



Amazon CodeCatalyst

Source repositories 編

周平 兼 国

Solutions Architect
2023/12

自己紹介

国兼 周平 / Kunikane Shuhei

Amazon Web Services Japan ソリューションアーキテクト

経歴:

1. 国内 Sier にて、アプリケーションエンジニアとして金融/公共の分野で開発
2. インターネットサービス企業にて、バックエンド開発者としてネット証券取引サービスやネットバンキングサービスの立ち上げ
3. AWS にて、Professional Services (*) のコンサルタントとしてモビリティのお客様を支援
4. ソリューションアーキテクトに転向

得意分野:

- 高負荷 WEB サービスの設計/開発
- CI/CD, IaC, Serverless, Container

* Professional Services: AWS の提供する有償のコンサルティングサービス



My favorite AWS Services



Amazon ECS



AWS Lambda



Amazon CodeCatalyst

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=PLzWGOASvSx6FlwIC2X1nObr1KcMCBBlqY>



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

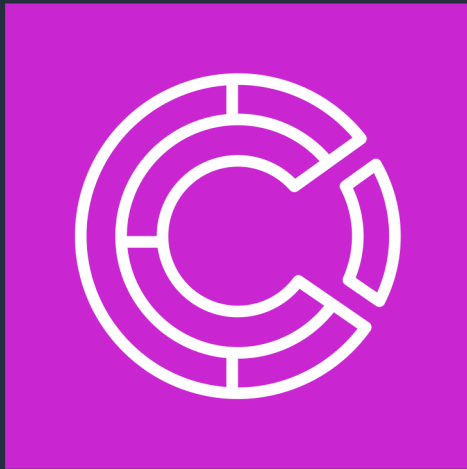
- チーム開発をするすべてのアプリケーション開発者

アジェンダ

1. Source Repositories 概要
2. リポジトリへの接続方法
3. コンソールでのリポジトリ操作
4. Pull Request
5. Branch ルール
6. 通知
7. クォータ

Source Repositories 概要

Source Repositories 概要



- ✓ Amazon CodeCatalyst 内で提供されるマネージドな Git 互換ソース管理
- ✓ 標準的な Git ツールや Git 対応 IDE が利用可能
- ✓ Space 固有のキーを使用した暗号化
- ✓ コンテンツの直接編集をサポート
- ✓ Pull Request 機能をサポート
- ✓ 1つの Project 内で複数のリポジトリを管理
- ✓ ワークフローの定義もソースコードの一部として保存

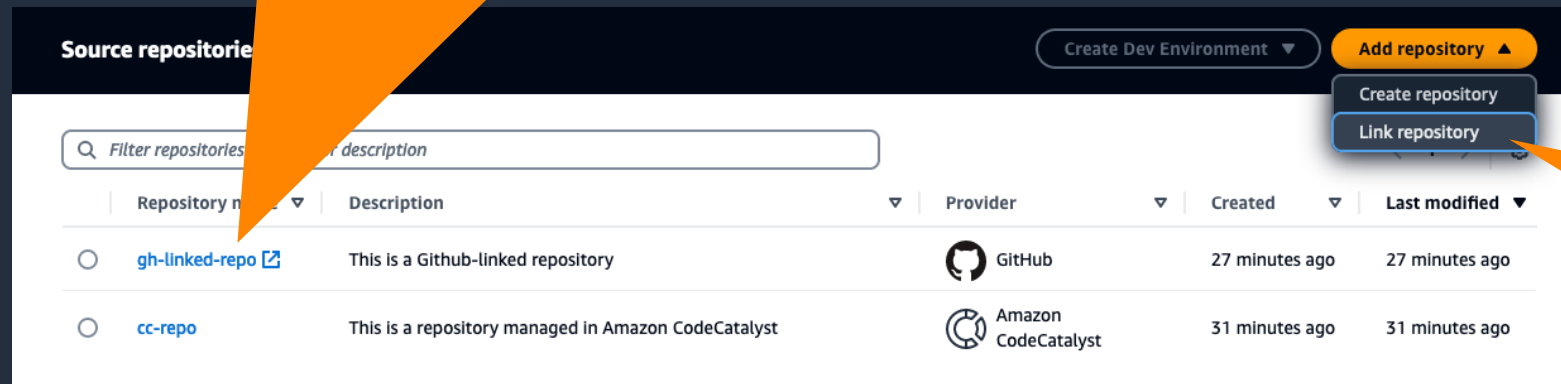
参考: git を学ぶ

- Pro Git
<https://git-scm.com/book/ja/v2> (日本語)
- Git Cheat sheet
<https://training.github.com/downloads/ja/github-git-cheat-sheet.pdf> (日本語)
- Git Immersion
<http://gitimmersion.com/>
- Git Reference
<https://git-scm.com/docs>
- git は HTTP プロトコルを利用する場合、デフォルトでは毎回ユーザー名、パスワードの入力を求めるが、`git config --global credential.helper` を利用することでクレデンシャル情報をキャッシュまたは保存することが可能
<https://git-scm.com/book/en/v2/Git-Tools-Credential-Storage>

GitHub repositories extension

- GitHub のリポジトリを CodeCatalyst にリンク
- GitHub でソースコードを管理しつつ CodeCatalyst の Workflow を利用

クリックすると GitHub の画面が開く
ソースコードの編集は GitHub で実施



Repository name	Description	Provider	Created	Last modified
gh-linked-repo	This is a Github-linked repository	GitHub	27 minutes ago	27 minutes ago
cc-repo	This is a repository managed in Amazon CodeCatalyst	Amazon CodeCatalyst	31 minutes ago	31 minutes ago

Add repositoryから
新しいリンクをセットアップ

※ GitHub repositories extension については、Black Belt Amazon CodeCatalyst シリーズ Extensions 編および Projects, Blueprints 編も参照

**以降は
Amazon CodeCatalyst 内で提供されるリポジトリ
についての説明です**

**GitHub repositories extension を利用して
リンクをセットアップしている場合は
GitHub でリポジトリを管理します**

リポジトリの作成

Source repositories (2)

Create Dev Environment Add repository

Filter repositories by name or description

Repository name	Description	Provider	Created	Last modified
gh-linked-repo	This is a Github-linked repository	GitHub	about 1 hour ago	about 1 hour ago
cc-repo	This is a repository managed in Amazon CodeCatalyst	Amazon CodeCatalyst	about 1 hour ago	about 1 hour ago

Create source repository

Repository details

Repository name

Repository names must be between 1-100 valid characters. Valid characters include: A-Z, a-z, 0-9, periods, underscores, and hyphens.

Description - optional

Provide a description for this repository so that project users know what it is for. This is particularly important if you plan to add this repository to multiple projects.

Maximum character length: 255

.gitignore file

Add a .gitignore file to avoid adding certain types of files to your repository.

Cancel Create

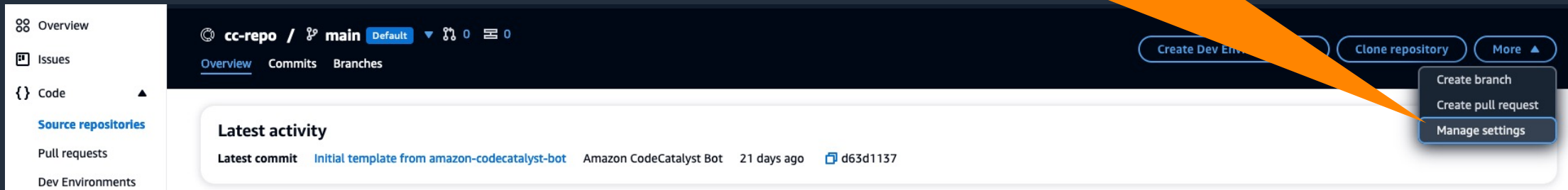
```
1 # Compiled class file
2 *.class
3
4 # Log file
5 *.log
6
7 # BlueJ files
8 *.ctxt
9
10 # Mobile Tools for Java (J2ME)
11 .mtj.tmp/
12
13 # Package Files #
14 *.jar
15 *.war
16 *.nar
17 *.ear
18 *.zip
19 *.tar.gz
20 *.rar
21
22 # virtual machine crash logs, see http://www.java.com/en/download/help/error_hotspot.xml
23 hs_err_pid*
```

gitignore を自動生成

Default Branch

- リポジトリを開いたときに最初に表示される Branch
- リポジトリ作成と同時に「main」という Branch が自動的に生成され、Default Branch となる

リポジトリの設定画面から変更可能



The screenshot shows the AWS CodeCommit interface for a repository named 'cc-repo'. The current branch is 'main', which is marked as the 'Default' branch. The 'More' dropdown menu is open, showing options: 'Create branch', 'Create pull request', and 'Manage settings'. The 'Manage settings' option is highlighted, indicating that the default branch can be changed from this menu.

リポジトリへの接続方法

リポジトリへの接続方法

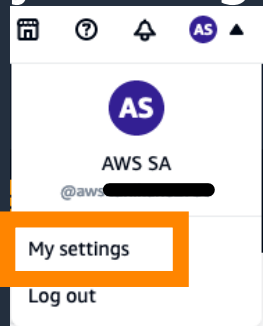
- Dev Environment からの接続
 - リポジトリと紐づけた Dev Environment を作成すると、自動的に Clone された状態で作業を開始できる
- ローカル Git ツールからの接続
 - PAT (Personal Access Token) を利用して Clone し、作業を開始する

※ Dev Environment については、[Black Belt Amazon CodeCatalyst シリーズ Dev Environments 編参照](#)

PAT (Personal Access Token)

- ローカル Git ツールから CodeCatalyst のリポジトリへのアクセスには PAT (Personal Access Token) を利用する
- PAT は User に関連付けられる
 - User が参加しているすべての Space/Project で共通して利用できる
- PAT 発行時は画面上に一度だけ表示される
 - ローカルコンピューターに安全に保存して利用する
 - 紛失または流出した場合は新しいものを作成し、既存のものを削除する

MySetting から発行



Personal access tokens (4)

The personal access tokens (PATs) that you create are associated with your user identity in CodeCatalyst. You can manage these PATs in your user settings across all of your CodeCatalyst spaces. Your PATs are automatically deleted when they expire, and they are no longer displayed.

[Delete](#) [Create](#)

ローカル Git ツールからリポジトリへの接続

The screenshot shows the AWS CodeCatalyst interface for a repository named 'cc-repo'. The 'Clone repository' button is highlighted with an orange box. Below it, the 'Latest activity' section shows a commit by 'Amazon CodeCatalyst Bot'. A modal dialog titled 'Clone using HTTPS' is open, displaying the 'HTTPS clone URL' and a 'Personal access token (password)' section. An orange arrow points from the 'Clone repository' button to the dialog, and another orange arrow points from the 'HTTPS clone URL' field to a terminal command box.

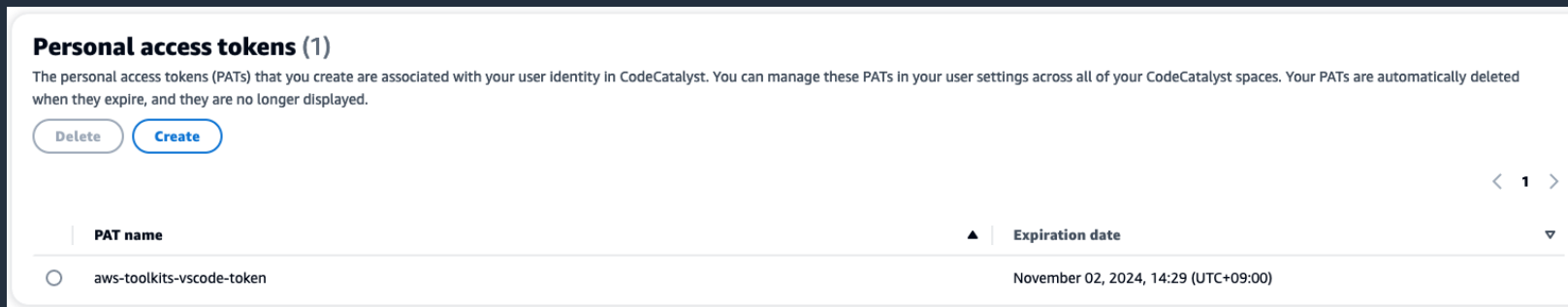
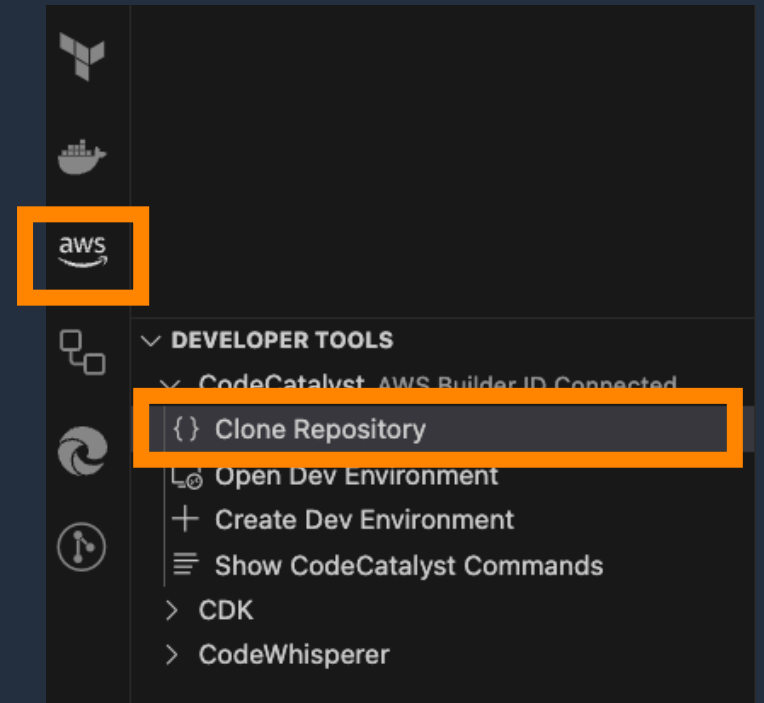
```
$ git clone <URL>
```

- IDE を利用する場合の手順は IDE のドキュメントに従う
- パスワードとして PAT を使う
- PAT は OS のクレデンシャル管理システムに保存するのがベストプラクティス

ローカル Git ツールからリポジトリへの接続 - with AWS Toolkit for VS Code

- AWS Toolkit for VS Code を利用してリポジトリを clone することができる

- この場合は PAT の生成は自動的に行われる



コンソールでのリポジトリ基本操作

コンソールからのコンテンツ参照

- ツリーを展開して内容を表示するファイルを指定
- Branch を切り替えての表示も可能
- タブで分けて複数のファイルを表示可能

The screenshot displays the AWS Code console interface for a repository named 'spa-app' on the 'main' branch. The left sidebar shows a file tree with 'router/index.ts' selected. The main area shows the code for 'index.ts' with two tabs: 'index.ts' and 'main.ts'. An orange callout box labeled 'Branch の切り替え' points to the branch dropdown menu in the top right, which lists 'main', 'hotfix01', 'hotfix02', 'hotfix03', 'feature02', and 'feature01'. Another orange callout box labeled 'タブで複数ファイルを表示' points to the file tabs. A third orange callout box labeled '編集画面へ' points to the 'Edit' button in the top right corner of the code editor area.

コンソールからのコンテンツ編集

- コンテンツ参照画面から遷移した編集画面で編集、Commit を作成

```
vue-spa-example < Code Source repositories Pull requests Dev Environments
spa-app > Edit file
spa-app: main / src/router/index.ts
1 import { createRouter, createWebHistory } from "vue-router";
2 import HomeView from "../views/HomeView.vue";
3
4 // This is example comment
5 const router = createRouter({
6   history: createWebHistory(import.meta.env.BASE_URL),
7   routes: [
8     {
9       path: "/",
10      name: "home",
11      component: HomeView,
12    },
13    {
14      path: "/about",
15      name: "about",
16      // route level code-splitting
17      // this generates a separate chunk (About.[hash].js) for this route
18      // which is lazy-loaded when the route is visited.
```

表示中の Branch に Commit を作成し
変更を反映

Commit changes

You're committing this file directly to the branch. If you would like a review of this file, you can create a pull request after committing the file to the branch.

Commit message
Provide a description for this pull request so that project users know what it is for.

feature: Add task view

Character length: 1,000

Cancel Commit

Commit 履歴の表示

The screenshot shows the GitHub interface for a repository named 'spa-app'. The 'Commits' tab is active, displaying a list of recent commits. An orange callout box labeled 'Branch の切り替え' (Branch switching) points to the branch dropdown menu, which is currently set to 'main'. A second orange callout box labeled '変更内容確認画面へ' (To change content confirmation screen) points to the commit titles. To the right, a separate window shows a list of branches: 'main' (Default), 'hotfix01', 'hotfix02', 'hotfix03', 'feature02', and 'feature01', with a 'View all' button at the bottom.

Commit title	Commit author	Commit date	Commit ID
feature: Enhance UX	aws-xxxxxx	2 minutes ago	5d34c2f5
feature: Add search feature	aws-xxxxxx	8 minutes ago	07729ac1

Commit 履歴の表示

Changes made in commit 69ed66e7

Parent commit
2e8c9f5bc5859145a6b202bd4d510d3e76315f7c

Parent commit date
17 minutes ago

Parent commit author
awskunikane2738

Parent commit message
feature: Enhance UX

Commit message
feature: Add thread view

Changes
Comparing the specified parent commit to this commit and displaying the changes made by this commit.

The differences are ready for review.

src/router/index.ts +3 -6

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 -     path: "/about",
14 -     name: "about",
15 -     // route level code-splitting
16 -     // this generates a separate chunk (About.[hash].js) for this
17 -     // route
18 -     // which is lazy-loaded when the route is visited.
19 -     component: () => import("../views/AboutView.vue"),
20   },
21 ];
```

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 +     path: "/thread",
14 +     name: "thread",
15 +     component: () => import("../views/ThreadShow.vue"),
16   },
17 ],
18 };
```

親 Commit が複数ある場合は
展開して比較対象を選択

Changes made in commit bd2d0b51

Parent commits

- 2e8c9f5bc5859145a6b202bd4d510d3e76315f7c
- 81de6b6c7a0b691e63210961ca8334d1cb47d3ab

Unified

Changes

Comparing the specified parent commit to this commit and displaying the changes made by this commit.

The differences are ready for review.

src/router/index.ts +3 -6

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 -     path: "/about",
14 -     name: "about",
15 -     // route level code-splitting
16 -     // this generates a separate chunk (About.[hash].js) for this
17 -     // route
18 -     // which is lazy-loaded when the route is visited.
19 -     component: () => import("../views/AboutView.vue"),
20   },
21 ];
```

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 +     path: "/thread",
14 +     name: "thread",
15 +     component: () => import("../views/ThreadShow.vue"),
16   },
17 ],
18 };
```

Split

Changes

Comparing the specified parent commit to this commit and displaying the changes made by this commit.

The differences are ready for review.

src/router/index.ts +3 -6

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 -     path: "/about",
14 -     name: "about",
15 -     // route level code-splitting
16 -     // this generates a separate chunk (About.[hash].js) for this
17 -     // route
18 -     // which is lazy-loaded when the route is visited.
19 -     component: () => import("../views/AboutView.vue"),
20   },
21 ];
```

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 +     path: "/thread",
14 +     name: "thread",
15 +     component: () => import("../views/ThreadShow.vue"),
16   },
17 ],
18 };
```


Pull Request

Pull Request

- ある Branch から別の Branch へのコード変更の内容についてメンバー間でレビューしたり、議論したりするための方法
- 問題なければ Merge することで宛先 Branch に変更を反映
- 変更の発生している Merge 元 Branch を「**Source Branch**」、Merge しようとしている宛先 Branch を「**Destination Branch**」と呼ぶ
- Issue とリンクさせることで、どんな課題が解決されるかを追跡しやすくすることができる

Pull Request 一覧から作成開始
(ほかにも作成できる動線はいくつかある)

The screenshot shows the AWS CodeCommit console interface. The left sidebar contains navigation options: Overview, Issues, Code, Source repositories, Pull requests (selected), and Dev Environments. The main content area displays the 'Pull requests' section for the 'vue-spa-example' repository. A search bar and filters (Open, Any author) are visible. Below the filters is a table with the following data:

Title	ID	Status	Source repository	Author	Reviewers	Created	Last modified
feature: Enhance UX	PR4	Open	spa-app	AS	-	less than a m...	less than a minute ago

Pull Request の作成

どの Branch からどの Branch に Merge しようとしているかを指定

Pull request details
spa-app

Destination branch
This is the branch where you want to merge the reviewed changes.
main

Source branch
This is the branch that contains the changes you want reviewed.
feature05

The destination branch must be different than the source branch.

Pull request title
feature: Enhance UX
Maximum character length: 150

Pull request description - optional
Provide a description for this pull request so that project users know what it is for.

H# B I ↶ {} “ ☰ ☷ | 🔗 📎 🗑️ 🗑️ | 🔍 🗑️ ↶ ↷

Change Summary
* Add thread view
Maximum character length: 10,000

Issues - optional
[Link Issues](#)

Required reviewers - optional
[+ Add required reviewers](#)

Optional reviewers - optional
[+ Add optional reviewers](#)

Issue とのリンクを設定

必須レビューを指定
(Merge のために Approve 必須)

Optional レビューを指定
(Approve がなくても Merge 可能)

Pull Request のレビュー

feature: Enhance UX PR5 Open Mergeable

spa-app main ← feature05 Create Dev Environment More Merge

Description [Changes](#) [Reports](#)

▼ **Comments on pull request (0)**

No comments

You don't have any general comments about the pull request, but you might have comments on specific lines and changes.

[Create comment](#)

Comparing main Generating code coverage report Unified Split

To Revision 1

✓ The differences are ready for review.

▼ **src/router/index.ts +3 -6**

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 -   path: "/about",
14 -   name: "about",
15 -   // route level code-splitting
16 -   // this generates a separate chunk (About.[hash].js) for this route
17 -   // which is lazy-loaded when the route is visited.
18 -   component: () => import("../views/AboutView.vue"),
19   },
20 }
```

```
@ -10,12 +10,9 @@
10     component: HomeView,
11   },
12   {
13 +   path: "/thread",
14 +   name: "thread",
15 +   component: () => import("../views/ThreadShow.vue"),
16   },
17 }
```

Additional info

Workflow runs (1)

Revision 1 (latest)

In progress onPullRequestBuildAndTest (Run-5d21b)

Issues (0)

[Link issues](#)

Feasible merge strategies

- Fast forward
- Squash

Workflow の状態
リンクから Workflow の設定や進捗、
テストレポートなど参照できる

変更内容を確認

Pull Request のレビュー

Pull Request 全体へのコメント

▼ Comments on pull request (0)

No comments

You don't have any general comments about the pull request, but you might have comments on specific lines and changes.

Create comment

▼ Comments on pull request (0)

No comments

You don't have any general comments about the pull request, but you might have comments on specific lines and changes.

Comment

H# B I S {} “ ≡ ≡ | 🔗 🖼️ 📄 - 🔍 📄 ↶ ↷ Cancel Save

This is a comment for entire this pull request.

特定の行へのコメント

```
12 {
13 - path: "/about",
14 - name: "about",
15 - // route level code-splitting
16 - // this generates a separate chunk (About.
17 [hash].js) for this route
18 - // which is then loaded when the route is
19 - // visited. This can be used to
20 - // delay loading certain assets
21 - // until they are needed, which helps
22 - // improve perceived load times.
23 - // Note: this feature is experimental
24 - // and may be removed in the future.
25 - // Learn more about code splitting:
26 - // https://reactjs.org/docs/code-splitting.html
27 - // You can also use the `react.lazy`
28 - // method.
```

```
12 {
13 - path: "/about",
14 - name: "about",
15 - // route level code-splitting
16 - // this generates a separate chunk (About.
17 [hash].js) for this route
18 - // which is then loaded when the route is
19 - // visited. This can be used to
20 - // delay loading certain assets
21 - // until they are needed, which helps
22 - // improve perceived load times.
23 - // Note: this feature is experimental
24 - // and may be removed in the future.
25 - // Learn more about code splitting:
26 - // https://reactjs.org/docs/code-splitting.html
27 - // You can also use the `react.lazy`
28 - // method.
```

Comment

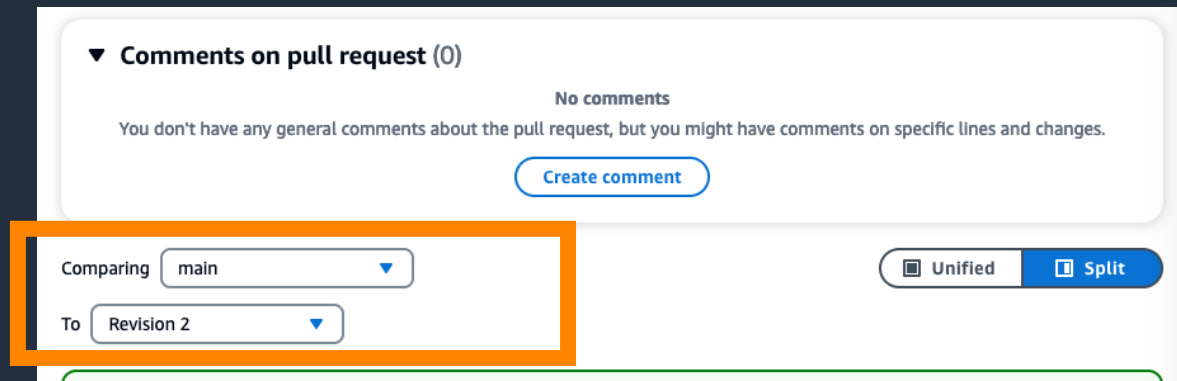
H# B I S {} “ ≡ ≡ | Cancel Save

🔗 🖼️ 📄 - 🔍 📄 ↶ ↷

Is this path correct?

Pull Request のレビュー

- Pull Request 作成後に Source Branch に発生した変更には Revision が採番される
- デフォルトでは Destination Branch の先頭 Commit と Pull Request の最新 Revision の間の差分が表示される



- 任意の Revision を比較対象とした差分表示をすることもできる
- Revision 同士の比較もできる

Pull Request のレビュー

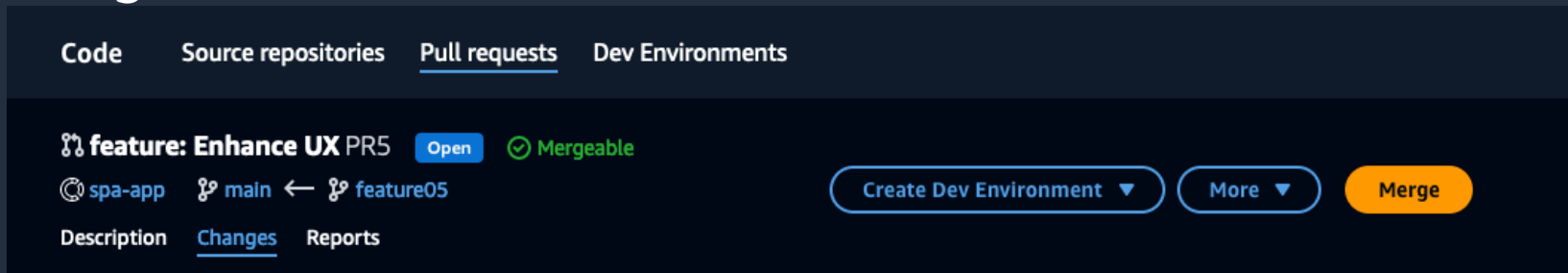
- レビューに指定されているメンバーは、変更内容を確認し、問題なければ Approve する
- Required レビューに指定されたメンバーが Approve するまで Merge できない

The screenshot shows a GitHub Pull Request (PR) interface. At the top, there are tabs for 'Code', 'Source repositories', 'Pull requests', and 'Dev Environments'. Below this, the PR title is 'feature: Enhance UX PR5' with 'Open' and 'Not mergeable' status. There are buttons for 'Create Dev Environment', 'More', and 'Approve'. The 'Approve' button is highlighted with an orange box and an arrow pointing to the right. Below the buttons, there is a section for 'Comments on pull request (0)'. The main part of the screenshot shows a code diff for 'src/router/index.ts'. The diff is split into two columns: the left column shows the original code with lines 13-18 highlighted in red (indicating deletions), and the right column shows the proposed changes with lines 13-15 highlighted in green (indicating additions). The changes include updating the route path from '/about' to '/thread' and adding a new component import for 'ThreadShow.vue'.

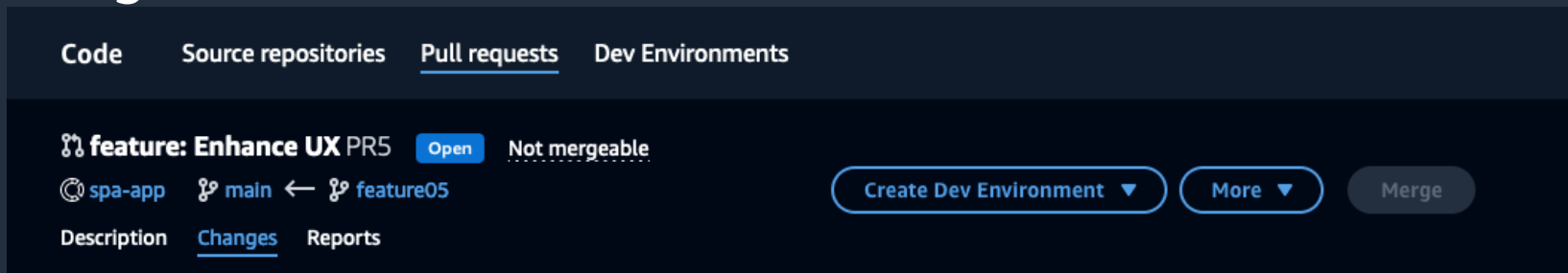
The screenshot shows a dialog box titled 'Approve PR5: feature: Enhance UX?'. The dialog contains the text: 'You are about to approve this pull request. Approvals can be revoked until the pull request is merged.' Below this, there is a section for 'Approval comment - optional' with the instruction 'Leave a comment about the author's work.' There is a text input field containing the text 'LGTM'. At the bottom right of the dialog, there are two buttons: 'Cancel' and 'Confirm'.

Pull Request の Merge

Merge 可能な状態



Merge 不可能な状態



- 考えられる理由:
 - Required レビューが Approve していない
 - コンフリクトが発生している -> ローカルリポジトリで解決して Push

Pull Request の Merge

Merge strategy
Choose how your merge will be made to the destination branch.

Fast forward merge
`git merge --ff-only`
Merges the branches and moves the destination branch pointer to the tip of the source branch. This is the default merge strategy in Git.

Squash and merge
`git merge --squash`
Combines all commits from the source branch into single merge commit in the destination branch.

Delete the source branch after merging this pull request. Source branch: [feature02](#)

Cancel Merge

- Merge 戦略は 2 種類から選択できる (Fast forward or Squash)
- 他の戦略 (例: Merge Commit) を採用する場合は git コマンドを利用してローカルリポジトリで Merge 後 push

Merge strategy
Choose how your merge will be made to the destination branch.

Fast forward merge
`git merge --ff-only`
Merges the branches and moves the destination branch pointer to the tip of the source branch. This is the default merge strategy in Git.

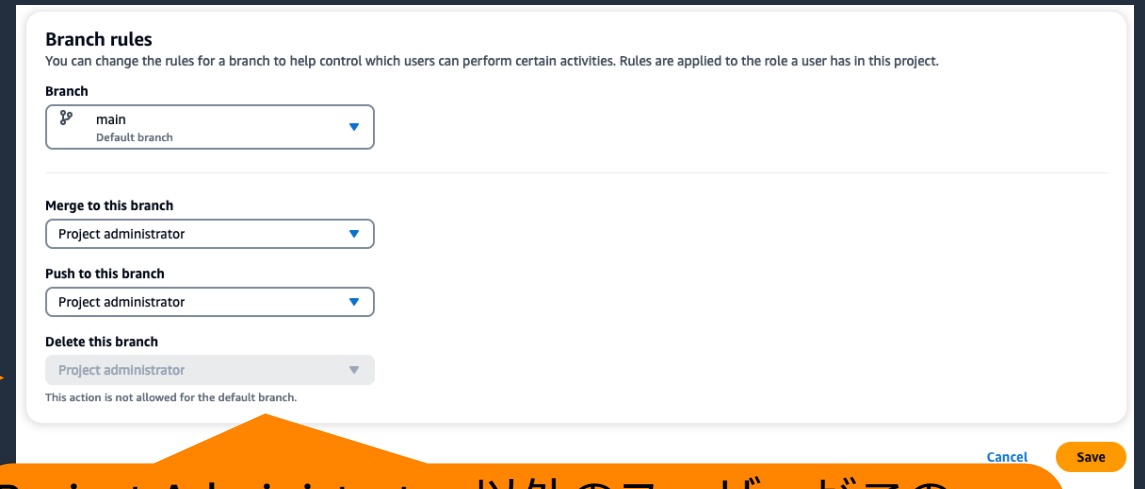
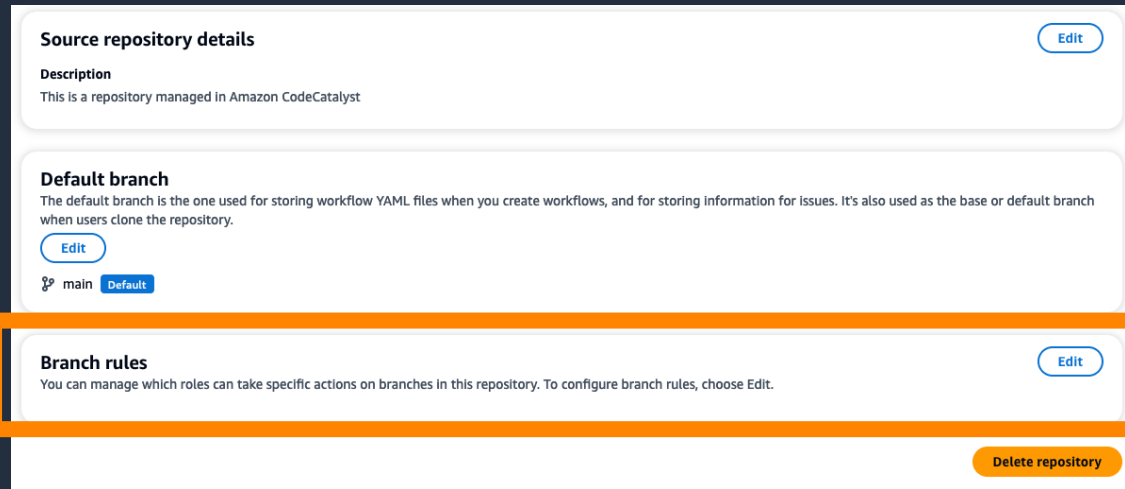
Squash and merge
`git merge --squash`
Combines all commits from the source branch into single merge commit in the destination branch.

- Branch の状態次第では Fast forward merge が選択できないことがある
- 解決方法は Git ドキュメントを参照 (例: Source Branch を Rebase する)

Branch ルール

Branch ルール

- Branch ごとに運用ルールを設定
 - Branch に Pull Request を Merge できる Role
 - Branch に直接変更を Push できる Role
 - Branch を削除できる Role（Default Branch 以外の場合に指定可能）



リポジトリの設定画面もしくは Branch 一覧画面から Branch ルール設定画面に遷移

Project Administrator 以外のユーザーがこの Branch に変更を反映するためには、Pull Request を作成し、Project Administrator に Merge を依頼する必要がある

通知



Slack Channel への通知

- Project 単位で設定（Project settingから設定）
- イベントごとに ON/OFF を切り替えることができる
- 複数 Channel に同時に通知することができる

通知できるリポジトリイベント

Pull request

created/revised, approved, comment added,
edited, closed, merged

E-Mail への通知

- Space × 個人単位で設定（My settingsから設定）
- イベントごとに ON/OFF を切り替えることができる

通知できるリポジトリイベント

Pull request

Created, Edited, Approved, Merged, Closed, Comment Added,
Assigned as a required reviewer,
Assigned as a optional reviewer

クォータ

クォータ

項目	上限
リポジトリ数	1 Space あたり 5,000
リポジトリサイズ	5GB Space のストレージ容量上限の影響も受ける
Git blob サイズ	ファイルの数や単一の Commit でのファイルの合計サイズは無制限 メタデータは 6MB 以下、単一の blob ファイルのサイズは 2GB まで
コンソールから参照・操作できるファイルのサイズ	6MB
一つの Pull Request にアサインできるレビュー数	100 optional, required 合算

詳細は公式ドキュメントを参照してください

<https://docs.aws.amazon.com/codecatalyst/latest/userguide/source-quotas.html>



Thank you!