



Amazon CodeCatalyst

Projects, Blueprints 編

Ryuji Hori

Solutions Architect
2023/12

自己紹介

名前：堀 竜慈

所属：アマゾンウェブサービスジャパン合同会社
西日本エンタープライズソリューション第二部
ソリューションアーキテクト

経歴：日系大手企業の SIer、
ベンチャー企業の フルスタックエンジニア

好きなAWSサービス：Amazon CodeCatalyst , Amazon SageMaker シリーズ

好きなアイス：ビスケットサンドアイス



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

Projects, Blueprints 編の対象読者

- Project の全体像を把握されたい方
- Blueprint を用いて効率よく Project を作成/管理したい方

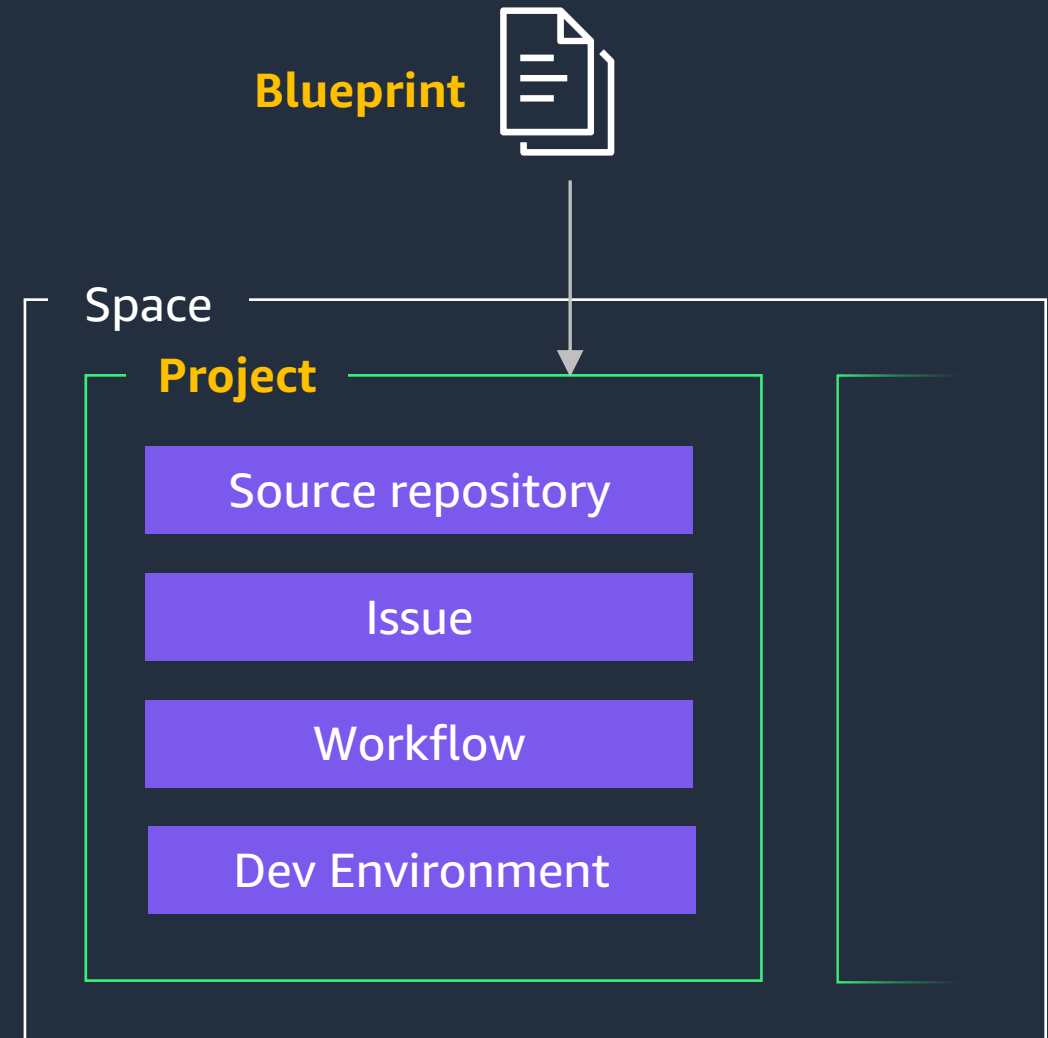
アジェンダ

1. Project の概要
2. Blueprint の概要
3. Project の構成要素
4. Project の作成
5. CodeCatalyst blueprint を用いた Project の作成手順
6. Project member の管理
7. Custom Blueprint 詳細
8. Custom Blueprint 作成/利用手順 概要
9. Custom Blueprint 更新/更新適用手順 概要
10. 通知
11. Project の削除
12. クォータ

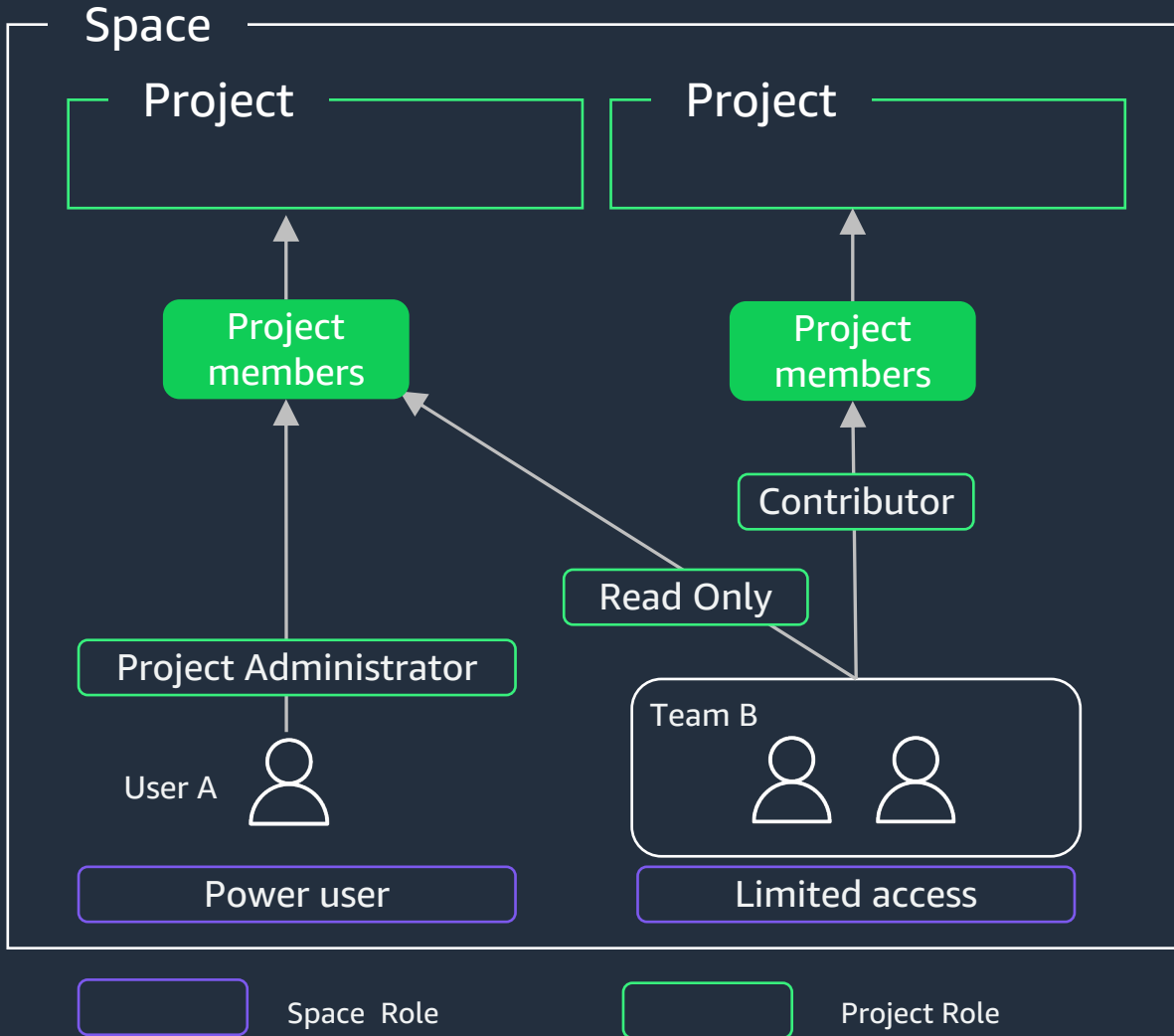
Project の概要

Space と Project の概要

- Space
 - 部署、グループ、会社などの**組織を表現**する概念
 - Project、メンバー、Extension、AWS Account Connection、VPC Connection の管理を行う
- Project
 - リモート開発環境、Git リモートリポジトリ、CI/CD パイプライン、Issue 管理など**チーム開発に必要なコンポーネントがカプセル化されたコラボレーション空間**
 - Project は1つの Space に関連付けられ、Space は複数の Project を所持できる
 - 空の状態で作成することも、Blueprint (後述)を利用して作成することも可能



Project member について



- User、Team (※) は複数の Project に Project member として参加することが可能

- Project member は Space ごと、Project ごとに与えられた Role により、各 Project で実行可能な操作が定められる(※)

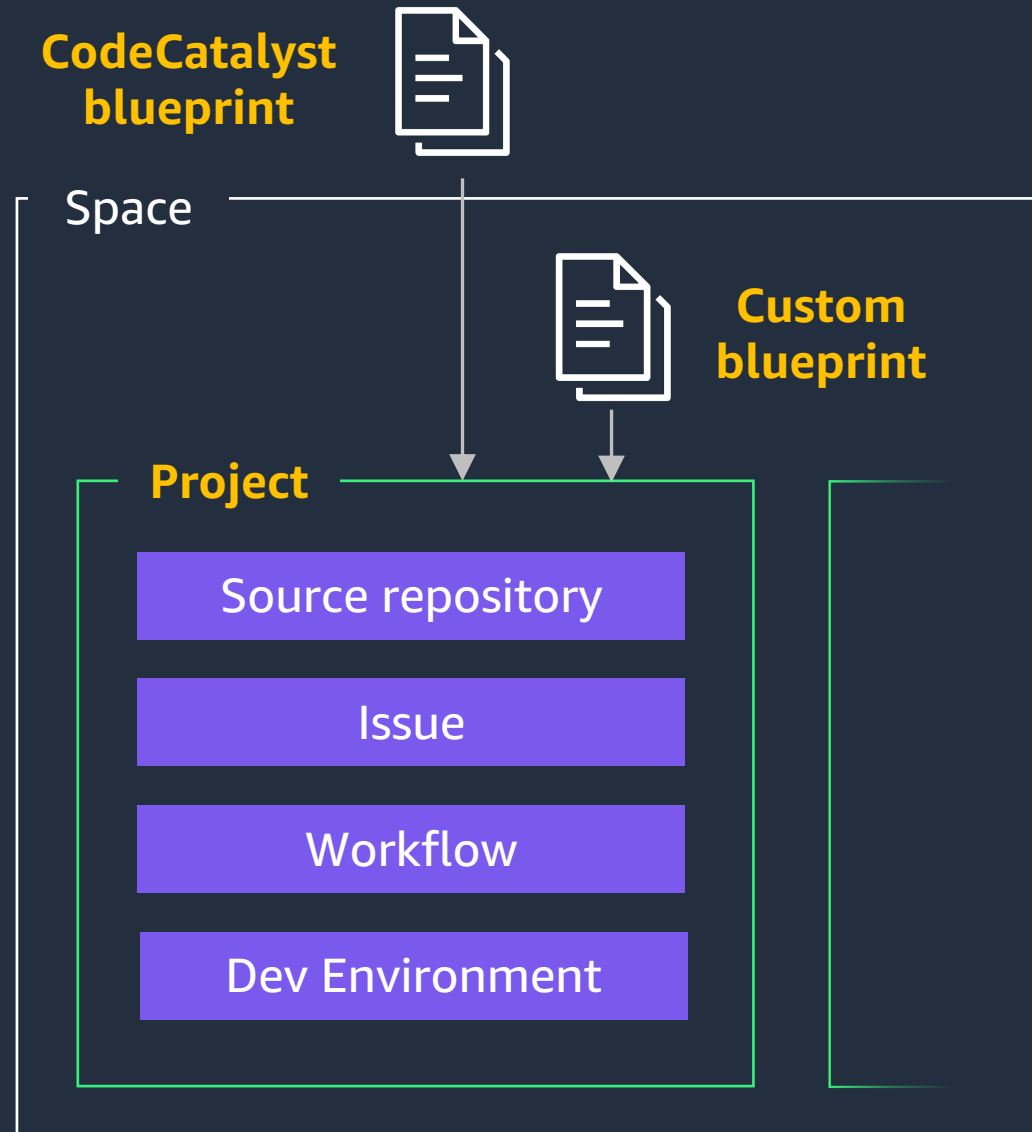
- 例えば、Project Administrator Role を持つ Project member は、対象 Project のメンバーの Project Role 管理 や User の招待が可能
 - Space/Project レベルで Role による User の権限管理が可能
 - Project Role の詳細は後述

※Team、Role の詳細については [Black Belt Amazon CodeCatalyst シリーズ Identity, permissions, and access 編](#)参照

Blueprint の概要

Blueprint について

- Project のテンプレート
- Project におけるアプリケーションコードやワークフロー、インフラの構成を定義
 - ▶ すぐに開発が始められる
- 使用した Blueprint は Project に紐付けられ、Blueprint の変更は Pull Request として展開後も伝播される
 - Blueprint の実体は、projen というOSSツールを用いてソースコードから合成されたものであり、そのため更新管理が行いやすい
- 定義済みの **CodeCatalyst blueprint**、独自に定義できる **Custom blueprint** の2種類に大別される
- 1つの Project に複数の Blueprint 適用可能




CodeCatalyst blueprint について

- Amazon CodeCatalyst により定義済みの Blueprint
- AWS 上でよくあるアプリケーション開発プロジェクトを支援
 - シングルページアプリケーション、Web 3 層構造、REST API、データの ETL 処理、VOD ウェブサービス、...
 - 18のテンプレートが展開済み (2023/12/12時点)

Modern three-tier web application >

About this blueprint

This Blueprint generates code in Python for the application layer and Vue front-end framework to build and deploy a well-architected 3-tier modern web application. This blueprint was inspired by the [Mythical Mysfits website](#). Mythical Mysfits have been used in several public [AWS workshops](#) and [Game Days](#). These workshops and game days seek to educate and demonstrate to our customers what is possible on AWS; through this Blueprint we seek to do the same.

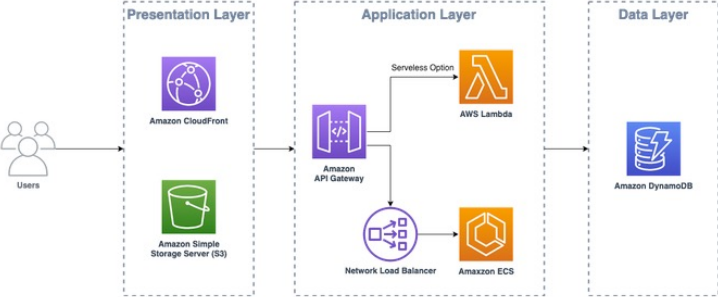


Architecture overview

This blueprint provisions a three-tier web application architecture. The resulting project creates the following AWS resources:

- AWS Lambda function or an Amazon Elastic Container Service (Amazon ECS) Fargate cluster and service.
- HTTP API Gateway (ECS Option)
- Amplify Hosting App or S3 Bucket + Cloudfront Distribution
- Amazon DynamoDB database

The project's deployment status can be viewed in the project's workflow. For the Lambda compute type, it might take up to 15 minutes to deploy.



Custom blueprint について

- ユーザにより独自に定義された **Blueprint**
- チームのベストプラクティスを定義して推進することを支援
- アプリケーションコード、ワークフローなど、さまざまな側面を定義可能
- Custom blueprint を更新した場合、Pull request として既存の Project に変更を伝播可能
 - ▶ 最新のベストプラクティスの反映が容易
- Space 内のメンバーに共有される (詳細後述)
- 利用するには **Enterprise Tier** が必要

The screenshot shows the 'Create blueprint' interface in AWS CodeCatalyst. The main form has the following sections:

- Blueprint name:** A text input field containing 'react-app-blueprint'. Below it, a note states: 'Blueprint names must be unique within a space. Names must be between 3 and 63 characters.'
- Blueprint details:** A section with a note: 'These details will be visible to the space users once the blueprint is published. You can change these details in the code repository of the project at a later time.'
- Blueprint Display Name:** A text input field containing 'react-app-blueprint'. Below it, a note asks: 'What do you want to call your new blueprint?'
- Description:** A text input field containing 'This blueprint creates an empty application.' Below it, a note asks: 'Add a description for your new blueprint.'
- Author name:** A text input field containing 'CodeCatalystBB'. Below it, a note asks: 'Who is the author of the blueprint?'

On the right side, there is a 'Blueprint Builder' sidebar showing a file explorer for the 'react-app-blueprint' project. The structure includes:

- react-app-blueprint
 - .codecatalyst
 - .projen
 - deps.json
 - files.json
 - tasks.json
 - src
 - wizard-configurations
 - blueprint.ts
 - defaults.json
 - index.ts
 - static-assets
 - .eslintrc.json
 - .gitattributes
 - .gitignore
 - .npmignore
 - .nvmrc
 - .projenrc.ts

Project の構成要素

Project の構成要素 – 全体

- Code 関連
 - Source repository
 - Pull requests
 - Dev Environment
- CI/CD 関連
 - Workflows
 - Environments
 - Compute
 - Secrets
- Issue
- Report
- Package
- Blueprint

※各項目の詳細な解説は [Black Belt Amazon CodeCatalyst シリーズ](#) の他編をご参照ください (一部未対応)

Project の構成要素 – Code 関連

- Source repository
 - Project 用の Git リモートリポジトリ
 - CodeCatalyst 内に作成、または GitHub と接続が可能
- Pull request
 - Branch 間のコード変更を、他の Project member とレビュー、コメント、Merge するための機能
- Dev Environment
 - CodeCatalyst で使用できるクラウドベースの開発環境

Project の構成要素 – CI/CD 関連 (1/2)

- Workflows
 - コードをビルド、テスト、デプロイするパイプラインを定義・実行する
 - GUI でも YAML 形式でも定義可能
- Environments
 - Workflows が連携可能な AWS 環境を設定
 - VPC への接続 (VPC Connection) の設定も行える

Project の構成要素 – CI/CD 関連 (2/2)

- Compute
 - Workflow 実行に用いられる CPU、メモリ、OS の設定情報
 - 設定変更するには Standard tier 以上の tier へ加入が必要
- Secrets
 - 認証情報のような機密データを管理 (例: DB パスワード)
 - Workflows から参照することが可能

Project の構成要素 – その他

- Issues
 - Project 内の課題管理機能
 - CodeCatalyst 内に作成、または Jira と接続が可能
- Reports
 - Workflows にて実施したテスト、コードカバレッジ、ソフトウェア構成解析（SCA）等の品質レポート

Project の構成要素 – その他

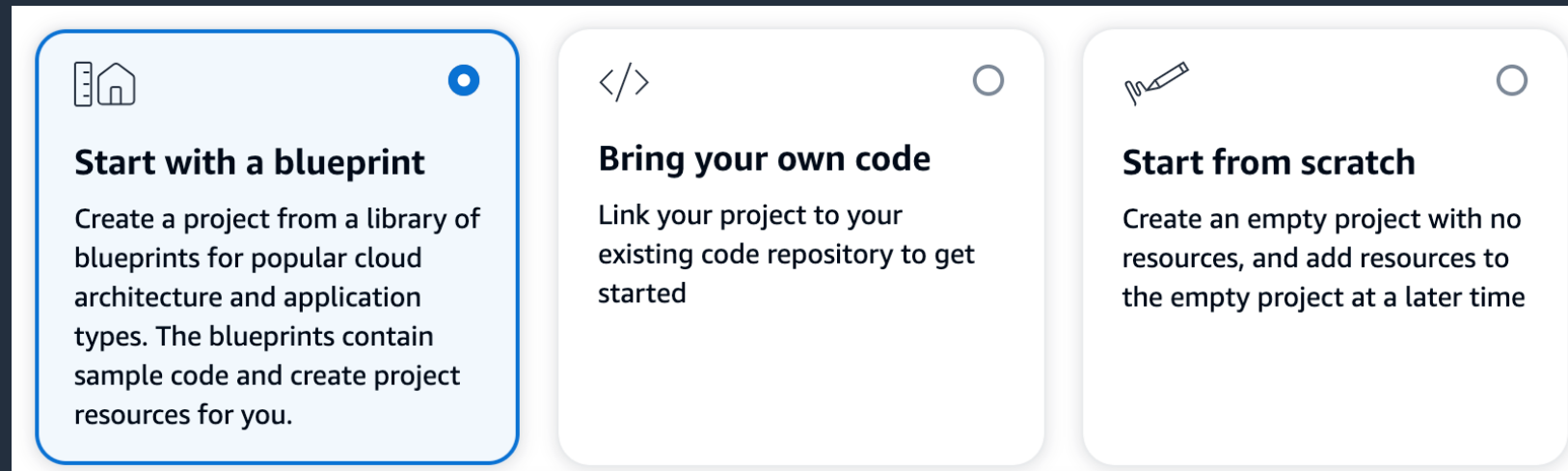
- Package
 - フルマネージドなパッケージリポジトリサービス
 - ソフトウェアパッケージを安全に保存、共有可能
- Blueprint(※)
 - Project に使用されている Blueprint の情報
 - 対象の Blueprint のアップデート通知を受け取り、Project へ反映できる
 - Blueprint の追加、更新、削除、設定変更も可能

※ Project の構成要素としての Blueprint

Project の作成

Project 作成方法の種類

- Project 作成するには3種類の方法がある
- ① Blueprint を用いる
- ② 既存のソースリポジトリを参照
- ③ 空の Project から始める



The screenshot displays three options for creating a project in AWS, each with an icon and a radio button. The first option, 'Start with a blueprint', is selected and highlighted with a blue border. The second option is 'Bring your own code', and the third is 'Start from scratch'.

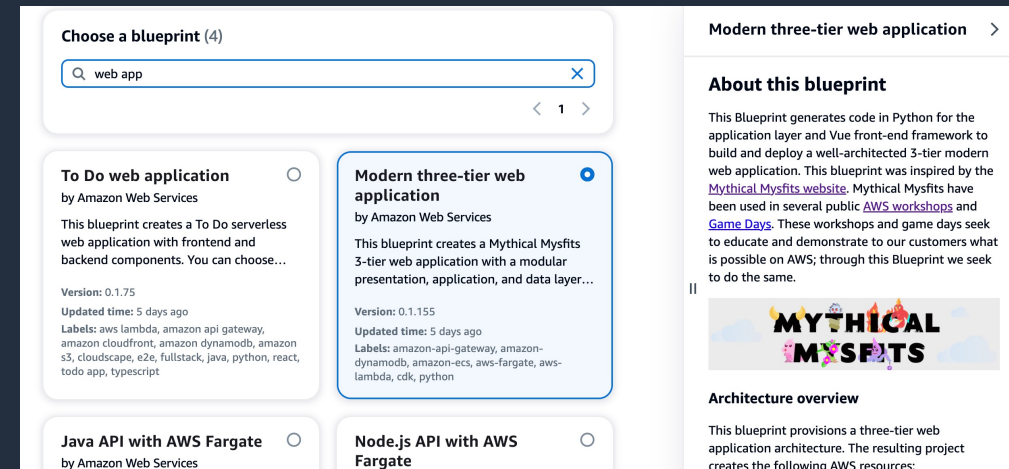
Option	Icon	Radio Button	Description
Start with a blueprint	House and checklist	Selected (filled)	Create a project from a library of blueprints for popular cloud architecture and application types. The blueprints contain sample code and create project resources for you.
Bring your own code	Code symbols	Unselected (empty)	Link your project to your existing code repository to get started
Start from scratch	Pencil	Unselected (empty)	Create an empty project with no resources, and add resources to the empty project at a later time

Project 作成方法① – Blueprint を用いる

- 手軽に環境構築したり、チーム独自のベストプラクティスを適用して構築/管理したい場合におすすめ

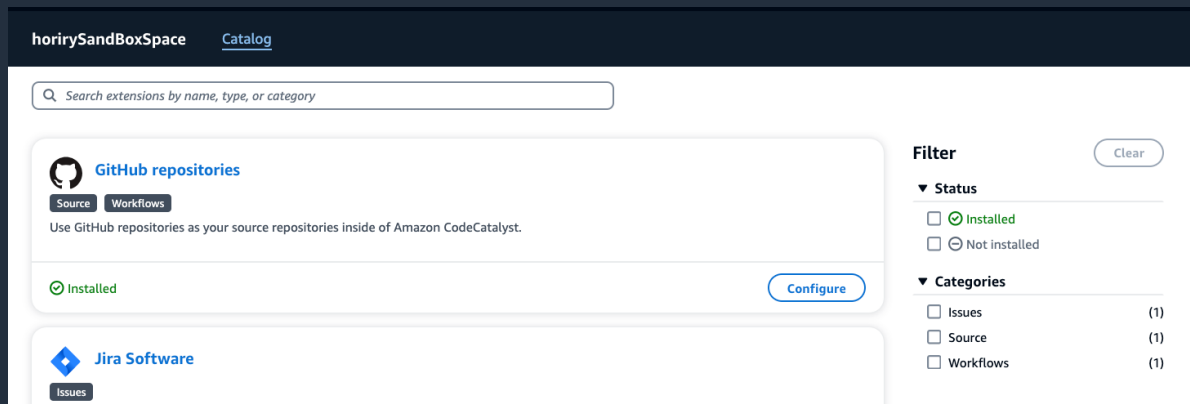
▶ Project の設定にかかるオーバーヘッドを削減

- Project 作成後も、採用した Blueprint への変更が伝播されるため、最新のベストプラクティスを保ちやすい
- 用意されている多様な CodeCatalyst blueprint や、独自の Custom blueprint を選択可能



Project 作成方法② – 既存のソースリポジトリを参照

- 既存のリポジトリを用いてプロジェクト開発したい場合に推奨
- 既存のリポジトリを Source repository として参照した上で、Project を作成
- 現在は GitHub リポジトリのみ対応しており、事前に GitHub repositories Extensions のインストールが必要
- Project 作成後に、既存リポジトリを参照することも可能



※ Space Administrator Role もしくは Power User Role が必要

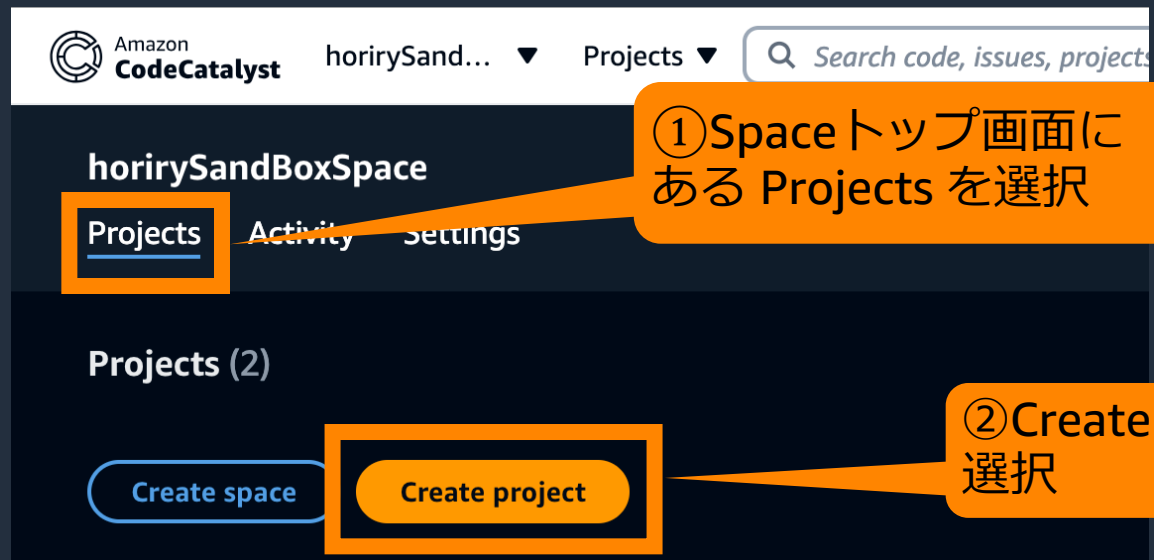
※Extensions の詳細については Black Belt Amazon CodeCatalyst シリーズ Extensions 編参照

Project 作成方法③ – 空の Project から始める

- 自身の開発プロジェクトに沿った Project を0から構築したい場合に使用
- 何も構築されていない Project をスクラッチ作成
- この方法を用いる場合は、Blueprint により作成された Project の内容を参考にしながら構築することを推奨

CodeCatalyst blueprint を用いた Project の作成手順

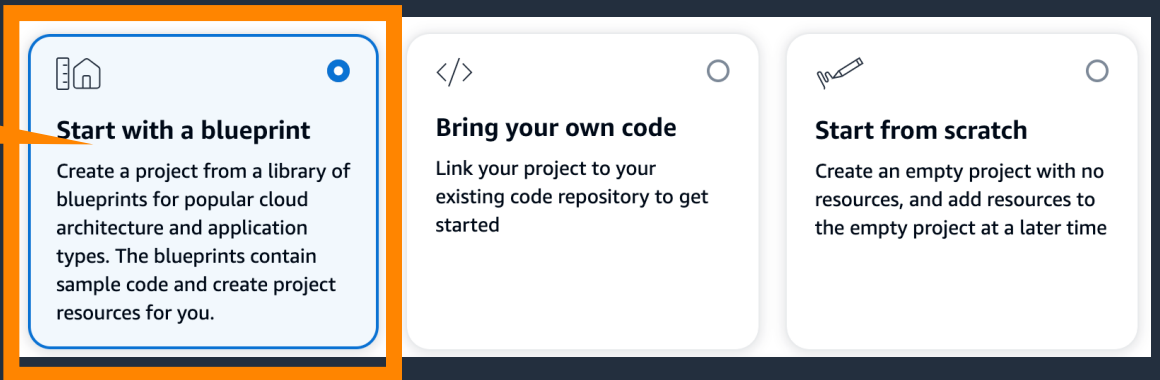
CodeCatalyst blueprint を用いたプロジェクト作成手順 (1/5)



① Space トップ画面にある Projects を選択

② Create project を選択

③ Start with a blueprint を選択



※ Space Administrator Role もしくは Power User Role が必要

CodeCatalyst blueprint を用いたプロジェクト作成手順 (2/5)

Choose a blueprint (4)

web app

To Do web application by Amazon Web Services
This blueprint creates a To Do serverless web application with frontend and backend components. You can choose Java, TypeScri...
Version: 0.1.74
Updated time: 3 days ago
Labels: aws lambda, amazon api gateway, amazon cloudfront, amazon dynamodb, amazon s3, cloudscape, e2e, fullstack, java, python, react, todo app, typescript

Modern three-tier web application by Amazon Web Services
This blueprint creates a Mythical Mysfits 3-tier web application with a modular presentation, application, and data layers. The application...
Version: 0.1.154
Updated time: 3 days ago
Labels: amazon-api-gateway, amazon-dynamodb, amazon-ecs, aws-fargate, aws-lambda, cdk, python

Java API with AWS Fargate by Amazon Web Services
This blueprint creates a containerized Java web application backed by Amazon DynamoDB using AWS Fargate.
Version: 1.0.10
Updated time: 2 months ago
Labels: container, copilot, docker, dynamodb, ecs, java, spring

Node.js API with AWS Fargate by Amazon Web Services
This blueprint creates a containerized node.js web application backed by Amazon DynamoDB using AWS Fargate and AWS...
Version: 1.0.10
Updated time: 2 months ago
Labels: container, copilot, docker, ecs, node

① 使用したい Blueprint に関連しそうな単語を検索エリアに入力


② 表示された Blueprint の説明を参考に Blueprint を選択

CodeCatalyst blueprint を用いたプロジェクト作成手順 (3/5)

Modern three-tier web application >

About this blueprint

This Blueprint generates code in Python for the application layer and Vue front-end framework to build and deploy a well-architected 3-tier modern web application. This blueprint was inspired by the [Mythical Mysfits website](#). Mythical Mysfits have been used in several public [AWS workshops](#) and [Game Days](#). These workshops and game days seek to educate and demonstrate to our customers what is possible on AWS; through this Blueprint we seek to do the same.

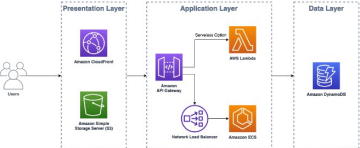


Architecture overview

This blueprint provisions a three-tier web application architecture. The resulting project creates the following AWS resources:

- AWS Lambda function or an Amazon Elastic Container Service (Amazon ECS) Fargate cluster and service.
- HTTP API Gateway (ECS Option)
- Amplify Hosting App or S3 Bucket + Cloudfront Distribution
- Amazon DynamoDB database

The project's deployment status can be viewed in the project's workflow. For the Lambda compute type, it might take up to 15 minutes to deploy.



① 選択した Blueprint の概要や、Blueprint によりデプロイされるサービスのAWSアーキテクチャが表示されるので確認

Connections and permissions

This blueprint requires a custom development role. To create one, click "Add an existing IAM role" from the add IAM role options. The IAM role needs to contain the CodeCatalyst trust policy, as well as the following permissions:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudformation:*",
        "ecr:*",
        "ssm:*",
        "s3:*",
        "amplify:*",
        "codeguru-reviewer:*",
        "iam:Get*",
        "iam:PassRole",
        "iam:CreateRole",
        "iam>DeleteRole",
        "iam:TagRole",
        "iam:UpdateRole",
        "iam:AttachRolePolicy",
        "iam:DetachRolePolicy",
        "iam:PutRolePolicy",
        "iam:CreatePolicy",
        "iam:CreateServiceLinkedRole",
        "iam>DeletePolicy",
        "iam:CreatePolicyVersion",
        "iam>DeletePolicyVersion",
        "iam:PutRolePermissionsBoundary",
        "iam>DeleteRolePermissionsBoundary",
        "sts:AssumeRole",
        "sts:GetCallerIdentity"
      ],
      "Resource": "*"
    }
  ]
}
```

② AWS アカウントに接続する際に必要な、IAM ロールの権限や信頼ポリシーの情報も確認

③ 確認を終えたら Next を選択

Cancel **Next**

CodeCatalyst blueprint を用いたプロジェクト作成手順 (4/5)

Project name
Name your project
You cannot change a project's name after it has been created.

Modern-horiry-app

Project names must be unique within a space. Names must be between 3 and 63 characters.

Project resources
This blueprint creates the following project resources

AWS accounts
Account
The account where your project is deployed.

[AWS Account ID]

Looking for a different AWS account? [Add an AWS account.](#)

Deployment Role
CodeCatalyst deployment role [View role requirements](#)

CodeCatalystWorkflowDevelopmentRole-[IAM Role Name]

Expected role capabilities codecatalyst*. [Create or add an AWS role](#)

① Project 名を入力

② Space に登録済みの、
AWS アカウント一覧が表示される(※)
→ 接続する AWS アカウント ID を選択

③ Space に登録済みの、選択した AWS アカウントの
IAM ロール一覧が表示される(※)
→ 接続に使用する IAM ロールを選択 / 新規作成

※ Space への AWS アカウントと IAM ロールの連携については Black Belt Amazon CodeCatalyst シリーズ Space 編参照

CodeCatalyst blueprint を用いたプロジェクト作成手順 (5/5)

Compute platform
Compute Platform

Lambda
 ECS Fargate

Frontend Hosting Option
Services used to host frontend

Amplify Hosting
 CloudFront + S3

Deployment Region
Enter the Deployment Region

us-west-2

Application name
Please add a name for your application

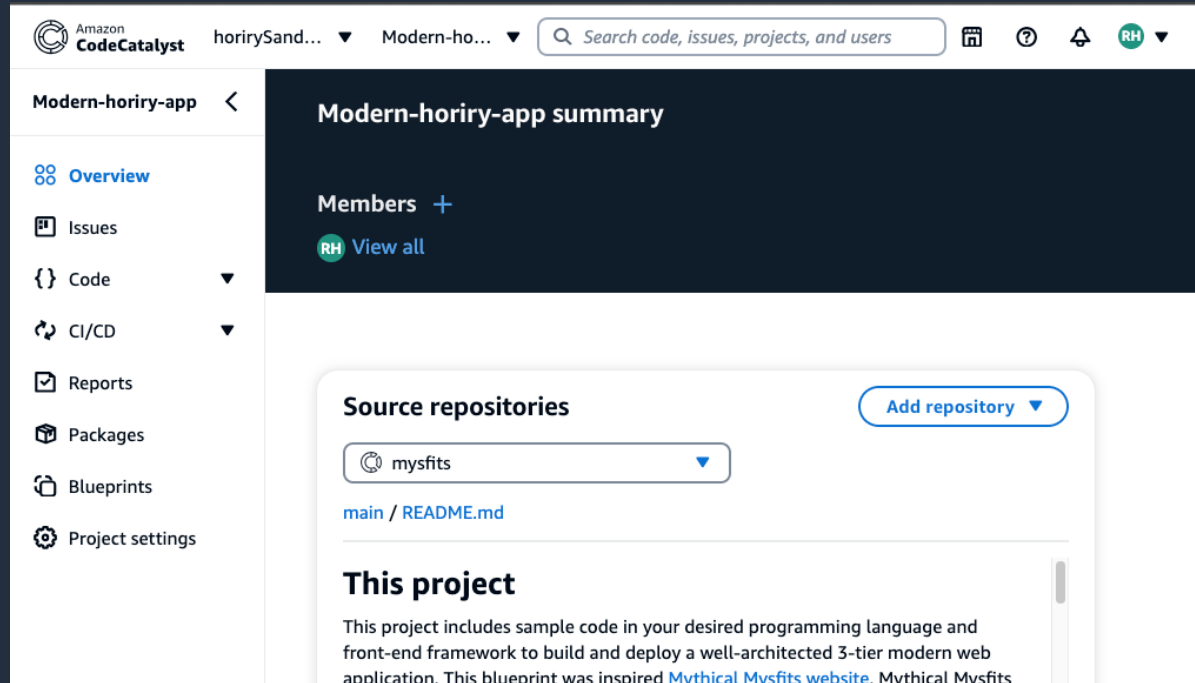
mysfitsv412p

①各設定項目を入力していく
(表示される設定項目は選択した
Blueprint により異なる)

②すべての入力を終わったら、
Create project を選択

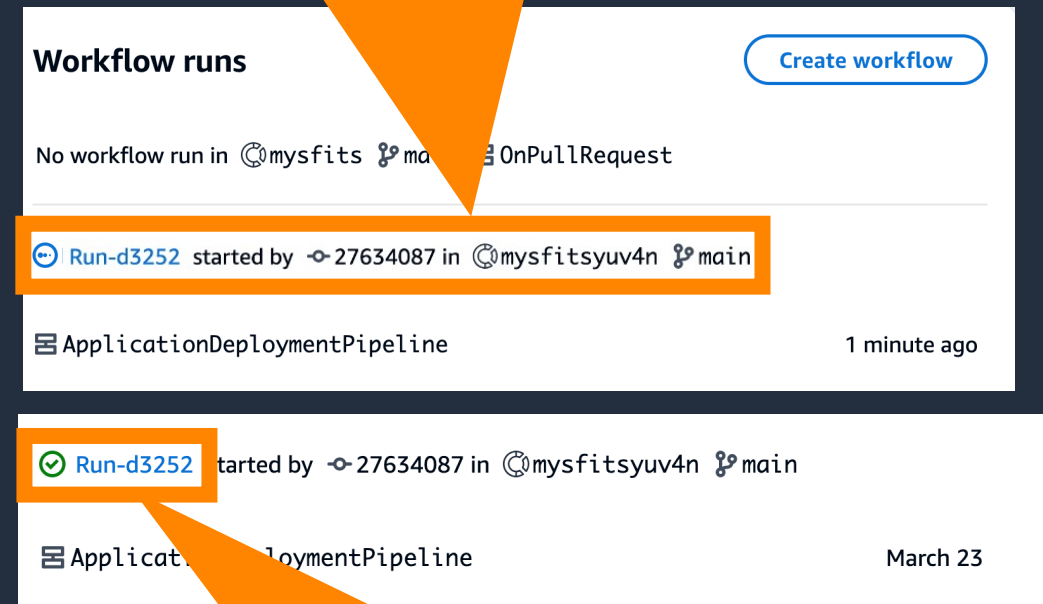
Cancel Back **Create project**

CodeCatalyst blueprint を用いたプロジェクト確認手順 (1/2)



①しばらくすると Project の作成が完了し、Project の概要画面が表示される

②Project 作成が完了すると、デプロイのための Workflow が自動で実行される(数分待つと完了する)



③Workflow の実行が完了したら、成功した Run のリンクを選択する

CodeCatalyst blueprint を用いたプロジェクト確認手順 (2/2)

☰ Run-d3252

Visual

YAML

Artifacts

Reports

Variables

① Run の Variables を選択

Variable name ▾

Value ▾

endpointUrl

https://[REDACTED].cloudfront.net

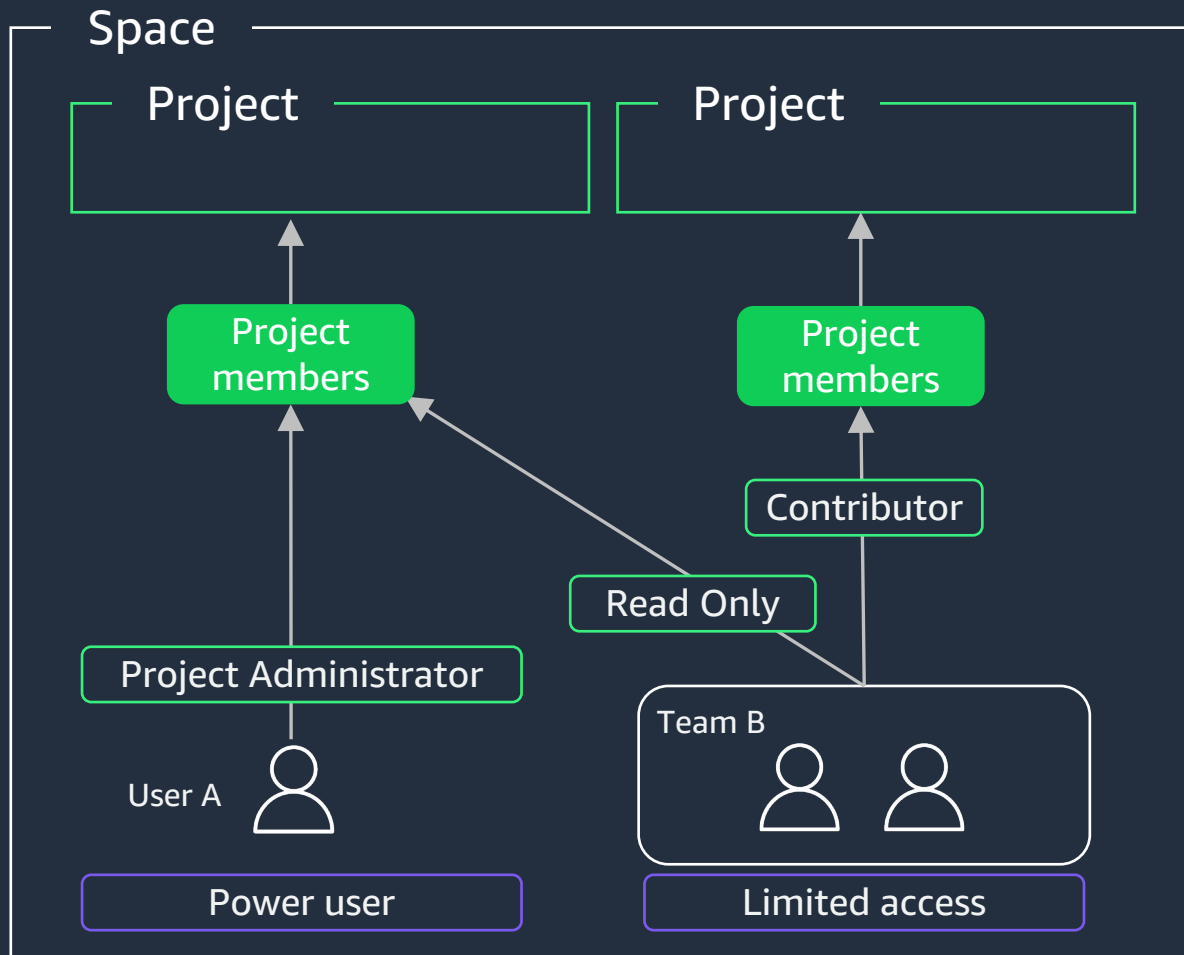
② Variables 項目から、アプリケーションの動作を確認するための情報 (この例ではURL) を取得

③ デプロイされたアプリケーションの動作を確認



Project member の管理

Project member について



Space Role

Project Role

- User、Team (※) は複数の Project に Project member として参加することが可能

Project member は Space ごと、Project ごとに与えられた Role により、各 Project で実行可能な操作が定められる(※)

例えば、Project Administrator Role を持つ Project member は、対象 Project のメンバーの Project Role 管理や User の招待が可能

- Project Role の詳細は後述

※Team、Role の詳細については [Black Belt Amazon CodeCatalyst シリーズ Identity, permissions, and access 編](#) 参照

Project Role

<https://docs.aws.amazon.com/codecatalyst/latest/userguide/ipa-roles.html>

Role	Description
Project Administrator	<ul style="list-style-type: none">Project に対して最も強力な権限をもつ RoleProject 設定の編集、Project への User 招待、User の Project 権限管理、Project の削除などあらゆる操作が可能
Contributor	<ul style="list-style-type: none">Project に参加する多くのメンバーが持つことを想定された Roleコード、Workflow、Issue などを操作できる
Reviewer	<ul style="list-style-type: none">Pull Request のレビューや Issue の管理を目的として Project にアクセスする User が持つことを想定された Roleコードや Workflow に対しては参照のみ可能
Read Only	<ul style="list-style-type: none">Project リソースの参照のみが可能な Role

※ 詳細については [Black Belt CodeCatalyst シリーズ Identity, permissions, and access 編](#) 参照

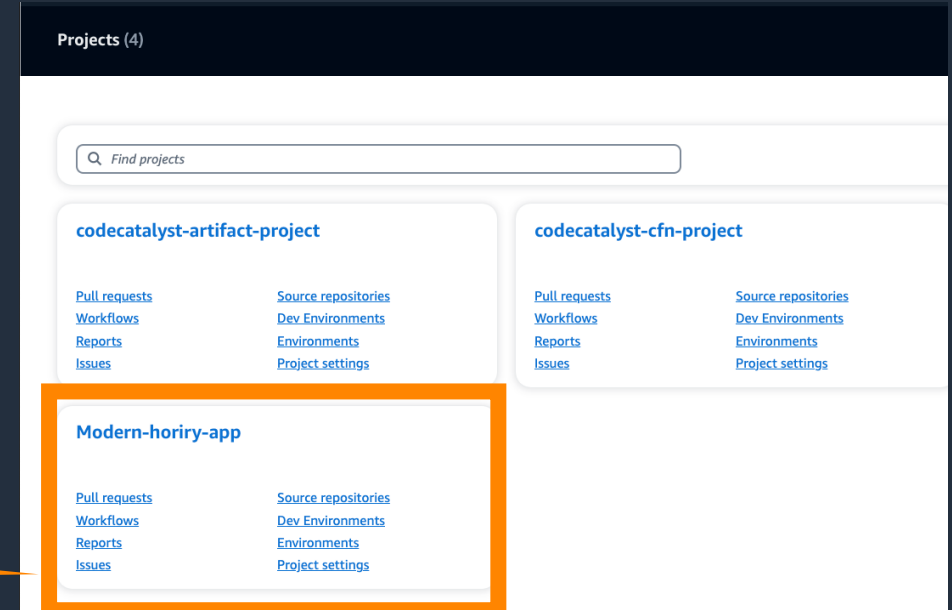
Project member 招待の手順 (1/2)

～Space 外部のメンバー招待～

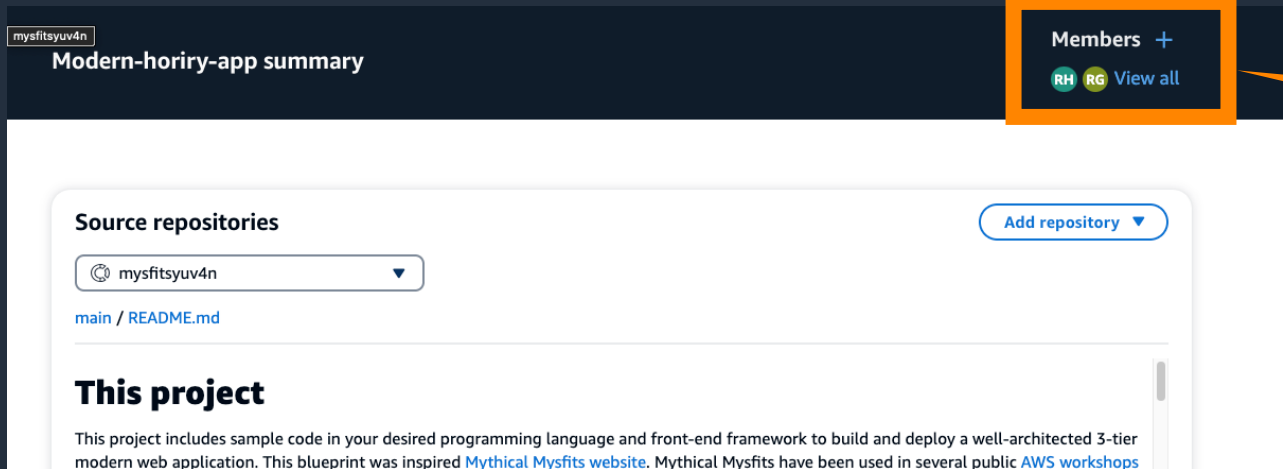
※ 本手順は Builder ID User をサポートする Space における手順となっています。IAM Identity Center を用いた ID フェデレーションをサポートする Space における手順とは多少異なります。

https://docs.aws.amazon.com/ja_jp/codecatalyst/latest/userguide/setting-up-topnode.html

① Project 一覧画面にて、メンバーを招待したい Project を選択



② Project Overview 画面にある "Members" の横にある "+" を選択

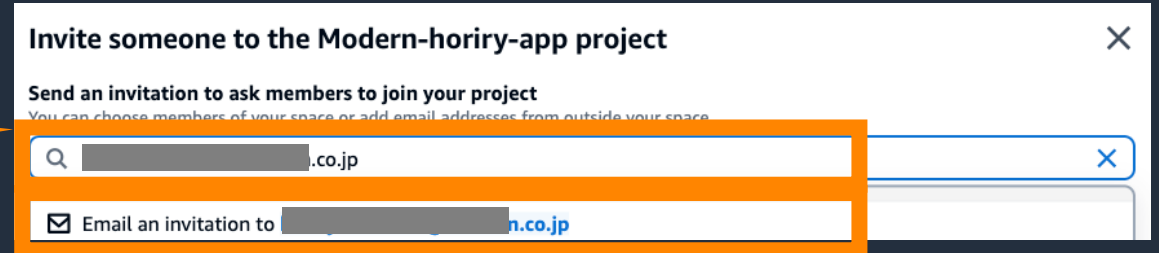


※ ユーザを招待するメンバーは、Project へユーザを招待するための権限を持つ Role が必要

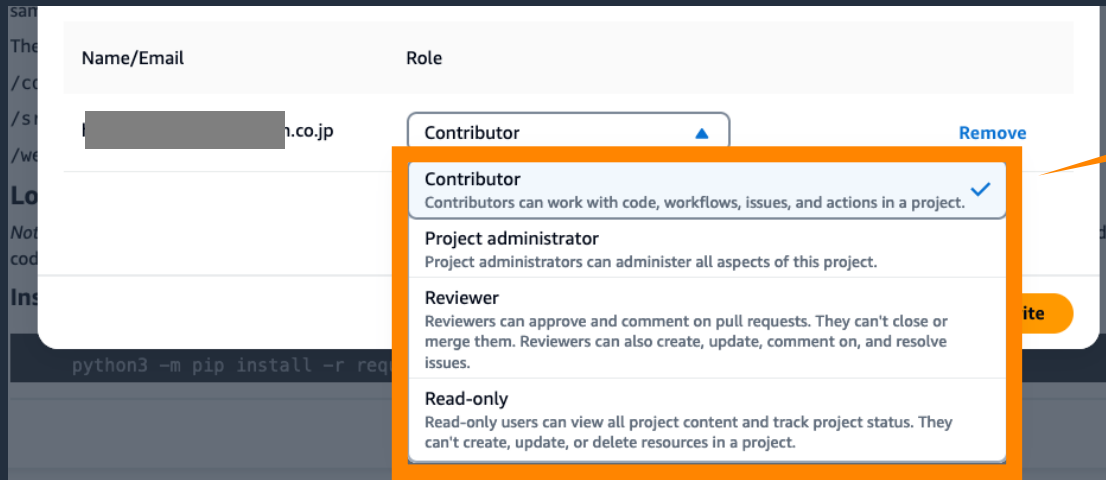
Project member 招待の手順 (2/2)

～Space 外部のメンバー招待～

① 「Send an invitation～」下にある入力欄に、招待したいメンバーの Email アドレスを入力



② 「Email an invitation to～」とメッセージが出てくるので、選択



Name/Email	Role	
[redacted]@n.co.jp	Contributor	Remove

- Contributor**
Contributors can work with code, workflows, issues, and actions in a project. ✓
- Project administrator**
Project administrators can administer all aspects of this project.
- Reviewer**
Reviewers can approve and comment on pull requests. They can't close or merge them. Reviewers can also create, update, comment on, and resolve issues.
- Read-only**
Read-only users can view all project content and track project status. They can't create, update, or delete resources in a project.

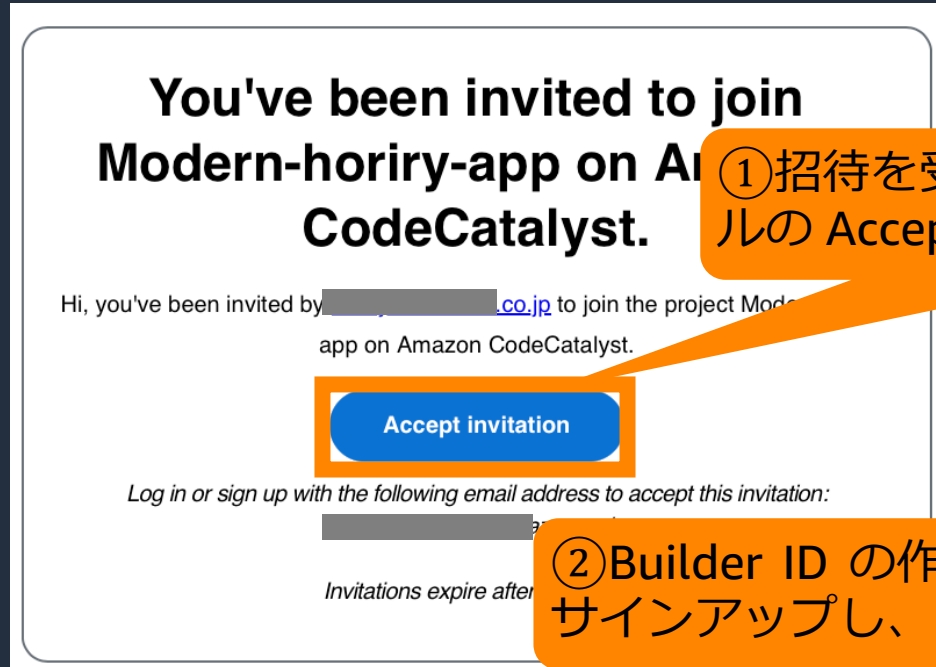
③ 招待するメンバーの Role を選択

④ Invite ボタンを選択



Project member 招待の受け入れ手順 (1/3)

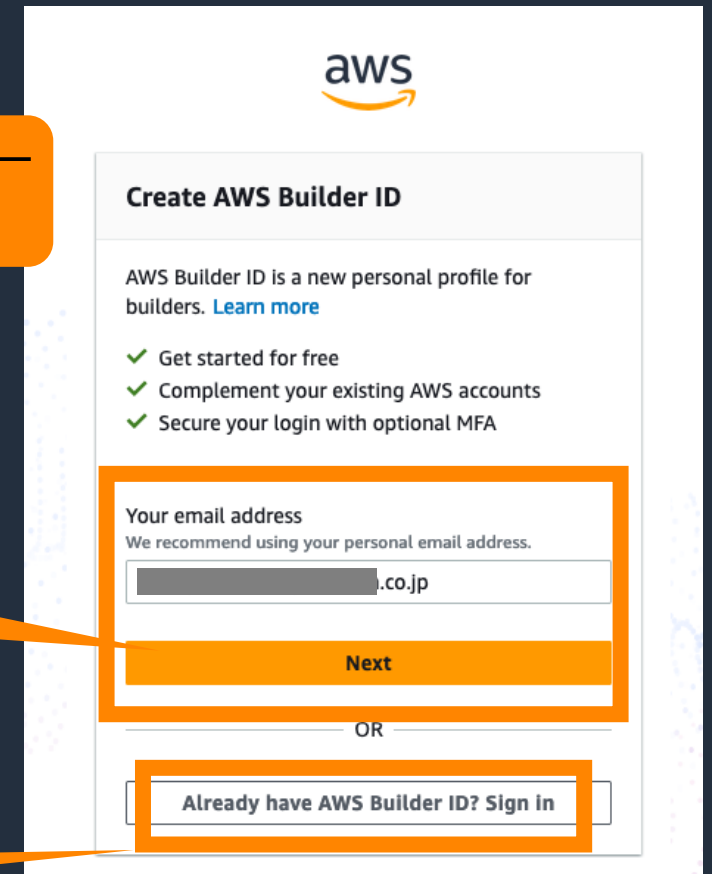
～Space 外部のメンバー招待～



① 招待を受け取ったユーザは、招待メールの Accept Invitation を選択する

② Builder ID の作成を求められるので、サインアップし、ログインする

②' 既にBuilder IDを所持している場合は、ログインする



Project member 招待の受け入れ手順 (2/3)

～Space 外部のメンバー招待～

The screenshot shows the 'Create your CodeCatalyst alias' page. At the top left, the Amazon CodeCatalyst logo is highlighted with an orange box. A callout points to this logo with the text: ① ログインすると、Amazon CodeCatalyst の画面に遷移する. Below the logo, the page title 'Create your CodeCatalyst alias' is followed by the instruction 'Choose a unique alias' and a note: 'Others can use your alias to @mention you in comments and pull requests.' A text input field contains the alias 'horryExternal' and is highlighted with an orange box. A callout points to this field with the text: ② 対象の Project と Space 内で用いるエイリアスを入力する. Below the input field, there is a 'Next' button, also highlighted with an orange box. A callout points to this button with the text: ③ Next を選択する.

Project member 招待の受け入れ手順 (3/3)

～Space 外部のメンバー招待～

①招待された Project のOverview画面が表示される

②"View all" を選択し、Project member 一覧画面へ移動する

③Project member 一覧画面から、自身の Role を確認できる

Modern-horiry-app summary

Members +

View all

Source repositories

mysfits

main / README.md

This project

Modern-horiry-app settings

General Members Notifications

Project members

View and manage members associated with this project.

Cancel invitation Remove Change role Invite

Name	Alias	Role	Email	Status
Yamada Tarou	@yamadaTarou	Contributor	[REDACTED]	Member
Hori demo	@horiryDemo	Contributor	[REDACTED]	Member
horiry external	@horiryExternal	Contributor	[REDACTED]	Member

Custom blueprint 詳細

Custom blueprint について

- ユーザにより独自に定義された **Blueprint**
- チームのベストプラクティスを定義して推進することを支援
- アプリケーションコード、ワークフローなど、さまざまな側面を定義可能
- Custom blueprint を更新した場合、Pull request として既存の Project に変更を伝播可能
 - ▶ 最新のベストプラクティスの反映が容易
- Space 内のメンバーに共有される (詳細後述)
- 利用するには **Enterprise Tier** が必要

Create blueprint

Blueprint name

Name your blueprint

Add a name for your new blueprint.

react-app-blueprint

Blueprint names must be unique within a space. Names must be between 3 and 63 characters.

Blueprint details

These details will be visible to the space users once the blueprint is published. You can change these details in the code repository of the project at a later time.

Blueprint Display Name

What do you want to call your new blueprint?

react-app-blueprint

Description

Add a description for your new blueprint.

This blueprint creates an empty application.

Author name

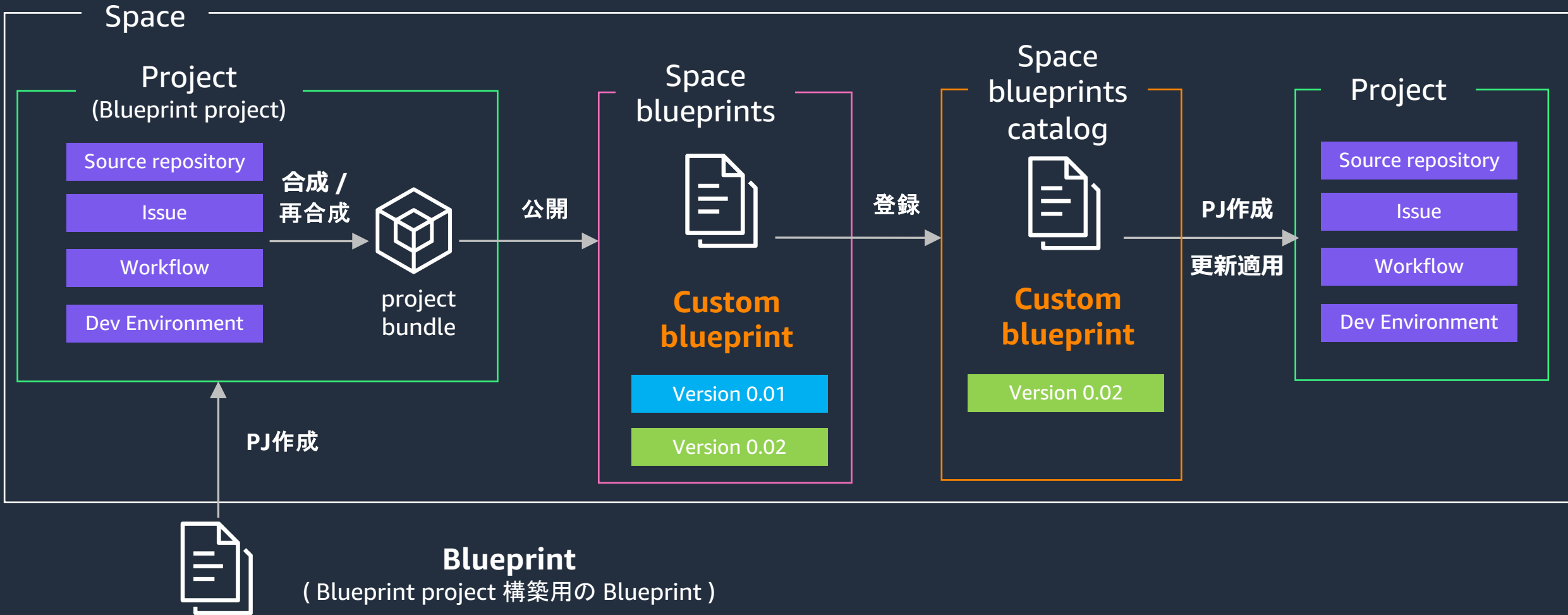
Who is the author of the blueprint?

CodeCatalystBB

Blueprint Builder

- ▼ react-app-blueprint
 - ▶ .codecatalyst
 - ▼ .projen
 - deps.json
 - files.json
 - tasks.json
 - ▼ src
 - ▶ wizard-configurations
 - blueprint.ts
 - defaults.json
 - index.ts
 - ▶ static-assets
 - .eslintrc.json
 - .gitattributes
 - .gitignore
 - .npmignore
 - .nvmrc
 - .projenrc.ts

Custom blueprint 概念図



※ Blueprint project の作成、 Custom blueprint の公開には、対象のSpace の Space Administrator Role、 Power User Role が必要

Custom blueprint 関連用語説明 (1/2)

- Blueprint project
 - Custom blueprint を開発し、Space へ公開するための Project
- Space blueprints
 - Blueprint project によって Space へ公開された Custom blueprint の一覧
 - Space blueprints catalog(後述) へ登録することで、Space 内のメンバーが対象の blueprint を利用可能になる
 - Custom blueprint はバージョン管理される
- Space blueprints catalog
 - Space 内のメンバーが Project に利用可能な Custom blueprint の一覧
 - Space blueprints にある Custom blueprint のうち、登録者が指定したバージョンのみが利用可能な状態となる

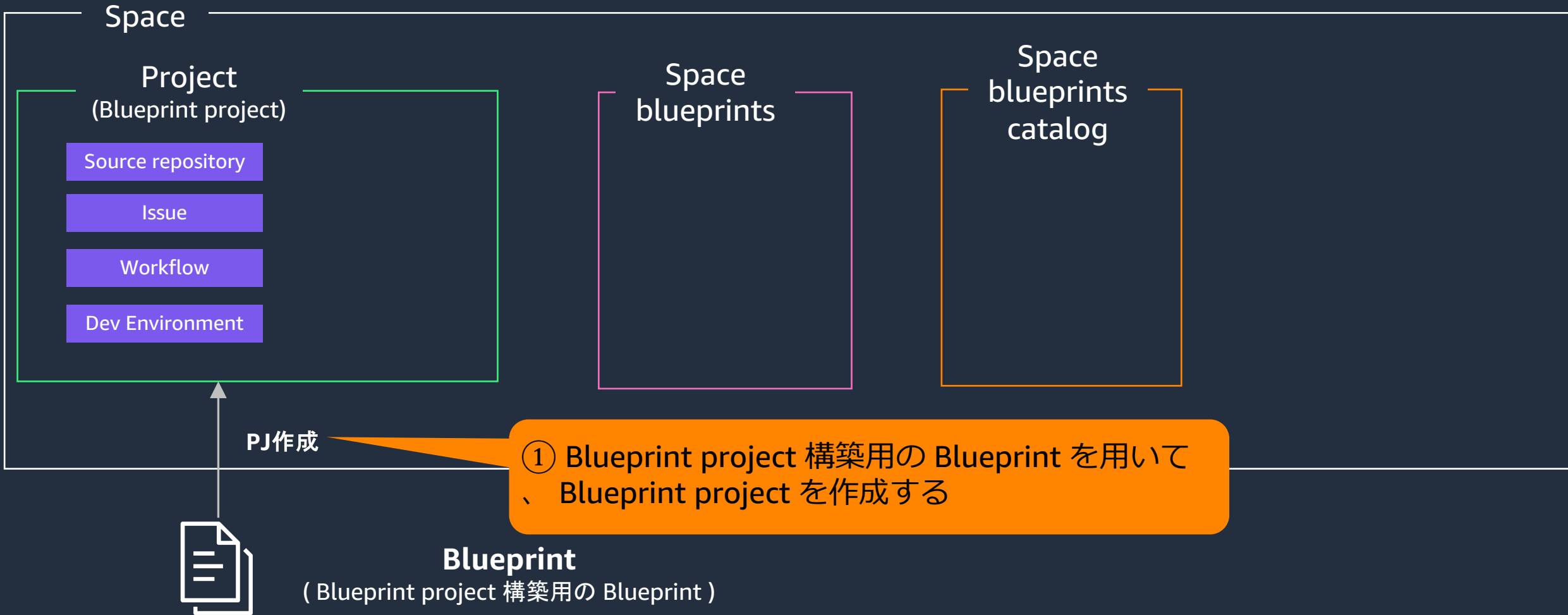
Custom blueprint 関連用語説明 (2/2)

- 合成 / 再合成
 - Project のソースコード、構成、リソースを表す project bundle を生成 / 再生成するプロセス
- project bundle
 - Project 内のソースコード、構成、リソースを表したもの
 - Blueprint を Project に用いる際は、この project bundle の内容がデプロイされる
- projen
 - Custom blueprint の合成、再合成、公開等を実現するために用いられているOSS
 - プロジェクトをコードで定義、管理することを実現
 - AWS CDK の作成者 Elad Ben-Israel らが中心に開発

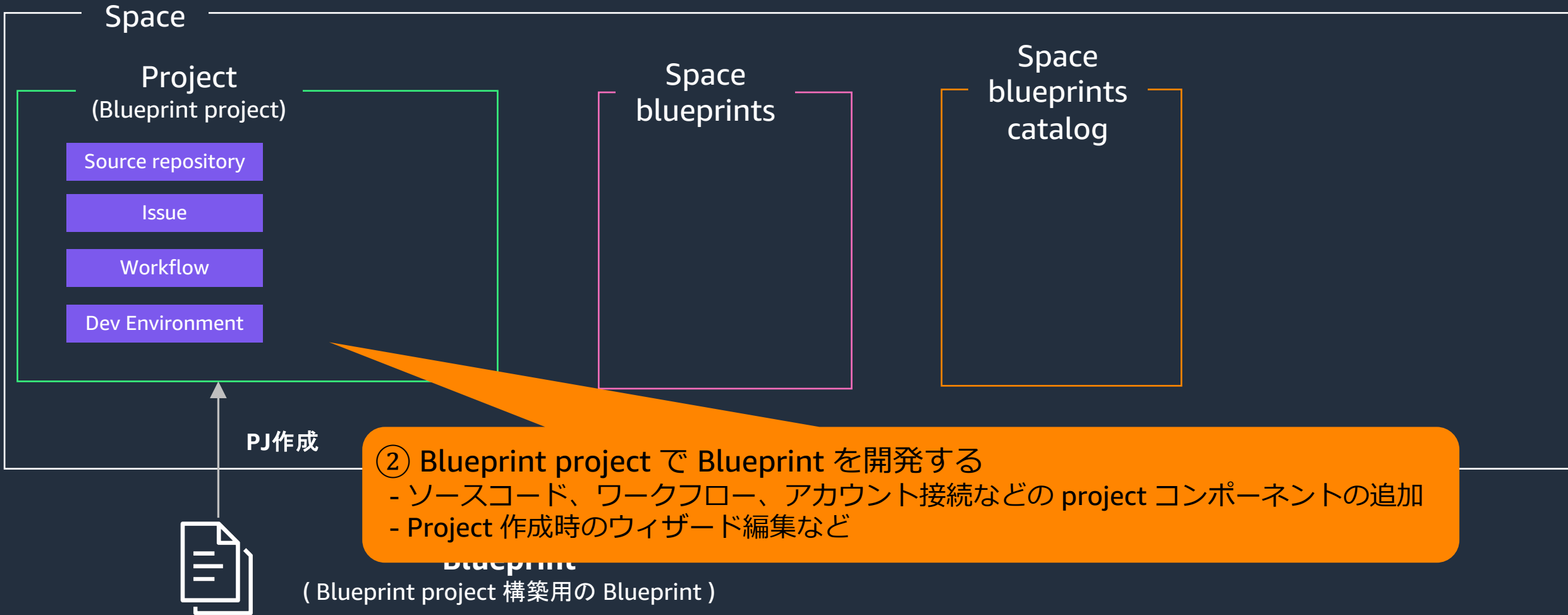
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/custom-bp-concepts.html>, <https://github.com/projen/projen>

Custom blueprint 作成/利用手順 概要

Custom blueprint 作成/利用手順 概要 (1/6)

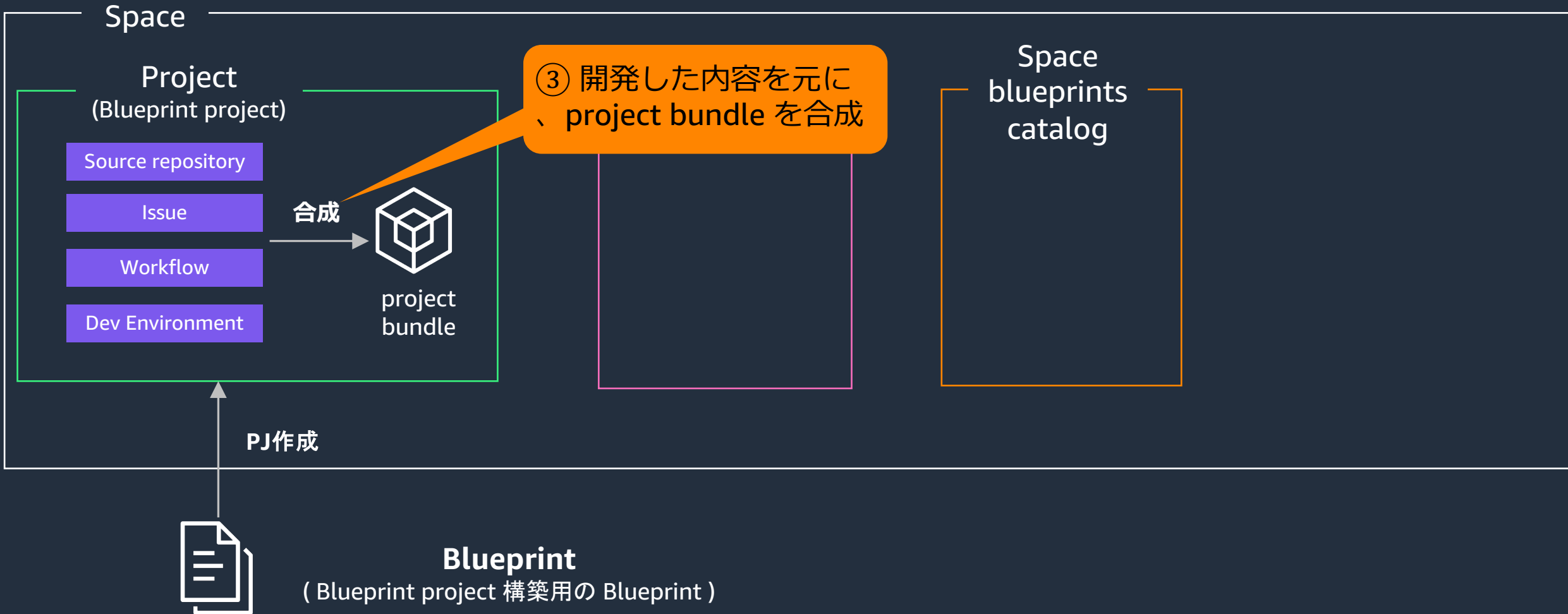


Custom blueprint 作成/利用手順 概要 (2/6)



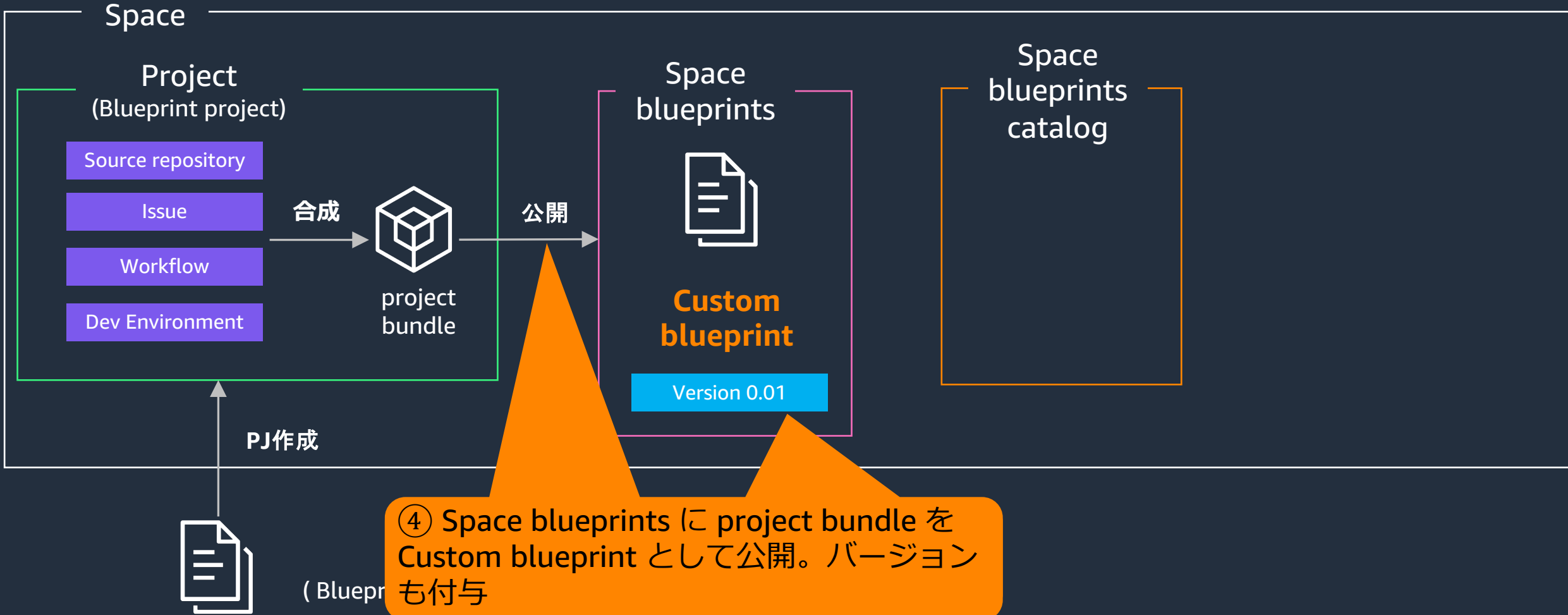
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/getting-started-bp.html>

Custom blueprint 作成/利用手順 概要 (3/6)

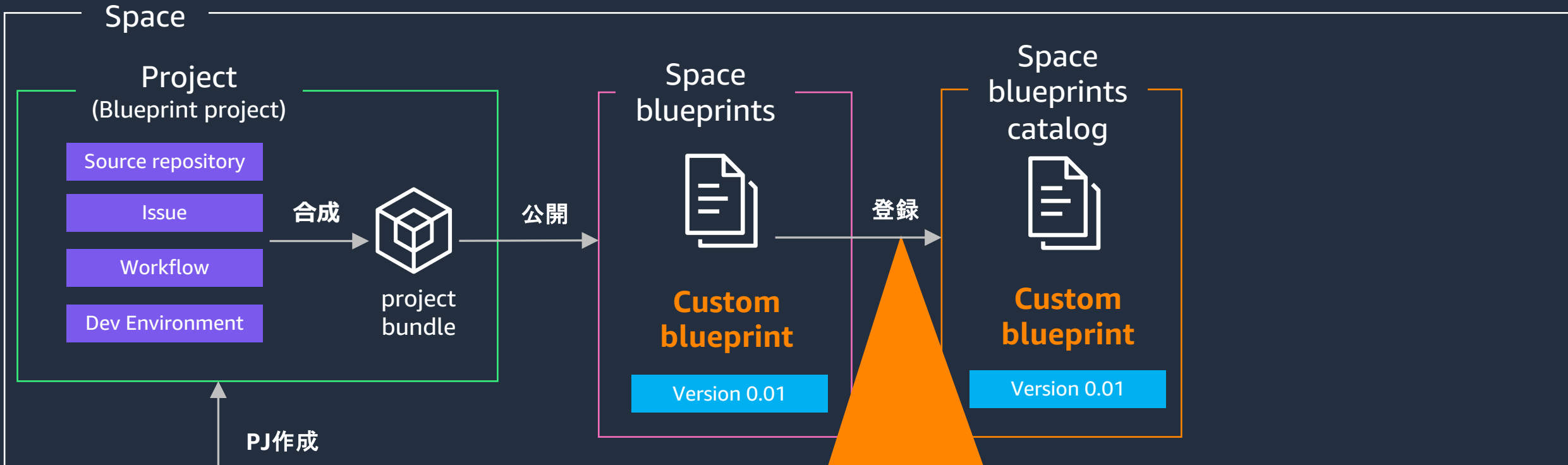


<https://docs.aws.amazon.com/codecatalyst/latest/userguide/getting-started-bp.html>

Custom blueprint 作成/利用手順 概要 (4/6)



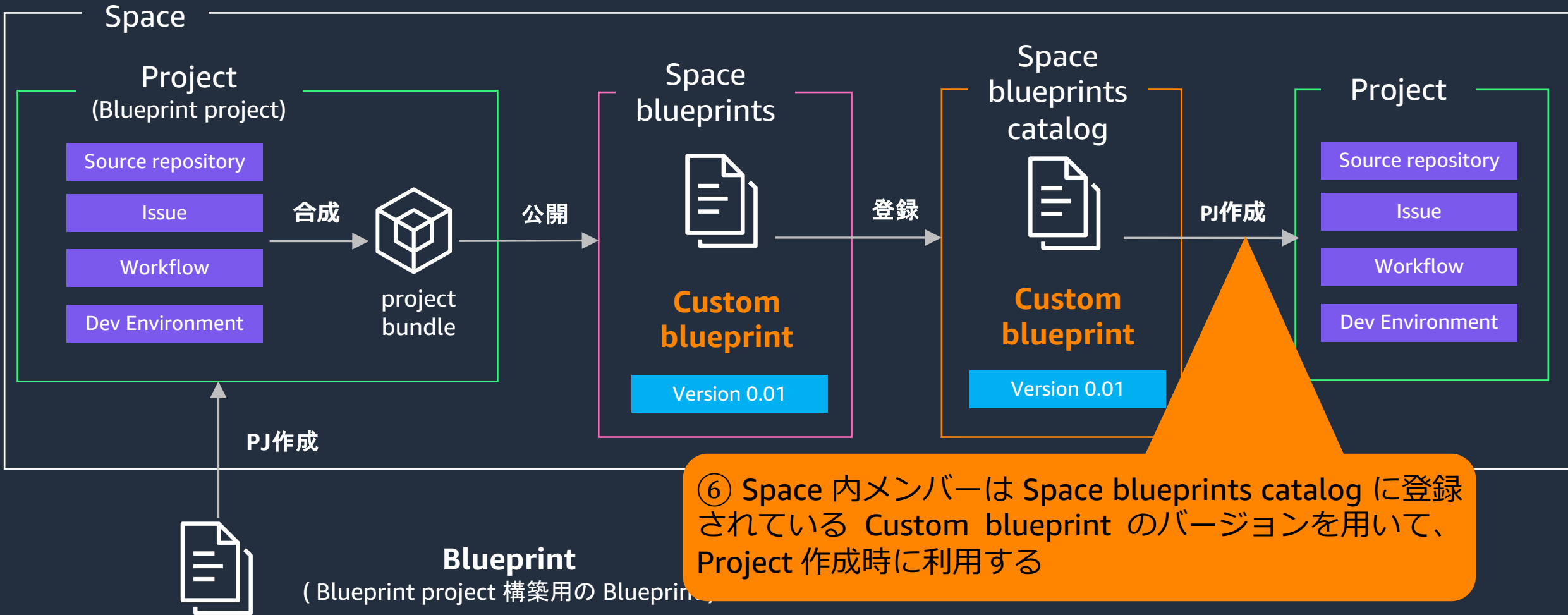
Custom blueprint 作成/利用手順 概要 (5/6)



⑤ Space blueprints にある Custom blueprint から、登録したい Custom blueprint のバージョンを選択して、Space blueprints catalog に登録

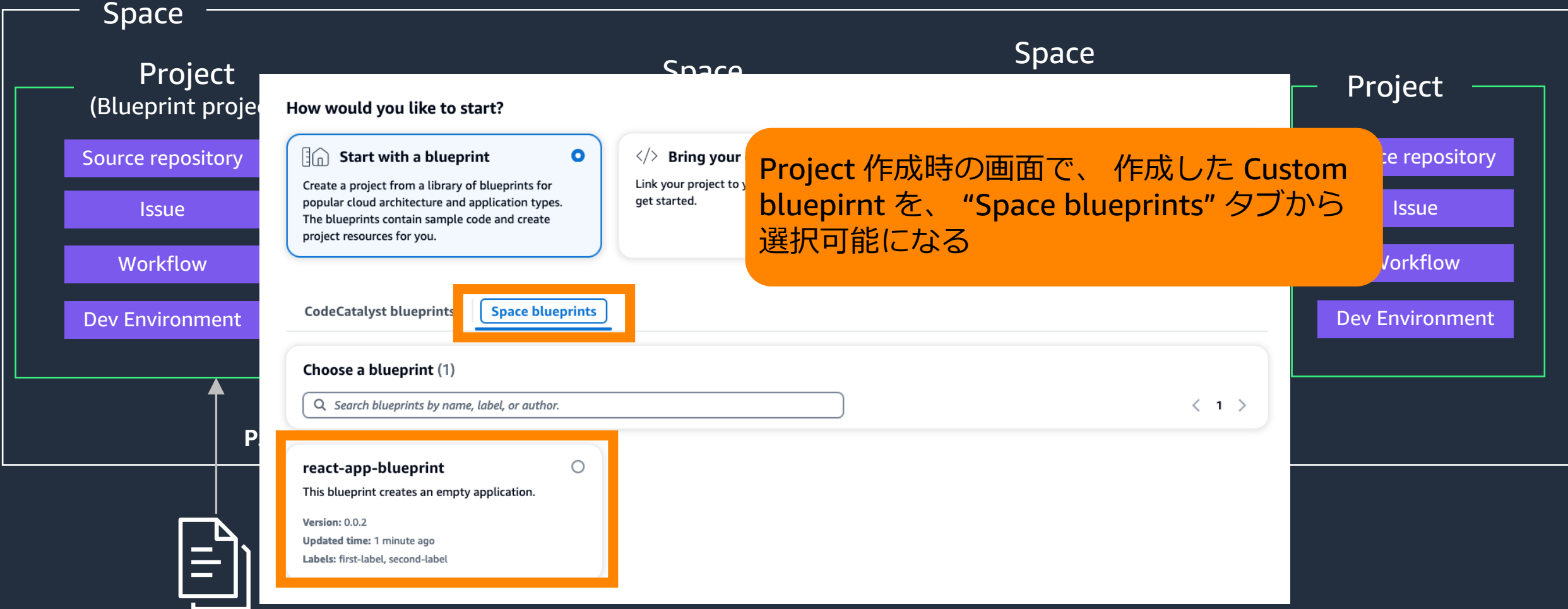
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/getting-started-bp.html>

Custom blueprint 作成/利用手順 概要 (6/6)



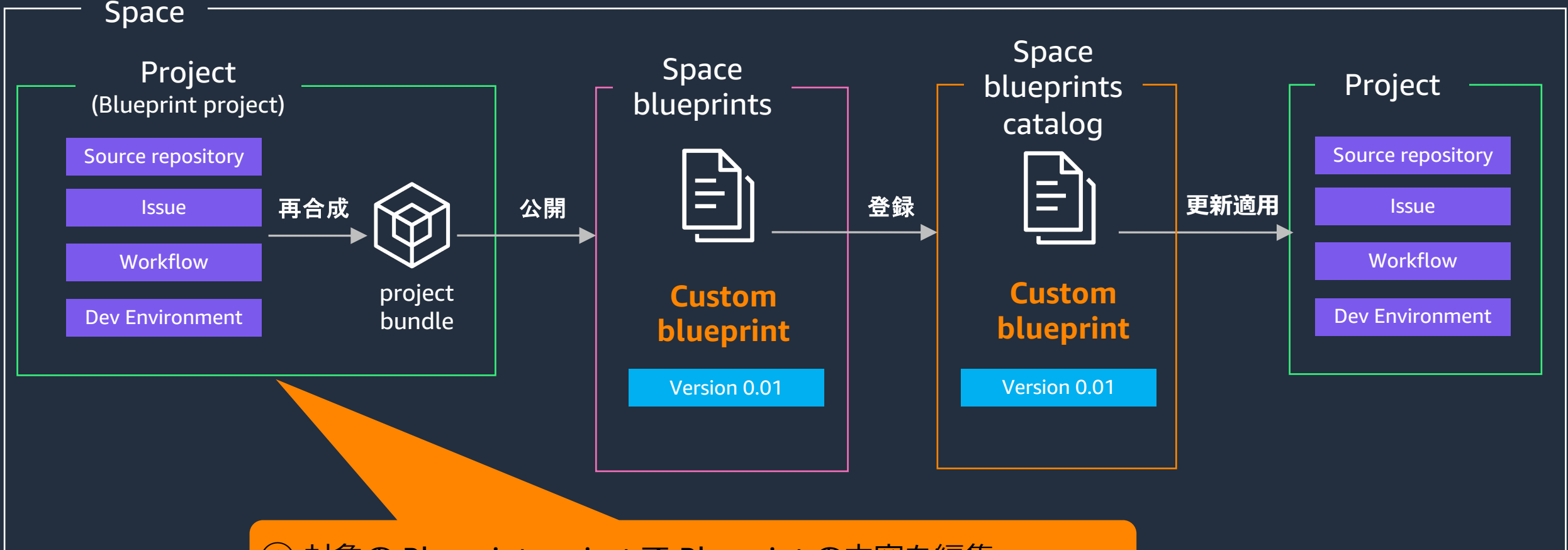
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/getting-started-bp.html>

Custom blueprint 作成/利用手順 概要 (補足)



Custom blueprint 更新/更新適用手順 概要

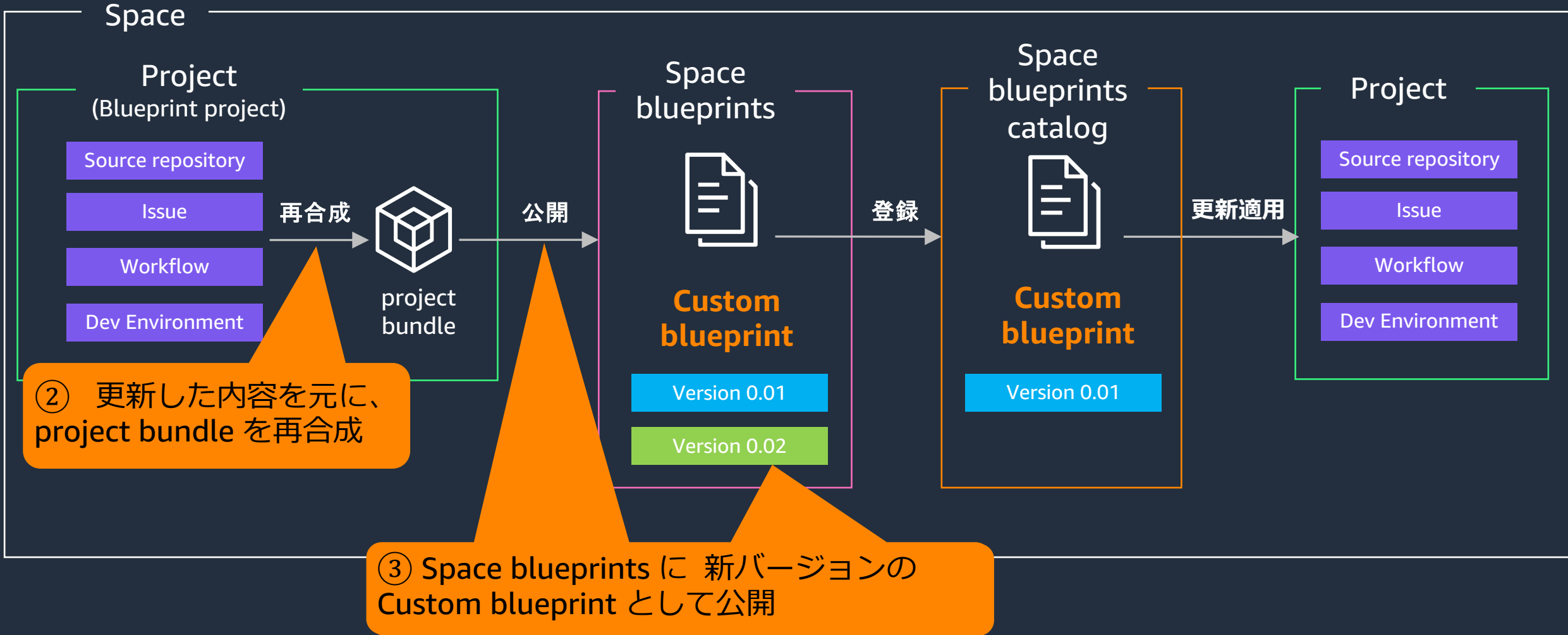
Custom blueprint 更新/更新適用の手順 概要 (1/5)



① 対象の Blueprint project で Blueprint の内容を編集

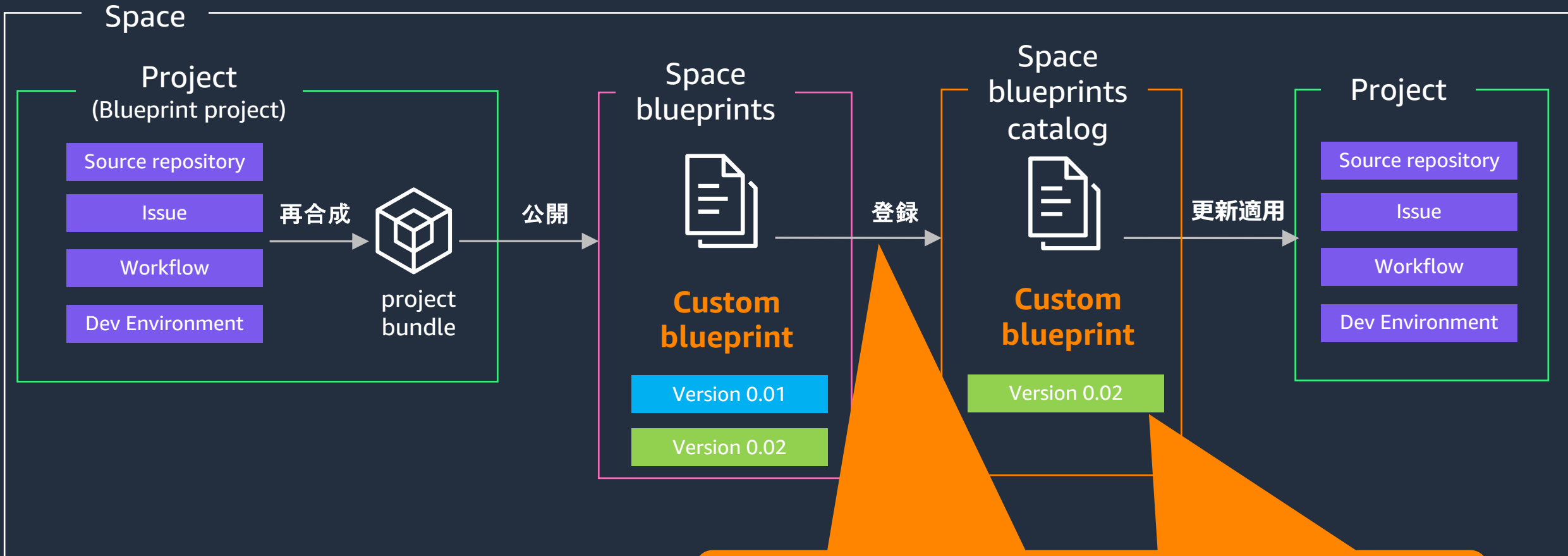
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/blueprint-getting-started-tutorial.html>

Custom blueprint 更新/更新適用の手順 概要 (2/5)



<https://docs.aws.amazon.com/codecatalyst/latest/userguide/blueprint-getting-started-tutorial.html>

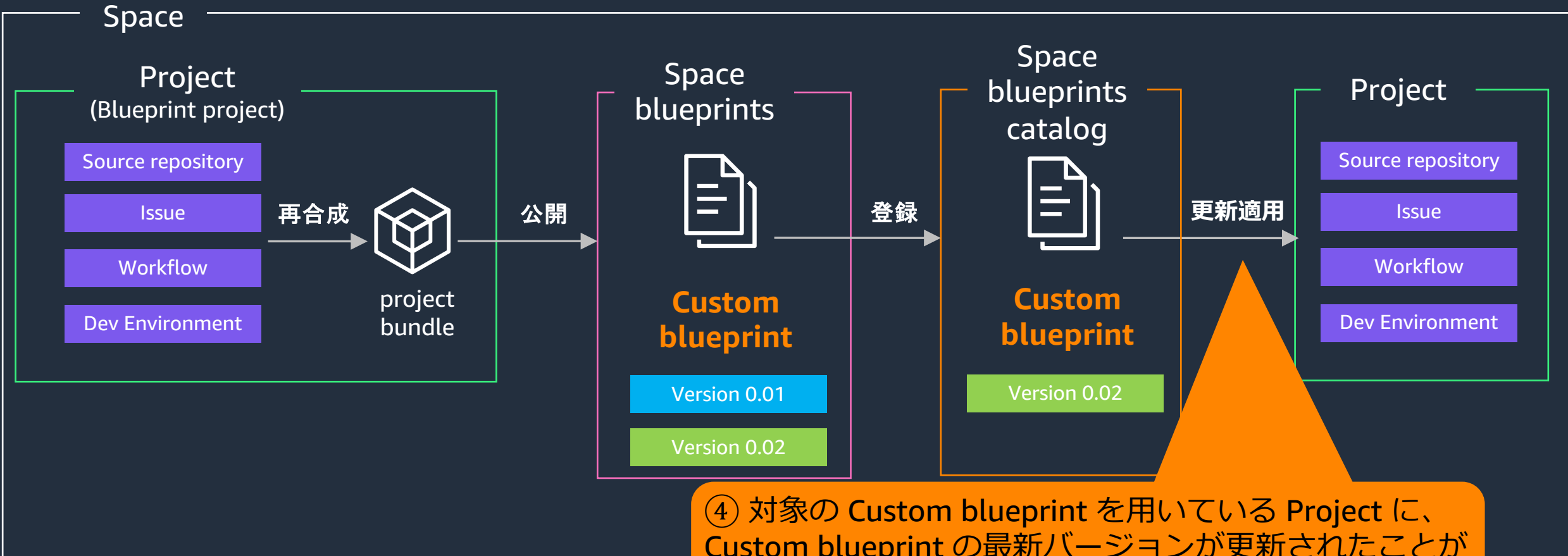
Custom blueprint 更新/更新適用の手順 概要 (3/5)



④ Space blueprints にある 新バージョンの Custom blueprint を Space blueprints catalog に登録する

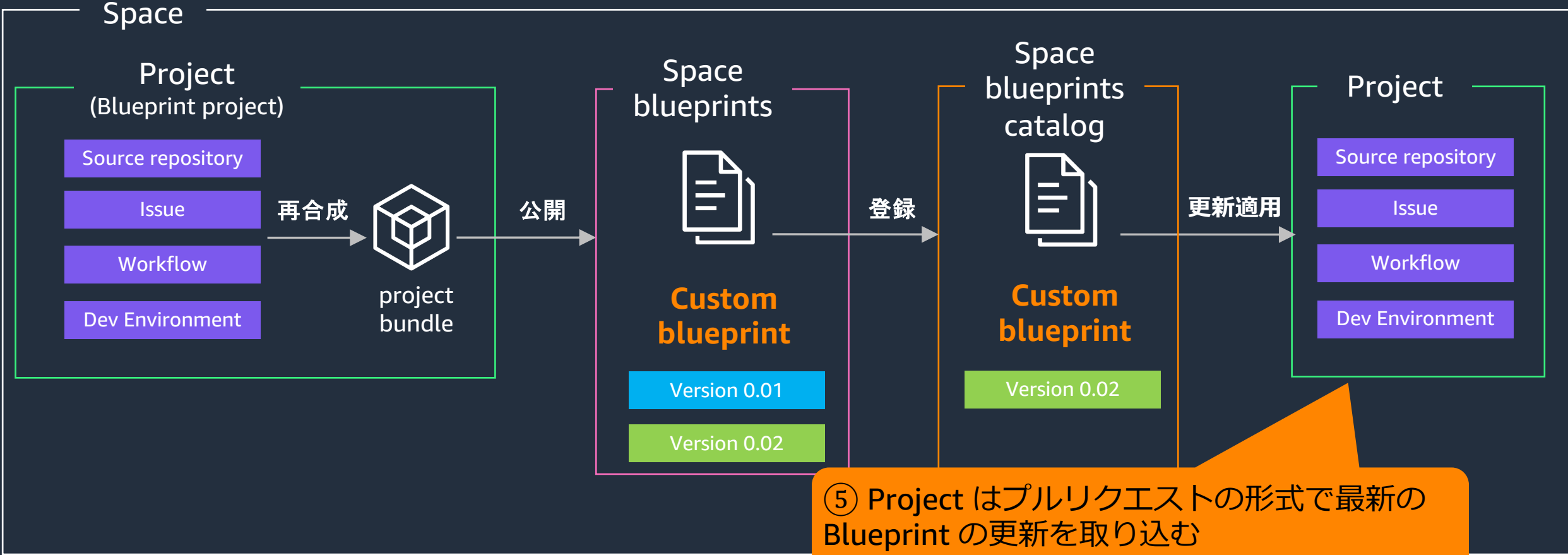
<https://docs.aws.amazon.com/codecatalyst/latest/userguide/blueprint-getting-started-tutorial.html>

Custom blueprint 更新/更新適用の手順 概要 (4/5)



④ 対象の Custom blueprint を用いている Project に、Custom blueprint の最新バージョンが更新されたことが通知される

Custom blueprint 更新/更新適用の手順 概要 (5/5)



<https://docs.aws.amazon.com/codecatalyst/latest/userguide/blueprint-getting-started-tutorial.html>

Custom blueprint 更新/更新適用の手順 概要 (補足)

The screenshot shows the AWS CodeCatalyst interface. On the left, the 'Blueprints' section is highlighted with an orange box. It displays a notification: 'Update react-app-blueprint to latest version 0.0.3' with an 'Update blueprint' button. Below this is a table of blueprints:

Blueprint name	Alias	Author	Status	Current version	Last updated
react-app-blueprint	-	CodeCatalystBB	Version 0.0.3 available	0.0.2	16 minutes ago

On the right, a 'Project' section is outlined in green, listing 'Source repository', 'Issue', 'Workflow', and 'Dev Environment'. An arrow labeled '更新適用' (Apply update) points from the blueprint update notification to the project. Below this, a pull request interface is shown for 'chore(resynthesis): update [amazon-codeca... PR1'. It includes a 'Mergeable' status, a diff view for 'hello-world.txt' showing a change from 'Hello, world!' to 'Hello, world!', and buttons for 'Create Dev Environment', 'Close', 'Approve', and 'Merge'.

Custom blueprint を利用している全ての Project に対し通知が送られる

Blueprint のアップデートを受け入れると、Pull request が作成される。この Pull request を取り込むことで、更新適用が完了する

<https://docs.aws.amazon.com/codecatalyst/latest/userguide/blueprint-getting-started-tutorial.html>

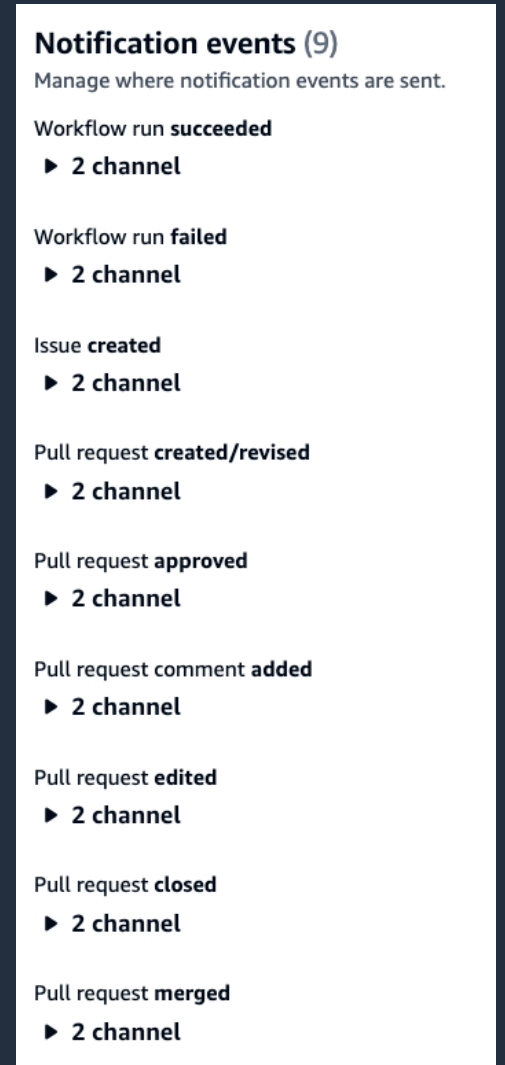
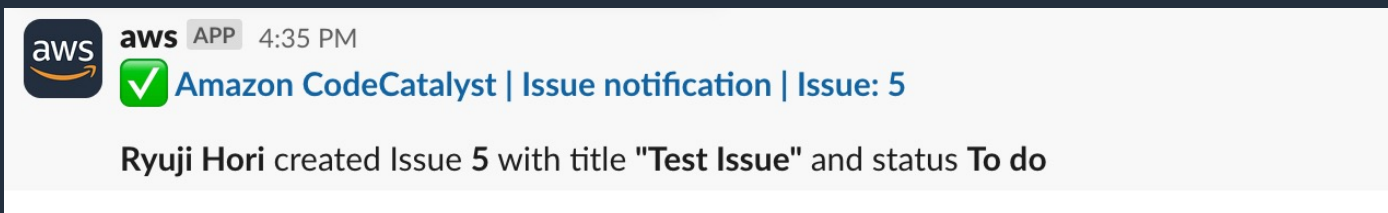


通知



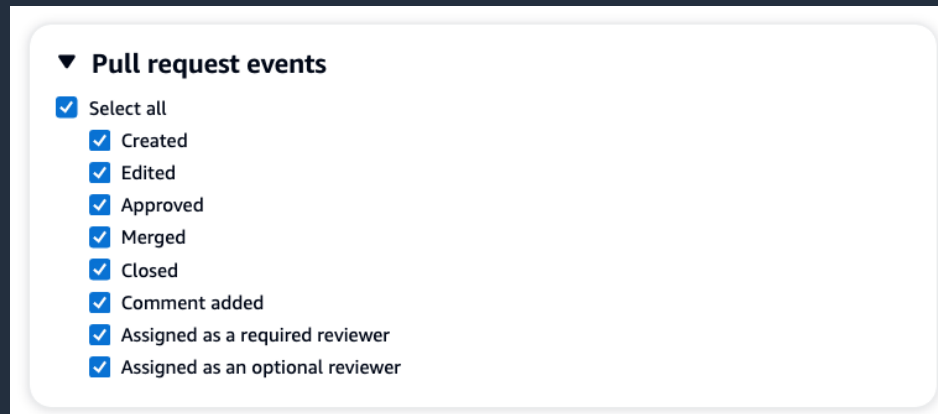
Slack Channel への通知

- 特定のグループで通知をシェアしたい場合に活用
- Project を監視するための通知を Slack Channel に対して設定可能
 - Pull request 作成通知、 Workflow 実行成功の通知など
- 複数 Channel に同時通知可能
- Project の Project settings メニューから設定
 - Project Administrator Role または Space Administrator Role が必要



Email への通知

- 自分が受け取りたい通知を、個人単位で設定したい場合に活用
- Space 内の各 Project の自分への通知を、個人単位で設定可能
- Slack Channel 向けの通知と、通知できる内容が異なるので注意
- Space の My setting メニューから設定可能
 - Role に関わらず全ての User が設定可能




Project の削除

Project の削除

- Project のリソースへのアクセスがすべて削除される
- AWS アカウントへデプロイされたリソースは別途 AWS アカウントでの削除が必要
- 削除には Project Administrator Role または Space Administrator Role が必要

Delete Modern-horiry-app

 Deleting this project deletes all repositories, issues, and artifacts in the projects for all members. To avoid incurring charges to your AWS account, delete any AWS resources and resource stacks deployed by the workflows in this project.

Delete project

<https://docs.aws.amazon.com/codecatalyst/latest/userguide/projects-delete.html>

クォータ

クォータ

Project のクォータ

項目	上限
Space あたりの Project 数	100
User が所属できる Project 数	1000
Project に所属できるメンバーの数	10000

Blueprint のクォータ

項目	上限
Space あたりの Blueprint 数	100

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



Thank you!