

# AWS CodePipeline Permission Access - Setup Guide

## Overview

AWS CodePipeline does not have a built-in permission management feature. Instead, access control is managed through AWS IAM (Identity and Access Management).

This guide covers how to create IAM users with limited access for developers, allowing them to:

- |  <b>Allowed:</b>   |  <b>Not Allowed:</b>               |
|---|---|
| <ul style="list-style-type: none"><li>• View pipeline status</li><li>• View execution logs</li><li>• Re-trigger pipeline (Release change)</li><li>• Retry failed stage</li><li>• Stop execution</li></ul> | <ul style="list-style-type: none"><li>• Create pipeline</li><li>• Edit pipeline</li><li>• Delete pipeline</li></ul> |

[Overview](#)

[Step 1: Create IAM Policy](#)

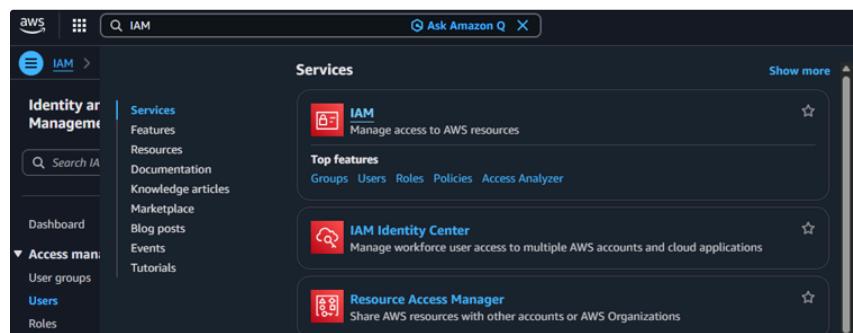
[Step 2: Create User Group](#)

[Step 3: Create First User](#)

[Step 4: Share Login Info with Developer](#)

## Step 1: Create IAM Policy

-  A policy defines what actions are allowed. In this step, we create a custom policy that grants limited CodePipeline access.



Search/Access 'IAM' service in AWS

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with a search bar and a menu. The main area has a search bar at the top and a table titled "Policies (1444)". The table lists various policies with icons and links. A tooltip at the bottom of the sidebar says "Open 'Policies' on the left side of the menu".

Policy name
<a href="#">AccessAnalyzerServiceRolePolicy</a>
<a href="#">AccountManagementFromVercel</a>
<a href="#">AdministratorAccess</a>
<a href="#">AdministratorAccess-Amplify</a>
<a href="#">AdministratorAccess-AWSElasticBeanstalk</a>
<a href="#">AIOpsAssistantIncidentReportPolicy</a>
<a href="#">AIOpsAssistantPolicy</a>
<a href="#">AIOpenSourceAdminPolicy</a>

Open 'Policies' on the left side of the menu

This screenshot shows the "Create policy" page. It features a "Description" field containing "Allow Access Analyzer to analyze resou..." and a note below it: "For use with accounts created through...". There are buttons for "Actions", "Delete", and "Create policy". A tooltip at the bottom says "Click 'Create policy' button to create a collection of limited access for developers".

Click 'Create policy' button to create a collection of limited access for developers

This screenshot shows the "Specify permissions" step of the policy creation wizard. It includes a "Policy editor" with JSON code, a "Visual" tab, and a "Statement" section. A tooltip at the bottom says "Click 'JSON' Button to open JSON Editor".

```
1: {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:123456789012:function:my-lambda"
    }
  ]
}
```

Click 'JSON' Button to open JSON Editor

### Policy editor

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "codepipeline:GetPipeline",  
8                 "codepipeline:GetPipelineState",  
9                 "codepipeline:GetPipelineExecution",  
10                "codepipeline>ListPipelineExecutions",  
11                "codepipeline>ListActionExecutions",  
12                "codepipeline>ListActionTypes",  
13                "codepipeline>ListPipelines",  
14                "codepipeline>ListTagsForResource",  
15                "codepipeline>ListRuleExecutions",  
16                "codepipelineStartPipelineExecution",  
17                "codepipelineStopPipelineExecution",  
18                "codepipelineRetryStageExecution"  
19            ],  
20            "Resource": "*"  
21        }  
22    ]  
23}
```

How it should look after applying the JSON file provided below

```
1 {  
2     "Version": "2012-10-17",  
3     "Statement": [  
4         {  
5             "Effect": "Allow",  
6             "Action": [  
7                 "codepipeline:GetPipeline",  
8                 "codepipeline:GetPipelineState",  
9                 "codepipeline:GetPipelineExecution",  
10                "codepipeline>ListPipelineExecutions",  
11                "codepipeline>ListActionExecutions",  
12                "codepipeline>ListActionTypes",  
13                "codepipeline>ListPipelines",  
14                "codepipeline>ListTagsForResource",  
15                "codepipeline>ListRuleExecutions",  
16                "codepipelineStartPipelineExecution",  
17                "codepipelineStopPipelineExecution",  
18                "codepipelineRetryStageExecution"  
19            ],  
20            "Resource": "*"  
21        }  
22    ]  
23}
```

**Note:** Below is a brief explanation of each permission included in this policy.

Permission	Description
GetPipeline	View pipeline configuration
GetPipelineState	View current pipeline status
GetPipelineExecution	View execution details

ListPipelineExecutions	View execution history
ListActionExecutions	View action-level execution details
ListActionTypes	View available action types
ListPipelines	View list of all pipelines
ListTagsForResource	View pipeline tags
ListRuleExecutions	View rule execution details
StartPipelineExecution	Re-trigger pipeline
StopPipelineExecution	Stop running pipeline
RetryStageExecution	Retry failed stage

5688 of 6144 characters remaining

Cancel Next

Click 'Next' Button

Review and create Info  
Review the permissions, specify details, and tags.

**Policy details**

**Policy name**  
Enter a meaningful name to identify this policy.  
  
Maximum 128 characters. Use alphanumeric and '+-,@\_,-' characters.

**Description - optional**  
Add a short explanation for this policy.  
  
Maximum 1,000 characters. Use alphanumeric and '+-,@\_,-' characters.

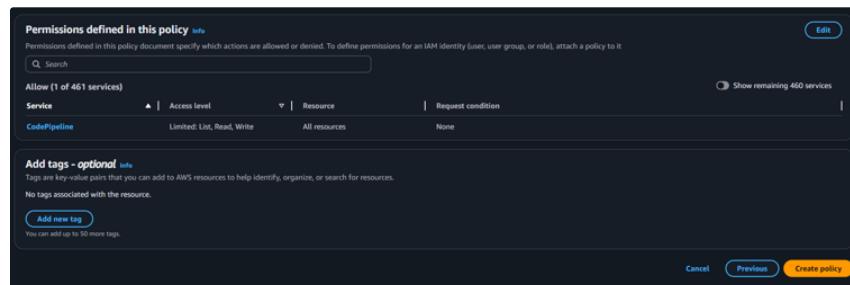
**Permissions defined in this policy** Info  
Permissions defined in this policy document specify which actions are allowed or denied. To define permissions for an IAM identity (user, user group, role, or instance profile), use the IAM console or AWS CloudFormation.

Search

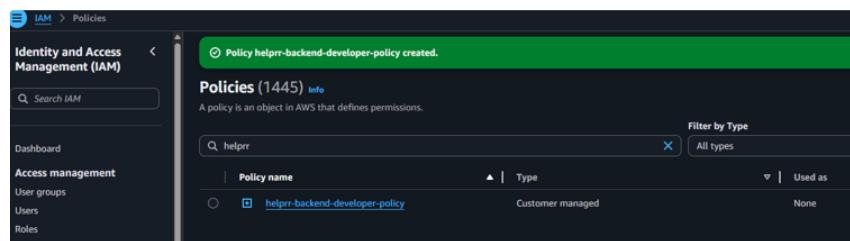
Fills up the Policy name and description

- Example:

- Policy name: `helper-backend-developer-policy`
- description: Limited access for backend developers to view, re-trigger, and stop pipelines



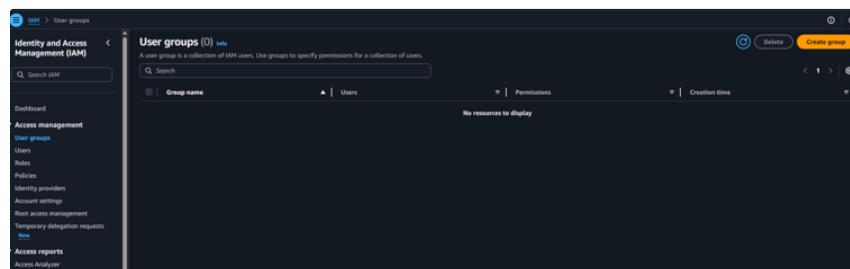
Confirm if 'CodePipeline' Service in the list and Click 'Create policy' button



The created success toast message showed and search the policy name to see whether it is under policy list

## Step 2: Create User Group

- ⓘ A user group allows you to **manage permissions for multiple users at once**. Instead of assigning the policy to each user individually, we attach it to a group. Any user added to this group will automatically have the same permissions.

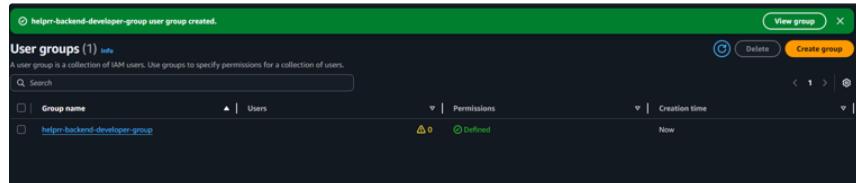


Open 'User groups' on the left side of the menu and Click 'Create group' Button

Fills up the 'User group name' and Select the custom policy created in previous step, and then Click the 'Create user group' button

- Example:

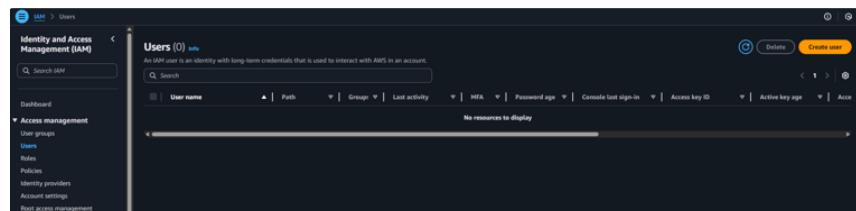
- group name: `helper-backend-developer-group`
- Attach permission policies: `helper-backend-developer-policy`



The success message will appear and the group will be shown in the list

## Step 3: Create First User

- Info** Now we create an IAM user for the developer. This user will be added to the group we just created, which automatically grants them the limited pipeline access.



Open 'User' on the left side of the menu and Click 'Create user' Button

The screenshot shows the 'Specify user details' step of the AWS IAM 'Create user' wizard. The 'User name' field is filled with 'Jin'. The 'Provide user access to the AWS Management Console - optional' checkbox is checked. Other sections include 'Console password' (set to 'Autogenerated password'), 'Review password' (checkbox checked), and a note about generating programmatic access keys.

Fills up the 'User name' form and check 'Provide user access to the AWS Management Console - optional', And then Click 'Next' Button

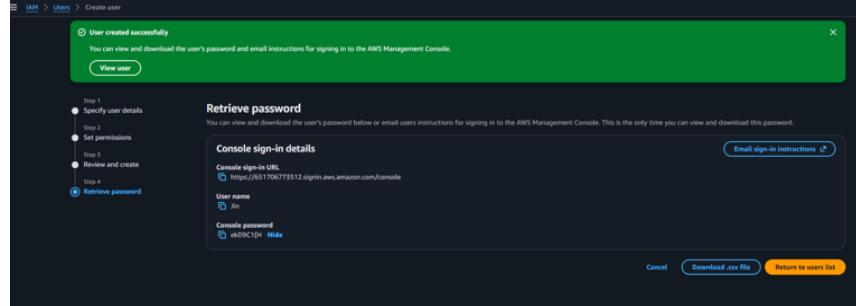
**⚠ Important:** You must check "Provide user access to the AWS Management Console". Without this, the developer cannot log in to AWS Console to view pipelines.

The screenshot shows the 'Set permissions' step of the AWS IAM 'Create user' wizard. Under 'Permissions options', 'Add user to group' is selected. In the 'User groups' table, there is one entry: 'helper-backend-developer-group' with the attached policy 'helper-backend-developer-policy'. At the bottom, there is a 'Set permissions boundary - optional' section.

Check the user group created in the previous step, and then Click the 'Next' Button

The screenshot shows the 'Review and create' step of the AWS IAM 'Create user' wizard. It displays the user details ('User name: Jin', 'Console password type: Autogenerated', 'Require password reset: Yes'), a permissions summary table, and a tags section indicating 'No tags associated with the resource'.

Review the settings and Click the 'Create user' Button



The success message will appear in green

**⚠ Important:** This is **the only time** you can view the password. Make sure to copy it before closing this page.

## Step 4: Share Login Info with Developer

After creating the user, you will see the login credentials. Share the following information with the developer:

- **Console sign-in URL:** The URL to access AWS Console
- **User name:** The username created
- **Console password:** The temporary password