



# Amazon CodeCatalyst

## Issues 編

江口 昌宏

Solutions Architect  
2023/12

# 自己紹介

名前：江口 昌宏

所属：アマゾン ウェブ サービス ジャパン合同会社

DNB ソリューション本部

ISV/SaaS ソリューショングループ

経歴：メガベンチャーの Java エンジニア

好きな AWS サービス：



AWS Lambda



# AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - <https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/>
  - <https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY>



ご感想は X (Twitter) へ！ハッシュタグは以下をご利用ください  
#awsblackbelt

# 内容についての注意点

- 本資料では資料作成時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (<https://aws.amazon.com/>) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっております。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の [AWS サポート窓口](#)へお問い合わせください
- 料金面でのお問い合わせに関しましては、[カスタマーサポート窓口](#)へお問い合わせください (マネジメントコンソールへのログインが必要です)

# Black Belt Amazon CodeCatalyst シリーズ



Overview 編

Spaces 編

Projects, Blueprints 編

Source repositories 編

Dev Environments 編

Workflow 編

Issues 編

Identity, permissions, and access 編

Extensions 編

## シリーズ構成

- 全体像をお伝えする Overview 編
- 各機能の詳細についてお伝えする各機能編

## シリーズの対象読者

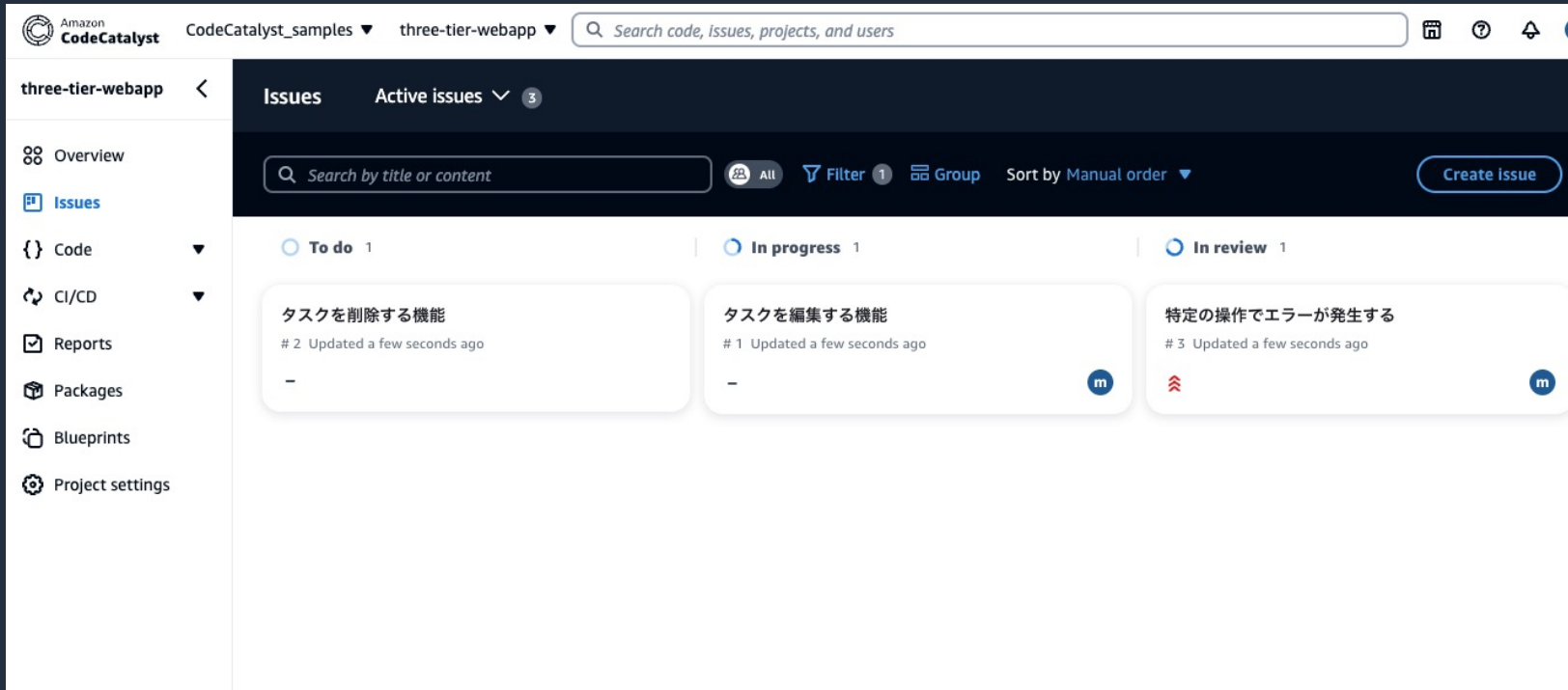
- チーム開発をするすべてのアプリケーション開発者

# アジェンダ

1. Issues の概要
2. Issue の作成
3. Issue を起点とした開発フロー
4. Issue の見積もり管理機能
5. Jira との連携
6. Amazon Q との連携 (preview)
7. まとめ

# Issue の概要

# Issues の概要



- Amazon CodeCatalyst は、Issue と呼ばれる開発者のタスクを管理する機能を備えている
- Issues の画面から、Issue の各ステータスや割り当てられた担当者を確認することが可能



# Issue の作成

# Issue の作成

**Create issue** [Close]

todo 項目の編集機能

**Description**

H# B I *↶* { } “ ≡ ≡ | 🔗 🖼️ 📄 — | 👁️ 🗑️ | ↶ ↷

\* todo 項目を編集する機能  
\* タイトルの変更  
\* description の変更

**Status** Priority

To do - No priority

**Labels**  
+ Add label

**Assignees**  
+ Add an assignee + Add me + Assign to Amazon Q Preview feature

You can assign 15 issues to Amazon Q per month. 13 issue assignments remaining this month.

**Custom Fields**  
+ Add custom field

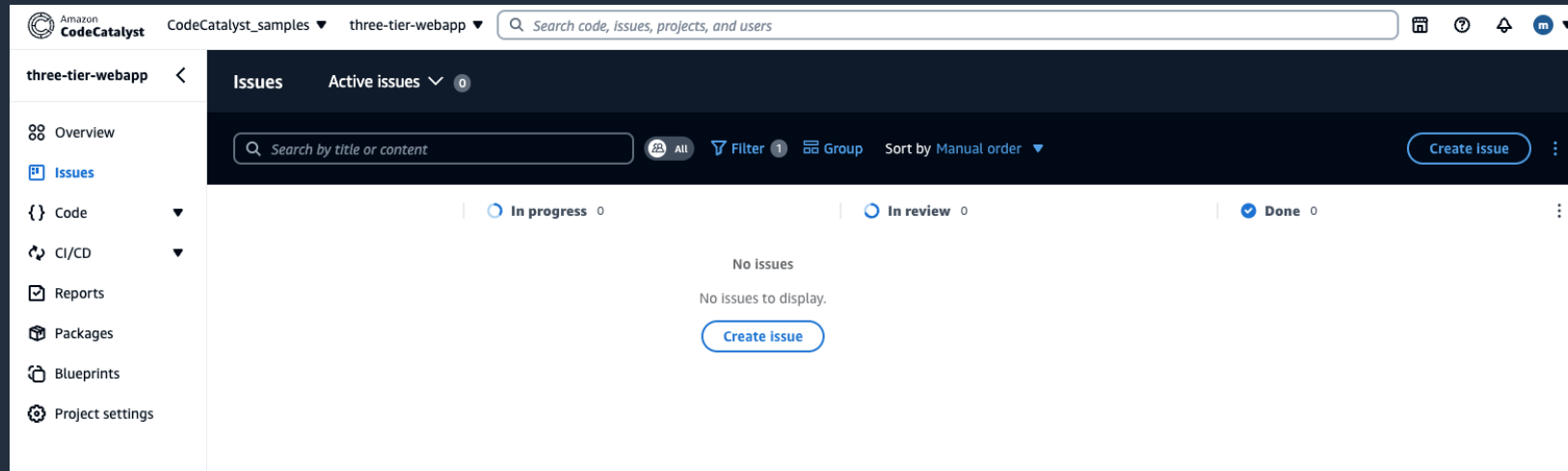
**Attachments**  
Upload file

Cancel Create issue

- Issue を作成して開発者のタスクを管理することが可能
- Status (Backlog, To do, In progress, In review, Done)
- Priority (No priority, High, Medium, Low)
- 上記を選択することで、Issues の一覧からそれぞれのステータスやプライオリティを可視化可能

# Issue を起点とした開発フロー

# 開発フロー



- カンバンボード形式での Issue を起点とした開発フローの例を説明します

# Issue 作成

### Create issue

**Description**

H# B I S {} “ ≡ ≡ | 🔗 🖼️ 📄 = | 👁️ 🗑️ | ↶ ↷

\* TODOリストを削除する機能  
\* レコードは論理削除とする  
|

**Status**  To do **Priority**

**Labels**  
[+ Add label](#)

**Assignees**  
[+ Add an assignee](#) [+ Add me](#) [+ Assign to Amazon Q](#) Preview feature

You can assign 15 issues to Amazon Q per month. 14 issue assignments remaining this month.

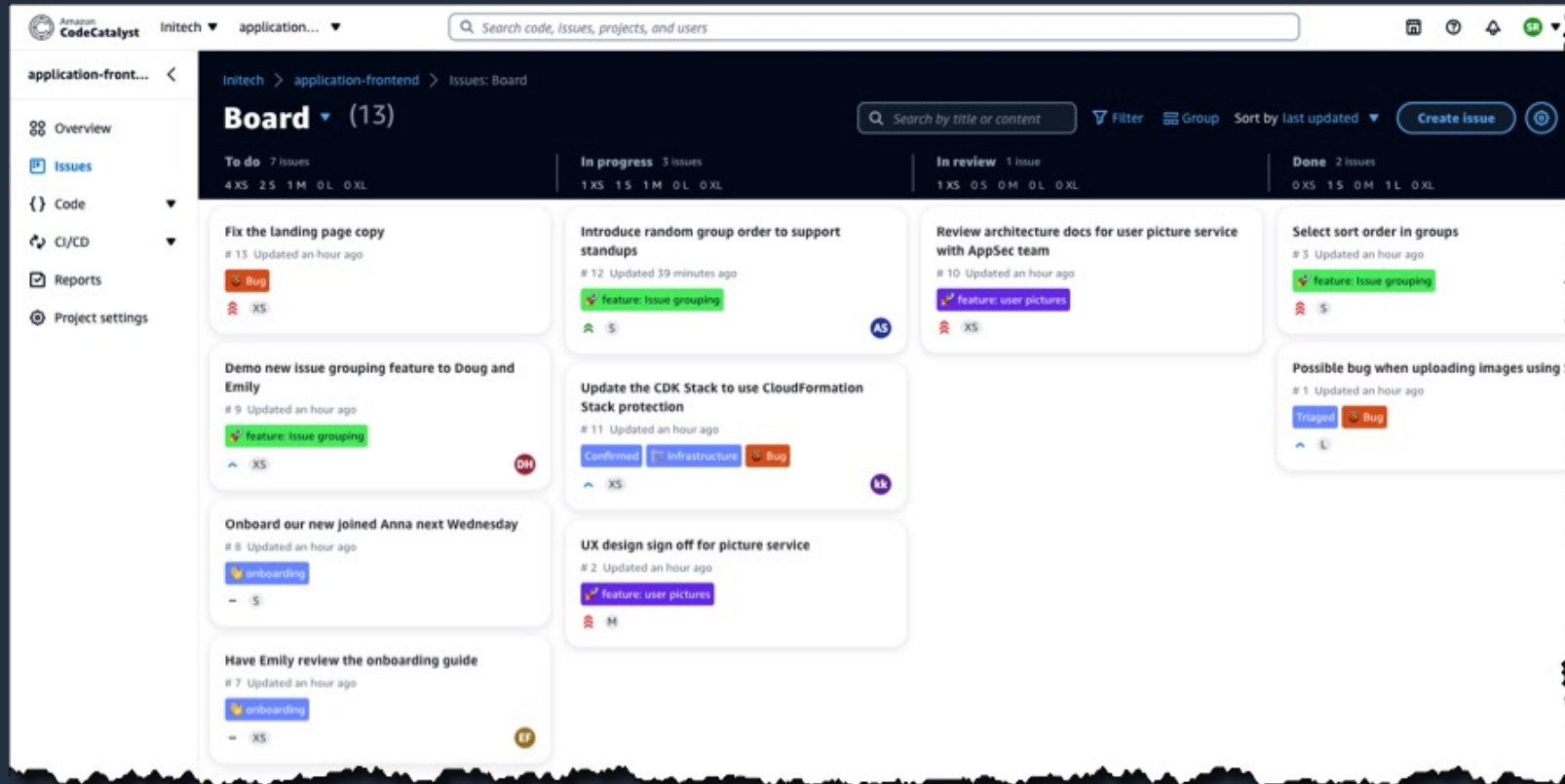
**Custom Fields**  
[+ Add custom field](#)

**Attachments**  
[📎 Upload file](#)

[Cancel](#) [Create issue](#)

- Issue を作成します

# カンバンでの Issue 管理



- Issue は、デフォルトで To do のステータスとして作成されます
- 一覧形式で、ソートやフィルタリングを行うことが可能です

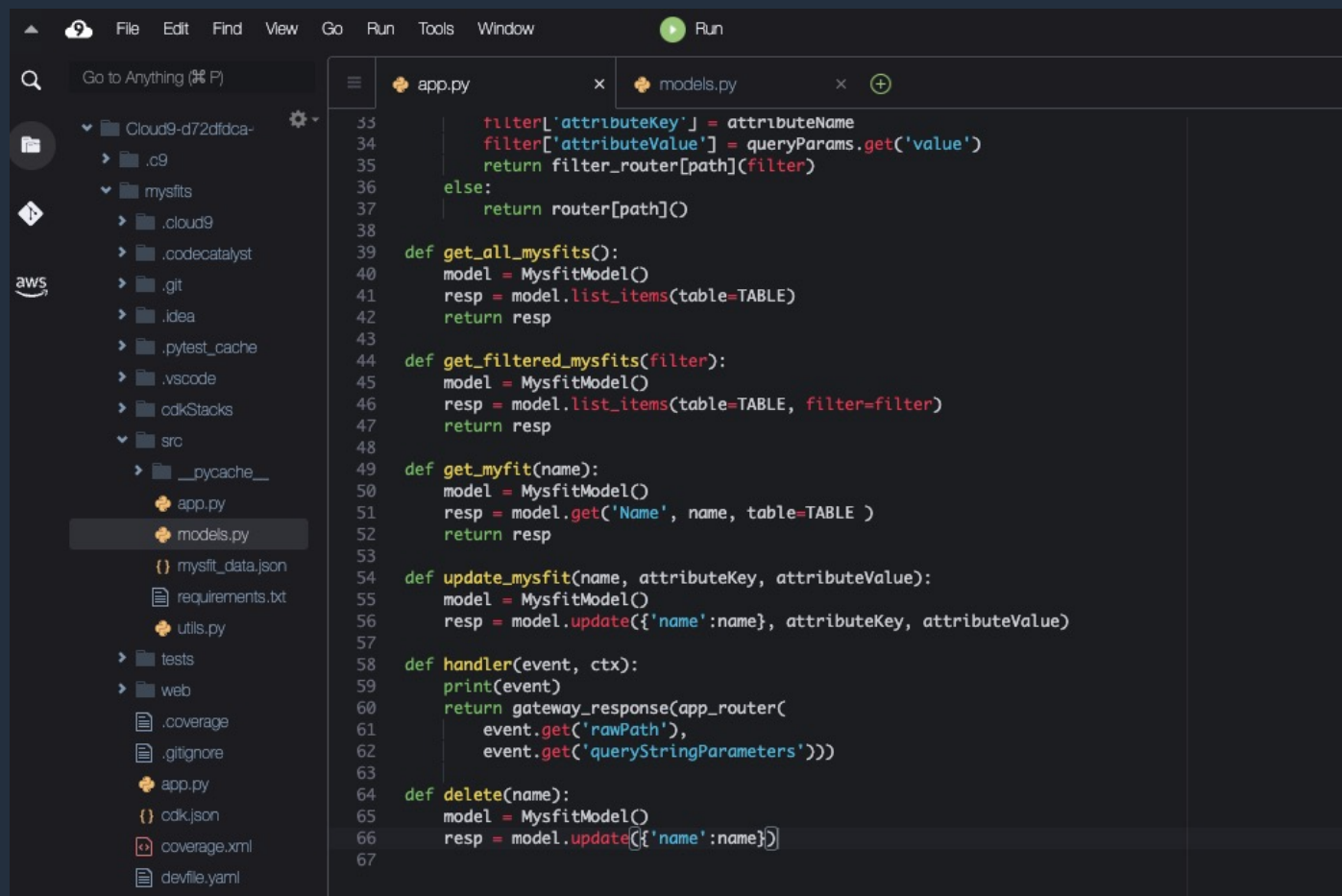
# Issue をアサイン、ステータス変更

The screenshot shows the AWS CodeCommit 'Issues' interface. The left sidebar displays a list of issues under 'Active issues' with a search bar and filters. The main area displays issue #8, 'タスクの削除機能', created 2 minutes ago by @eguchima. The issue is currently in the 'To do' status. The right sidebar shows options to change the status to 'In progress', add labels, assignees, and attachments. A comment box is visible at the bottom of the main area.

- Assignee を設定、[Move to: In Progress] のボタンでステータスを変更して開発作業に着手します



# ソースコードの修正



```
33     filter['attributeKey'] = attributeName
34     filter['attributeValue'] = queryParams.get('value')
35     return filter_router[path](filter)
36 else:
37     return router[path]()
38
39 def get_all_mysfits():
40     model = MysfitModel()
41     resp = model.list_items(table=TABLE)
42     return resp
43
44 def get_filtered_mysfits(filter):
45     model = MysfitModel()
46     resp = model.list_items(table=TABLE, filter=filter)
47     return resp
48
49 def get_myfit(name):
50     model = MysfitModel()
51     resp = model.get('Name', name, table=TABLE)
52     return resp
53
54 def update_mysfit(name, attributeKey, attributeValue):
55     model = MysfitModel()
56     resp = model.update({'name':name}, attributeKey, attributeValue)
57
58 def handler(event, ctx):
59     print(event)
60     return gateway_response(app_router(
61         event.get('rawPath'),
62         event.get('queryStringParameters')))
63
64 def delete(name):
65     model = MysfitModel()
66     resp = model.update({'name':name})
67
```

- 該当 Issue のソースコード修正、コミットを行います



# Pull Request を作成する

The screenshot shows the Amazon CodeCatalyst interface for the 'three-tier-webapp' project. The 'Pull requests' tab is selected, and a table of pull requests is displayed. The table has the following data:

Title	ID	Status	Source repository	Author	Reviewers	Created	Last modified
create API documentation in swa...	PR3	Open	mysfits			1 day ago	1 day ago
add function register()	PR2	Open	mysfits			14 days ago	14 days ago
add app.py	PR1	Open	mysfits			14 days ago	14 days ago

- Pull Request を作成します

※ Source repositories の詳細については Black Belt Amazon CodeCatalyst シリーズ Source repositories 編参照

# Pull Request の詳細、 Issue との関連付け

The screenshot displays the 'Create pull request' page in Amazon CodeCatalyst. The main content area is titled 'Create pull request' and contains the following sections:

- Pull request details:**
  - Source repository:** A dropdown menu set to 'mysfits'.
  - Destination branch:** A dropdown menu set to 'main'. Below it, a note states: 'This is the branch where you want to merge the reviewed changes. The destination branch must be different than the source branch.'
  - Source branch:** A dropdown menu set to 'develop'. Below it, a note states: 'This is the branch that contains the changes you want reviewed.'
  - Pull request title:** A text input field containing 'add delete function'. Below it, a note states: 'Maximum character length: 150'.
  - Pull request description - optional:** A text area with a placeholder 'Provide a description for this pull request so that project users know what it is for.' and a button 'Write description for me' with a 'Preview feature' tag.
- Issues - optional:** A section with a search bar 'Search by ID or title' and a message: 'We can't link any issues because you've already linked all the issues in this project.'
- Optional reviewers - optional:** A section with a button '+ Add optional reviewers'.

The left sidebar shows navigation options: Overview, Issues, Code, Source repositories, Pull requests (selected), Dev Environments, CI/CD, Reports, Packages, Blueprints, and Project settings. At the bottom left, there is a 'Tell us what you think!' section with a feedback message.

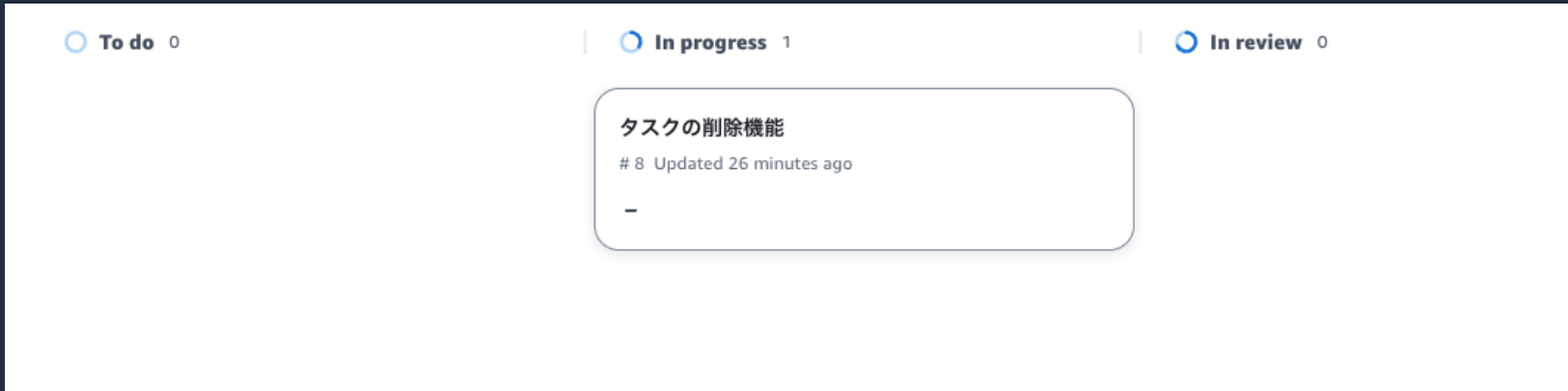
- Pull Request 作成時に、Issue との関連付けを設定可能です

# Issue から見た Pull Request

The screenshot displays the Amazon CodeCatalyst interface for an issue titled "タスクの削除機能" (Task Deletion Function). The issue is in the "In progress" state and was created 19 minutes ago by @eguchima. The main content area shows the issue title and a list of tasks: "TODOリストを削除する機能" and "レコードは論理削除とする". Below the tasks, there are "Comments (0)" and a "No comments" message. A comment box is visible, showing the user "masahiro.eguchi (you)" and a text input field with a "Send" button. On the right side, there are several action buttons: "Actions", "Move to: In review", "Status: In progress", "Priority: No priority", "Labels: + Add label", "Assignee: + Add an assignee + Add me", "+ Assign to Amazon Q (Preview feature)", "Attachments: Upload file", "Pull Requests: Open PR5 add delete function", "Link Pull Request", and "Custom Fields: + Add custom field". The left sidebar shows the navigation menu with options like Overview, Issues, Code, CI/CD, Reports, Packages, Blueprints, and Project settings. At the bottom left, there is a "Tell us what you think!" section with a "Give feedback" button.

- Issue 側からも、Pull Request が設定されていることがわかります。

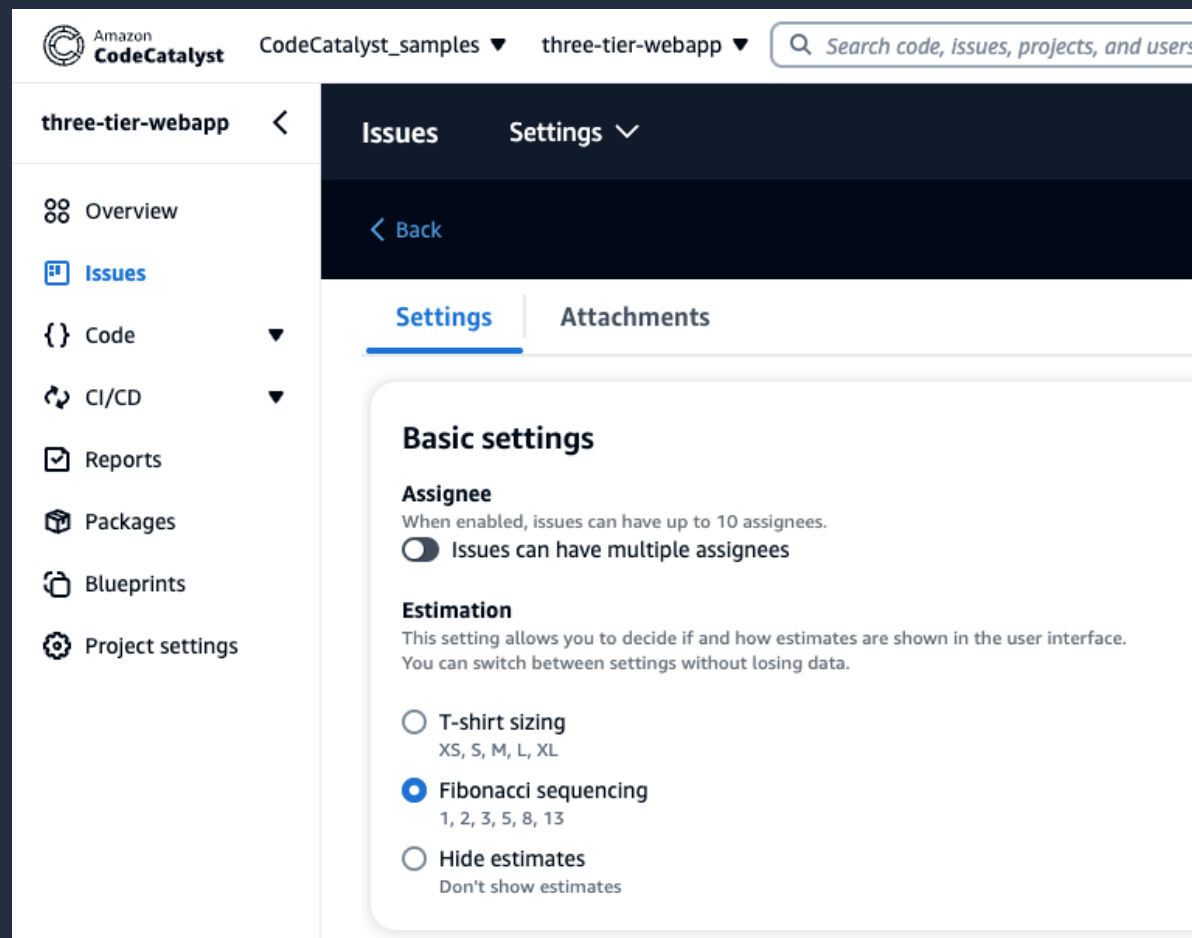
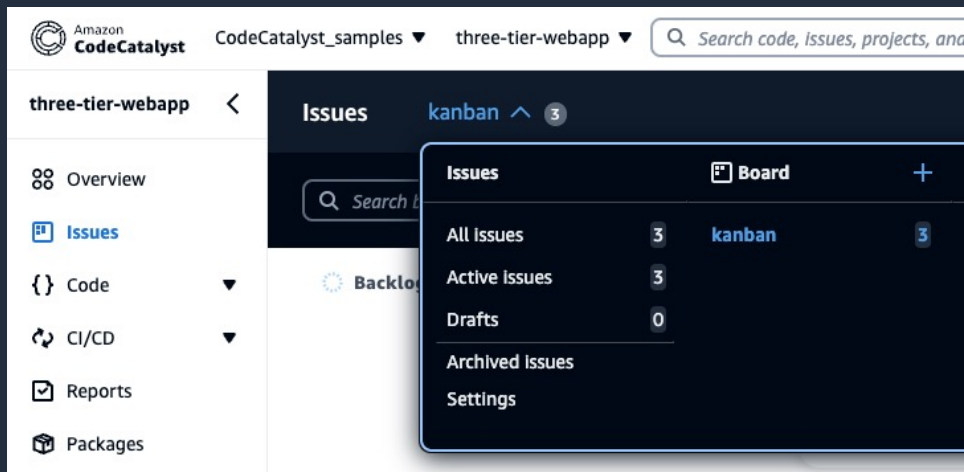
# Issue のステータスを変更する



- Issue を開き、[Move to : In review] を押すか、ドラッグ & ドロップでステータス変更することが可能です

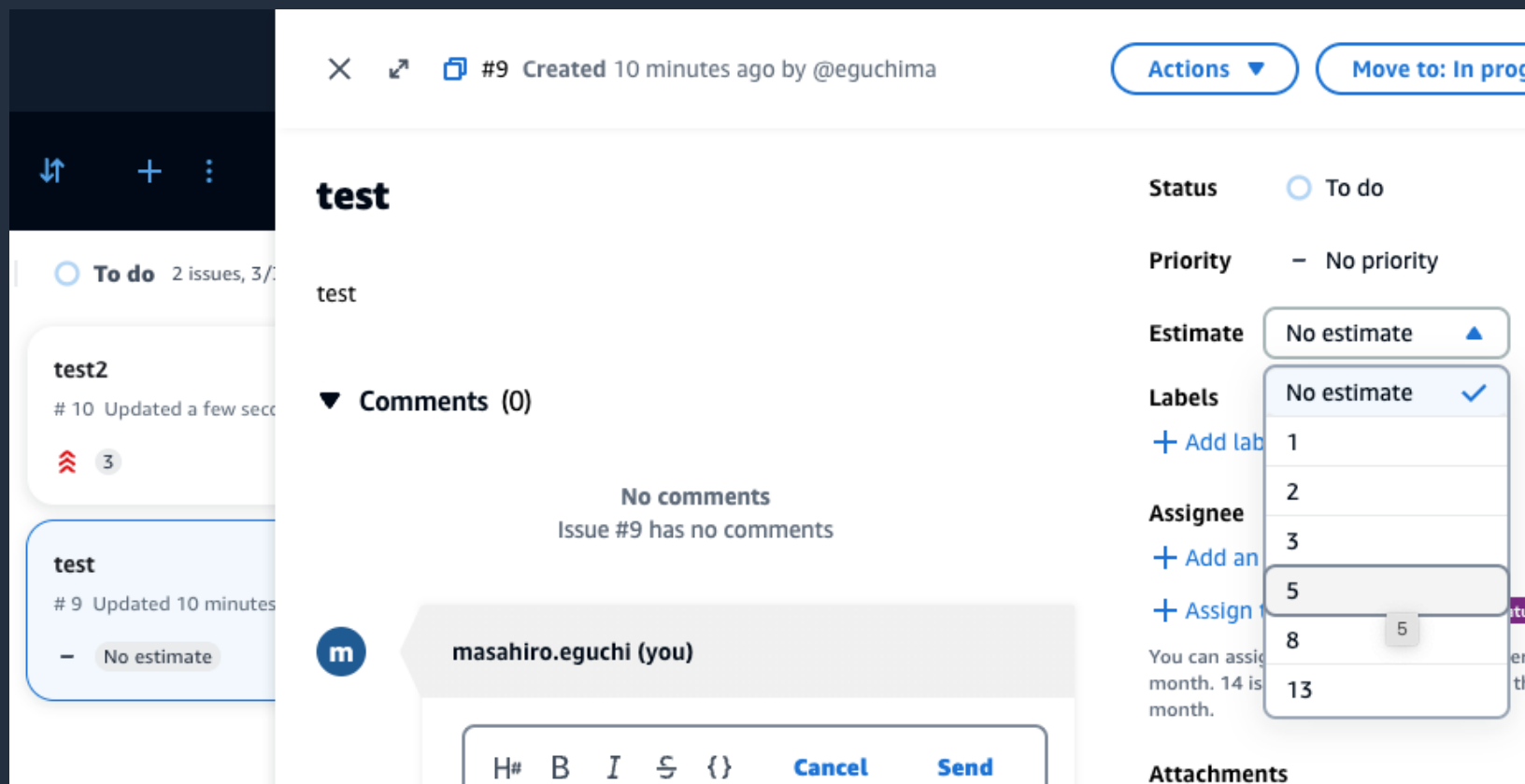
# Issue の見積もり管理機能

# Issue の見積もり管理機能



- Issues の Settings 画面から Estimation の機能をオンにすることで 見積もり値の表示が可能となる
- T-shirt sizing や Fibonacci が選択可能

# Issue の見積もり管理機能

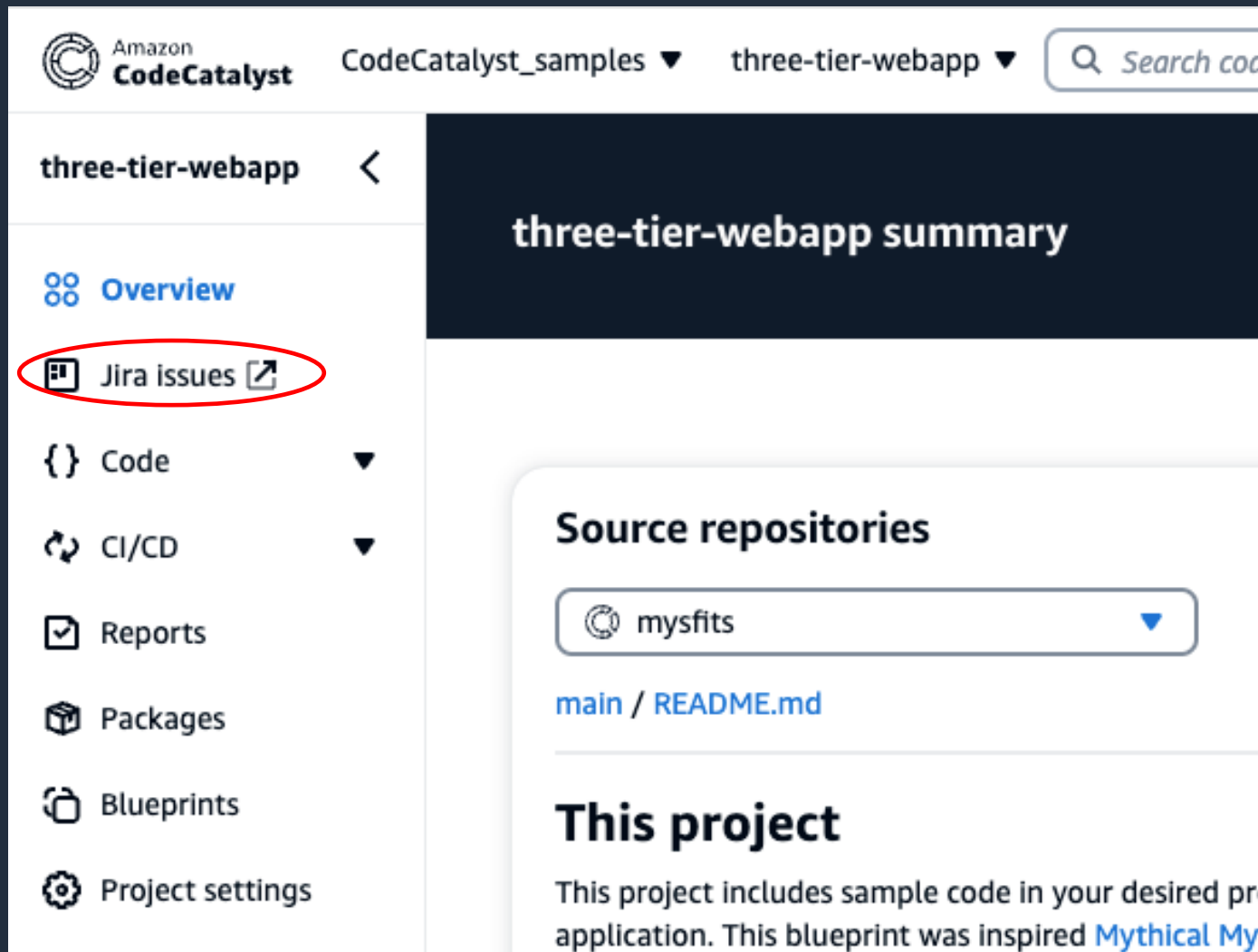


- Estimate の機能をオンにすることで、Issues の画面に見積もり値の表示が可能となる

# Jira との連携



# Jira との連携



- Extensions という拡張機能を用いることで Jira での issue 管理が可能となります。
- 詳細は Extentions 編をご確認ください

# Amazon Q との連携 (preview)

# Generative AI 機能の有効化

The screenshot shows the Amazon CodeCatalyst interface. At the top, there's a search bar and navigation tabs for 'CodeCatalyst\_samples', 'Projects', 'Activity', and 'Settings'. The 'Settings' page has a sidebar with various categories like 'General', 'AWS accounts', 'VPC connections', 'Billing', 'Generative AI', 'Machine resources', 'Dev Environments', 'Installed extensions', 'Members', 'Blueprints', 'Teams', and 'SSO'. The 'Generative AI' section is active, showing a 'Preview feature' badge and a toggle switch that is turned on. Below this, the 'Generative AI feature usage' section provides a summary of usage, including two progress bars: 'Issues assigned to Amazon Q' (1/15 issues, 7% consumed) and 'Written summaries for pull requests' (0/15 summaries, 0% consumed).

CodeCatalyst\_samples Projects Activity **Settings**

General  
AWS accounts  
VPC connections  
Billing  
**Generative AI**  
Machine resources  
Dev Environments  
Installed extensions  
Members  
Blueprints  
Teams  
SSO

**Generative AI** Preview feature

You can grant access to the generative AI features in Amazon CodeCatalyst for this space. These features have quotas that will be applied across all projects in the space.

Projects in this space can access generative AI features

**Generative AI feature usage**

You can review the usage of generative AI features in your space to help you understand how much they are being used and whether you are approaching any quota.

Issues assigned to Amazon Q	Written summaries for pull requests
15 a month	15 a month
<div style="width: 7%;"><div style="width: 7%;"></div></div> 1/15 issues 7% consumed	<div style="width: 0%;"><div style="width: 0%;"></div></div> 0/15 summaries 0% consumed

- Amazon CodeCatalyst は、Amazon Q (生成 AI) の機能が利用可能となった
- Generative AI 機能を有効化することで利用開始できる

# Issue を Amazon Q にアサインする (preview)

### Create issue

Description

H# B I S {} “ ≡ ≡ | 🔗 🖼️ 📄 = | 🔍 🗑️ ↶ ↷

---

Status To do Priority - No priority

Labels  
[+ Add label](#)

Assignees  
[+ Add an assignee](#) [+ Add me](#) [+ Assign to Amazon Q](#) Preview feature

You can assign 15 issues to Amazon Q per month. 14 issue assignments remaining this month.

▼ ⌵ Evaluate prerequisites

Waiting for user response. See comments below to respond.

🕒 Read repository

🕒 Generate background

🕒 Generate approach

🕒 Generate code

🕒 Create pull request

- Issue の担当者を Amazon Q にアサインすることで Issue から読み取った PullRequest の作成までを省力化できる

# 作成された Pull Request

The screenshot displays the Amazon CodeCatalyst interface for a Pull Request (PR2) titled "add function register()". The PR is currently "Open" and "Not mergeable". The interface shows a comparison between the "main" branch and "Revision 1 (53beb26c)".

**Files changed (3):**

- src
- app.py
- models.py
- tests

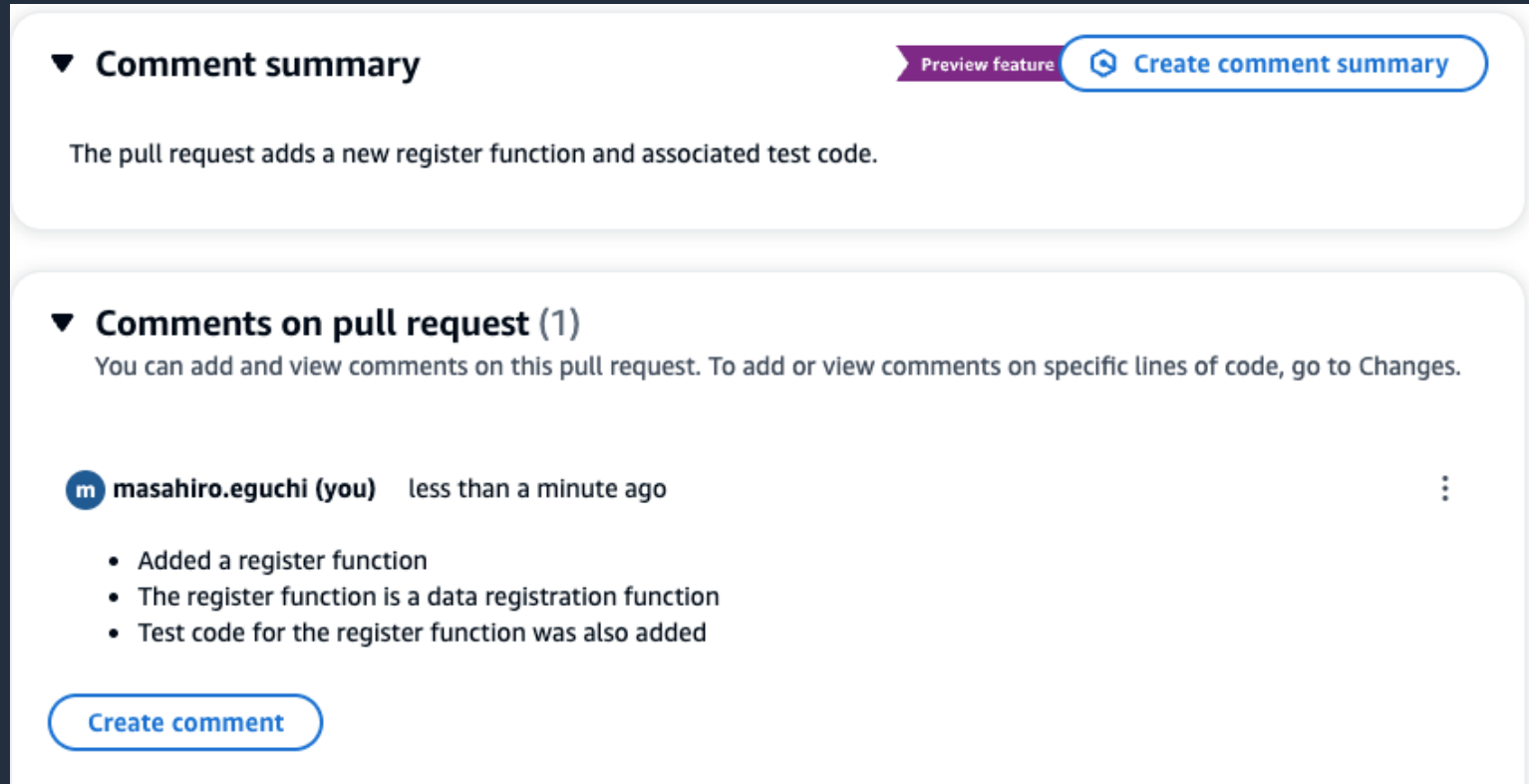
**Code Diff:**

```
14 - def app_router(path, queryParams=None):
15 -
16 -     router = {
17 -         '/mysfits': get_all_mysfits
18 -     }
19 -
20 -     filter_router = {
21 -         '/mysfits': get_filtered_mysfits
22 -     }
23 -
24 -     if queryParams:
25 -         filter = {}
26 -         attributeName=queryParams.get('filter')
27 -         if attributeName == "GoodEvil":
28 -             filter['indexName'] =
MysfitModel.GOODDEVILINDEX
29 -         elif attributeName == "LawChaos":
30 -             filter['indexName'] =
MysfitModel.LAWCHAOSINDEX
31 -         else:
32 -             raise Exception("Unknown
FilterExpression Key")
33 -         filter[attributeName] = attributeName

14 + def handler(event, ctx):
15 +     def register(username, password):
16 +         model = MysfitModel()
17 +         model.create({'username': username, 'password':
password})
18 +
19 +         return {'status': 'success'}
20 +
21 +     def app_router(path, queryParams=None):
22 +         router = {
23 +             '/mysfits': get_all_mysfits
24 +         }
25 +
26 +         filter_router = {
27 +             '/mysfits': get_filtered_mysfits
28 +         }
29 +
30 +         if queryParams:
31 +             filter = {}
32 +             attributeName=queryParams.get('filter')
33 +             if attributeName == "GoodEvil":
```

- Issue の内容から、ソースコードを生成、コミットし Pull Request が作成される

# Pull Request のサマリコメントを生成する (preview)



The screenshot shows a GitHub Pull Request interface. At the top, there is a section titled "Comment summary" with a "Preview feature" badge and a "Create comment summary" button. Below this, a text box states: "The pull request adds a new register function and associated test code." The next section is "Comments on pull request (1)", which includes a sub-header and a descriptive sentence: "You can add and view comments on this pull request. To add or view comments on specific lines of code, go to Changes." Below this, a comment by "masahiro.eguchi (you)" is shown, dated "less than a minute ago". The comment contains a bulleted list: "Added a register function", "The register function is a data registration function", and "Test code for the register function was also added". At the bottom of the comment section, there is a "Create comment" button.

- Pull Request の入力したコメントから、サマリコメントを生成することが可能

# Amazon Q Tier ごとの利用可能回数 (preview)

## (Preview) Amazon Q in CodeCatalyst

	Free	Standard	Enterprise
(Preview) Pull requests generated by Amazon Q feature development capability	-	15 PRs per space per month*	20 PRs per user per month (up to 300 per space per month)*
(Preview) Comment summaries or pull request descriptions generated by Amazon Q	5 per space per month*	15 per space per month*	20 per user per month (up to 300 per space per month)*

\*These limits may be subject to change. Using these features will require an Amazon Q Builder subscription after the preview.

- PullRequest を生成する機能については Standard Tier 以上の利用が必要
- 上記は 2023 年 12 月現在の料金となります  
<https://codecatalyst.aws/explore/pricing> をご覧ください

# まとめ



# まとめ

- Amazon CodeCatalyst は、統合開発環境であり Issue の管理機能を備えている
- CodeCatalyst で作成した Issue と PullRequest の紐づけを行って管理することが可能
- Amazon Q との連携により、PullRequest 作成までの省力化が可能



Thank you!