characters, such as capital letters, spaces, . (period), @ (at sign), or _ (underscore), are not allowed.</p> |
| Project names | Project names cannot exceed 100 characters in length, and cannot begin or end with an empty space. |
| Project descriptions | Any combination of characters between 0 and 1,024 characters in length. Project descriptions are optional. |
| Team members in an AWS CodeStar project | 100 |

| | |
|--|--|
| Display name in a user profile | Any combination of characters between 1 and 100 characters in length. Display names must include at least one character. That character cannot be a space. Display names cannot begin or end with a space. |
| Email address in a user profile | The email address must include an @ and end in a valid domain extension. |
| Federated access, root account access, or temporary access to AWS CodeStar | AWS CodeStar only supports IAM users. AWS CodeStar does not support federated users or temporary access credentials. Using AWS CodeStar with a root account is not recommended. |
| IAM roles | A maximum of 5,120 characters in any managed policy that is attached to an IAM role. |

Troubleshooting AWS CodeStar

The following information might help you troubleshoot common issues in AWS CodeStar.

Topics

- [Project creation failure: A project was not created \(p. 95\)](#)
- [Project creation: I see an error when I try to edit Amazon EC2 configuration when creating a project \(p. 96\)](#)
- [Project deletion: An AWS CodeStar project was deleted, but resources still exist \(p. 96\)](#)
- [Team management failure: An IAM user could not be added to a team in an AWS CodeStar project \(p. 97\)](#)
- [Service role issue: The service role could not be created \(p. 98\)](#)
- [Service role issue: The service role is not valid or missing \(p. 98\)](#)
- [Project role issue: AWS Elastic Beanstalk health status checks fail for instances in an AWS CodeStar project \(p. 98\)](#)
- [Project role issue: A project role is not valid or missing \(p. 99\)](#)
- [Project extensions: Can't connect to JIRA \(p. 99\)](#)
- [GitHub: Can't access a repository's commit history, issues, or code \(p. 99\)](#)

Project creation failure: A project was not created

Problem: When you try to create a project, you see a message that the creation failed.

Possible fixes: The most common reasons for failure are:

- A project with that ID already exists in your AWS account, possibly in a different region.
- The IAM user you used to sign in to the AWS Management Console does not have the permissions required to create a project.
- The AWS CodeStar service role is missing one or more required permissions.

- You have reached the maximum limit for one or more resources for a project (such as the limit on customer managed policies in IAM, Amazon S3 buckets, or pipelines in AWS CodePipeline).

Before you create a project, verify that you have the **AWSCodeStarFullAccess** policy applied to your IAM user. For more information, see [AWS CodeStar Access Permissions Reference \(p. 80\)](#).

When you create a project, make sure that the ID is unique and meets the AWS CodeStar requirements. Be sure you selected the **AWS CodeStar would like permission to administer AWS resources on your behalf** check box.

To troubleshoot other issues, open the AWS CloudFormation console, choose the stack for the project you tried to create, and choose the **Events** tab. There might be more than one stack for a project. The stack names will start with `awscodestar-`, followed by the project ID. Stacks might be under the **Deleted** filter view. Review any failure messages in the stack events and correct the issue listed as the cause of those failures.

Project creation: I see an error when I try to edit Amazon EC2 configuration when creating a project

Problem: When you edit the Amazon EC2 configuration options during project creation, you see an error message or greyed-out option, and cannot continue with project creation.

Possible fixes: The most common reasons for an error message are:

- The VPC in the AWS CodeStar project template (either the default VPC, or the one used when the Amazon EC2 configuration was edited) has dedicated instance tenancy, and the instance type is not supported for dedicated instances. Either choose a different instance type or a different Amazon VPC.
- Your AWS account has no Amazon VPCs. You might have deleted the default VPC, and not created any others. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>, choose **Your VPCs**, and make sure that you have at least one VPC configured. If not, create one. For more information, see [Amazon Virtual Private Cloud Overview](#) in the *Amazon VPC Getting Started Guide*.
- The Amazon VPC does not have any subnets. Choose a different VPC, or create a subnet for that VPC. For more information, see [VPC and Subnet Basics](#).

Project deletion: An AWS CodeStar project was deleted, but resources still exist

Problem: An AWS CodeStar project was deleted, but resources created for that project still exist. By default, AWS CodeStar deletes project resources when the project is deleted. Some resources, such as Amazon S3 buckets, are retained even if the user selects the **Delete associated resources along with AWS CodeStar project** check box, as the buckets might contain data.

Possible fixes: Open the [AWS CloudFormation console](#) and find one or more of the AWS CloudFormation stacks used to create the project. The stack names will start with `awscodestar-`, followed by the project ID. The stacks might be under the **Deleted** filter view. Review the events associated with the stack to discover the resources created for the project. Open the console for each of those resources in the region where you created the AWS CodeStar project, and then manually delete the resources for that deleted project.

Project resources that might remain include:

- One or more project buckets in Amazon S3. Unlike other project resources, project buckets in Amazon S3 are not deleted when the **Delete associated AWS resources along with AWS CodeStar project** check box is cleared.

Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.

- A source repository for your project in AWS CodeCommit.

Open the AWS CodeCommit console at <https://console.aws.amazon.com/codecommit/>.

- A pipeline for your project in AWS CodePipeline.

Open the AWS CodePipeline console at <https://console.aws.amazon.com/codepipeline/>.

- An application and associated deployment groups in AWS CodeDeploy.

Open the AWS CodeDeploy console at <https://console.aws.amazon.com/codedeploy/>.

- An application and associated environments in AWS Elastic Beanstalk.

Open the Elastic Beanstalk console at <https://console.aws.amazon.com/elasticbeanstalk/>.

- A function in AWS Lambda.

Open the AWS Lambda console at <https://console.aws.amazon.com/lambda/>.

- One or more APIs in API Gateway.

Open the API Gateway console at <https://console.aws.amazon.com/apigateway/>.

- One or more IAM policies or roles in IAM.

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

- An instance in Amazon Elastic Compute Cloud (Amazon EC2).

Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

If your project uses resources outside of AWS, for example a GitHub repository or issues in Atlassian JIRA, those resources are not deleted, even if the **Delete associated AWS resources along with CodeStar project** box is selected.

Team management failure: An IAM user could not be added to a team in an AWS CodeStar project

Problem: When you try to add a user to a project, you see an error message saying that the addition failed.

Possible fixes: The most common reason for this error is that the IAM user has reached the limit of managed policies that can be applied to a user in IAM. You might also receive this error if you are not a member of the Owner role in the AWS CodeStar project where you tried to add the user, or if the IAM user does not exist or was deleted.

Make sure you are signed in as an IAM user who is an owner in that AWS CodeStar project. For more information, see [Add Team Members to an AWS CodeStar Project](#) (p. 63).

To troubleshoot other issues, open the IAM console, choose the user you tried to add, and check how many managed policies are applied to that IAM user.

For more information, see [Limitations on IAM Entities and Objects](#). For limits that can be changed, see [AWS Service Limits](#).

Service role issue: The service role could not be created

Problem: When you try to create a project in AWS CodeStar, you see a message prompting you to create the service role. When you choose the option to create it, you see an error.

Possible fixes: The most common reason for this error is that you are logged into AWS with an account that does not have sufficient permissions to create the service role. To create the AWS CodeStar service role (`aws-codestar-service-role`), you must be logged in as an administrative user or with a root account. Log out of the console, and log back in with an IAM user that has the `AdministratorAccess` managed policy applied.

Service role issue: The service role is not valid or missing

Problem: When you open the AWS CodeStar console, you see a message indicating the AWS CodeStar service role is missing or not valid.

Possible fixes: The most common reason for this error is that an administrative user edited or deleted the service role (`aws-codestar-service-role`). If the service role was deleted, you will be prompted to create it. You must be logged in as an administrative user or with a root account to create the role. If the role was edited, it is no longer valid. Log in to the IAM console as an administrative user, find the service role in the list of roles, and delete it. Switch to the AWS CodeStar console and follow the on-screen instructions to create the service role.

Project role issue: AWS Elastic Beanstalk health status checks fail for instances in an AWS CodeStar project

Problem: If you created an AWS CodeStar project that includes Elastic Beanstalk before September 22, 2017, Elastic Beanstalk health status checks might fail. If you have not changed the Elastic Beanstalk configuration since you created the project, the health status check fails and reports a grey state. Despite the health check failure, your application should still run as expected. If you changed the Elastic Beanstalk configuration since you created the project, the health status check fails, and your application might not run correctly.

Fix: One or more IAM roles are missing required IAM policy statements. Add the missing policies to the affected roles in your AWS account.

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

(If you cannot do this, see your AWS account administrator for assistance.)

2. In the navigation pane, choose **Roles**.
3. In the list of roles, choose **CodeStarWorker-*Project-ID*-EB**, where *Project-ID* is the ID of one of the affected projects. (If you cannot easily find a role in the list, type some or all of the role's name in the **Search** box.)

4. On the **Permissions** tab, choose **Attach Policy**.
5. In the list of policies, select **AWSElasticBeanstalkEnhancedHealth** and **AWSElasticBeanstalkService**. (If you cannot easily find a policy in the list, type some or all of the policy's name in the **Search** box.)
6. Choose **Attach Policy**.
7. Repeat steps 3 through 6 for each affected role that has a name following the pattern **CodeStarWorker-*Project-ID*-EB**.

Project role issue: A project role is not valid or missing

Problem: When you try to add a user to a project, you see an error message saying that the addition failed because the policy for a project role is either missing or not valid.

Possible fixes: The most common reason for this error is that one or more project policies was edited in or deleted from IAM. Project policies are unique to AWS CodeStar projects, and cannot be recreated. The project cannot be used. Create a new project in AWS CodeStar, and migrate data to the new project. Clone project code from the unusable project's repository, and push that code to the new project's repository. Copy team wiki information from the old project to the new project. Add users to the new project. When you are sure you have migrated all data and settings, delete the unusable project.

Project extensions: Can't connect to JIRA

Problem: When you use the **Atlassian JIRA** extension to try to connect an AWS CodeStar project to a JIRA instance, the following message displays: "The URL is not a valid JIRA URL. Verify that the URL is correct."

Possible fixes:

- Make sure the JIRA URL is correct, and then try connecting again.
- Your self-hosted JIRA instance may not be accessible from the public Internet. Contact your network administrator to make sure your JIRA instance can be accessed from the public Internet, and then try connecting again.

GitHub: Can't access a repository's commit history, issues, or code

Problem: In the dashboard for a project that stores its code in GitHub, the **Commit history** and **GitHub Issues** tiles display a connection error, or choosing **Open in GitHub** or **Create issue** in these tiles displays an error.

Possible causes:

- The AWS CodeStar project may no longer have access to the GitHub repository.
- The repository may have been deleted or renamed in GitHub.

AWS CodeStar User Guide History

The following table describes important changes to the documentation since the last release of the *AWS CodeStar User Guide*.

- **Latest documentation update:** October 12, 2017

| Change | Description | Date Changed |
|---|---|------------------|
| AWS CodeStar supports GitHub | AWS CodeStar now supports storing project code in GitHub. For more information, see Create a Project (p. 46) . | October 12, 2017 |
| AWS CodeStar available in US West (N. California) and EU (London) | AWS CodeStar is now available in the US West (N. California) and EU (London) regions. For more information, see AWS CodeStar in the <i>Amazon Web Services General Reference</i> . | August 17, 2017 |
| AWS CodeStar available in Asia Pacific (Sydney), Asia Pacific (Singapore), and EU (Frankfurt) | AWS CodeStar is now available in the Asia Pacific (Sydney), Asia Pacific (Singapore), and EU (Frankfurt) regions. For more information, see AWS CodeStar in the <i>Amazon Web Services General Reference</i> . | July 25, 2017 |
| AWS CloudTrail supports AWS CodeStar | AWS CodeStar is now integrated with CloudTrail, a service that captures API calls made by or on behalf of AWS CodeStar in your AWS account and delivers the log files to an Amazon S3 bucket you specify. For more information, see Logging API Calls with CloudTrail (p. 90) . | June 14, 2017 |
| Initial release | This is the first release of the <i>AWS CodeStar User Guide</i> . | April 19, 2017 |

AWS Glossary

For the latest AWS terminology, see the [AWS Glossary](#) in the *AWS General Reference*.