aws
SUMMIT
ONLINE
JAPAN | MAY 11-12, 2021

AWS-30

AWS サービスで実現する 継続的インテグレーション / 継続的デリバリー (CI/CD) 入門

松本 雅博

アマゾン ウェブ サービス ジャパン株式会社
技術統括本部 西日本ソリューション部
ソリューションアーキテクト

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

自己紹介

松本 雅博（まつもと まさひろ）

技術統括本部 西日本ソリューション部

ソリューションアーキテクト

関西を中心に西日本のお客様をご支援

好きなサービス

- Code シリーズ
- AWS CloudFormation, AWS Cloud Development Kit(CDK)

本セッションの対象者とゴール

対象者

- 継続的インテグレーション / 継続的デリバリー・デプロイメント (CI/CD) を AWS サービスで実現する方法にご興味をお持ちの方

ゴール

- CI/CD に利用可能な AWS の各サービスについて理解し、活用していただく

本セッションで取り扱わない内容

- CI/CD の詳細
- 各サービスの詳細な利用方法

アジェンダ

- 継続的インテグレーション / 継続的デリバリー (CI/CD) が必要となる背景
- CI/CD に関連する AWS サービス
- CI/CD はじめの一歩
- まとめ

継続的インテグレーション /
継続的デリバリー (CI/CD) が必要となる背景

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

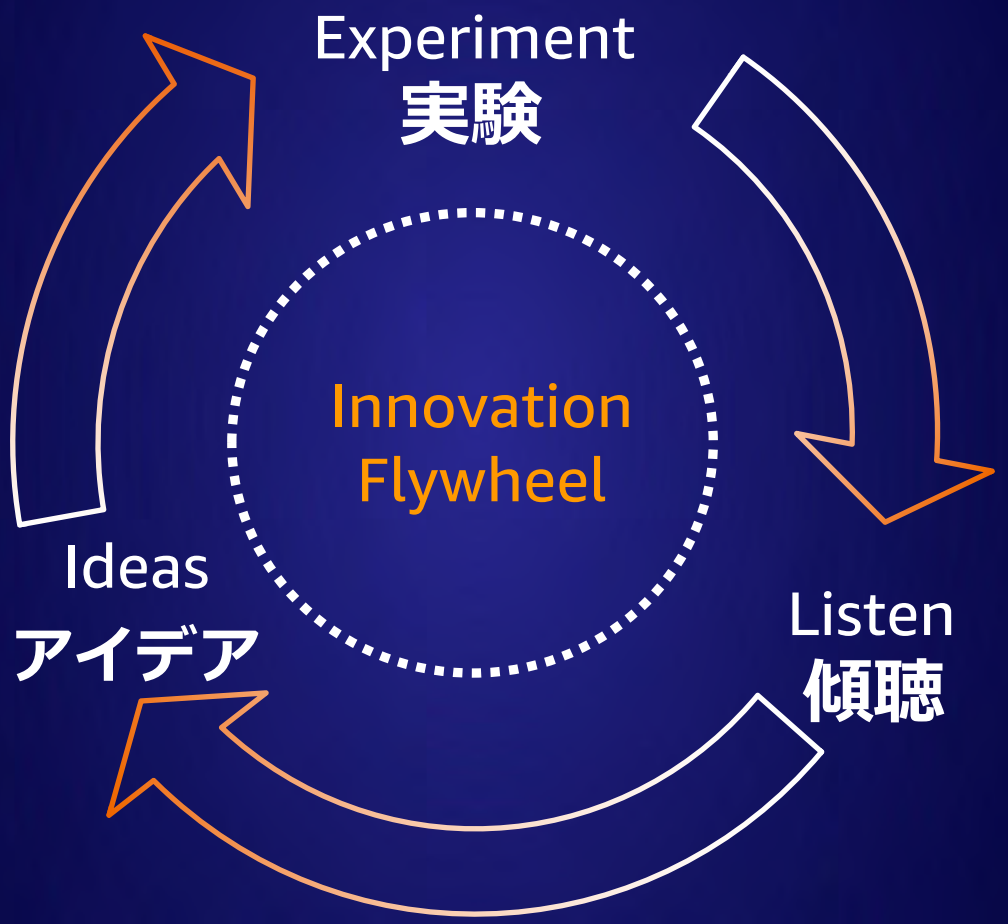
急速なイノベーションがビジネスを進化させる

新たなマーケット

新たな顧客価値

新たなデジタル製品とサービス

実験がイノベーションを加速する



ソフトウェア配布モデルは大きく様変わりしている

かつてのソフトウェア配布モデル

新しいソフトウェア配布モデル

OTA での配布が増加、Update をより高速に配信

ソフトウェア配布には機敏な動きが求められる

機敏な動きに必要なツールとは？

この新しいソフトウェア駆動の世界でソフトウェアを
リリースするのに必要なツールとは

- ソフトウェア開発のリリースプロセスの流れを管理するツール
- コードの不具合や潜在的な問題を正しくテスト／検査するツール
- アプリケーションをデプロイするツール

リリースプロセスに求められる 4 つのフェーズ

Source → Build → Test → Deploy

- ソースコードをチェックイン
- バージョン管理、ブランチ管理
- 新しいコードのピアレビュー

- Java、C#などのコードのコンパイル
- ユニットテスト
- スタイルチェッカー
- コードメトリック
- コンテナイメージの作成

- 他のシステムとの統合テスト
- 負荷テスト
- UIテスト
- 侵入テスト

- 本番環境にデプロイ

リリースプロセスのレベル

Source → Build → Test → Deploy

継続的インテグレーション (CI)

承認してからリリース

継続的デリバリー (CD)

継続的デプロイメント (CD)

CI/CD 導入による改善効果

30x
頻繁なデプロイ

440x
リードタイムの短縮

60x
ミスが減少

-21%
予期しない作業の削減

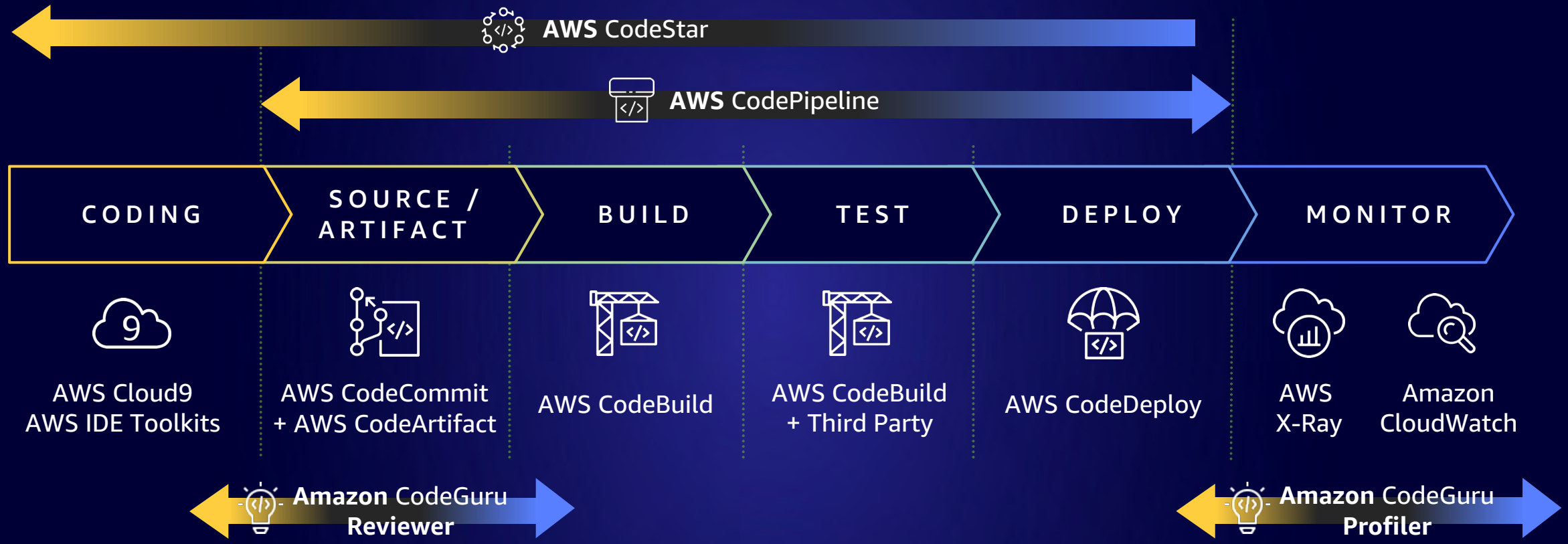
+44%
新しい作業

https://puppet.com/resources/whitepaper/2017-state-of-devops-report

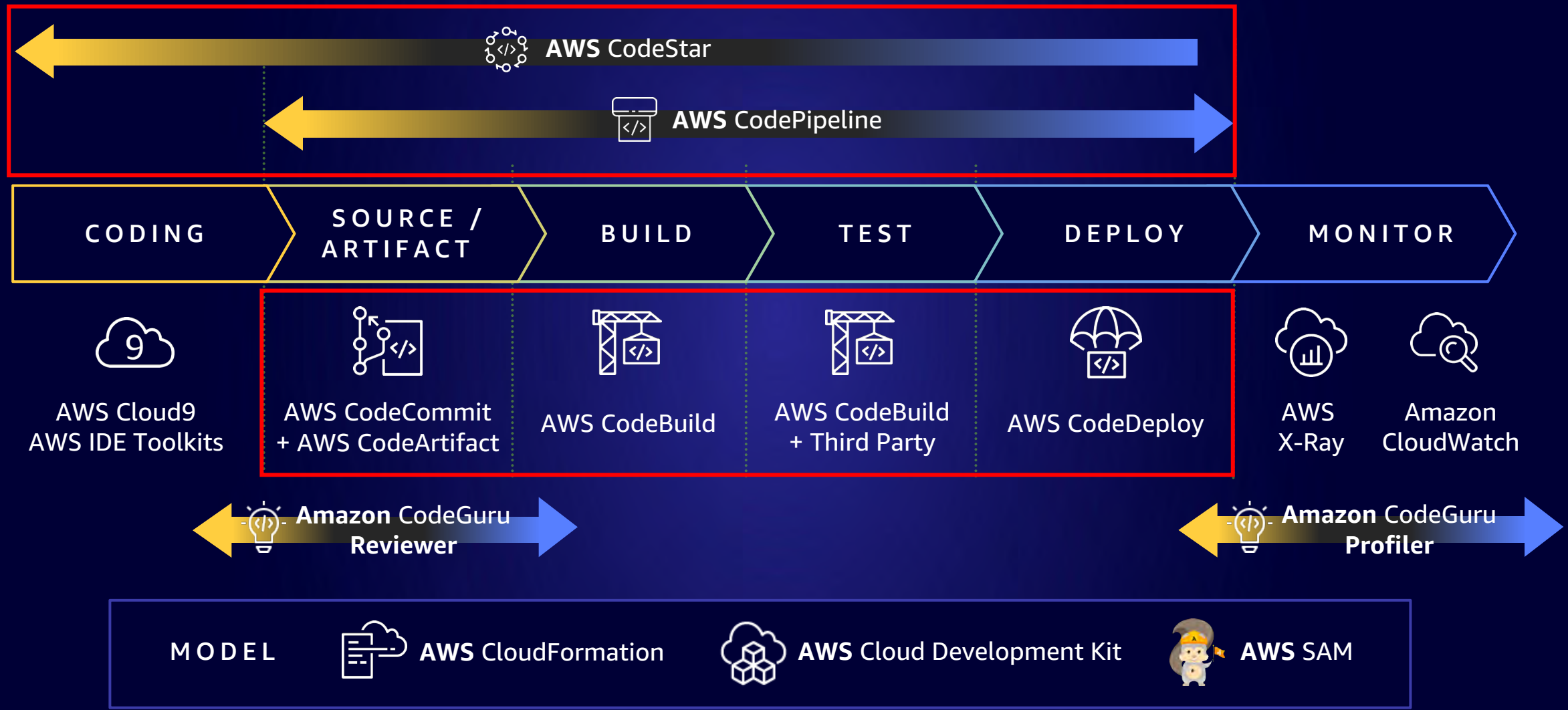
CI/CD に関連する AWSサービス

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス



CI/CD に関連する AWS サービス

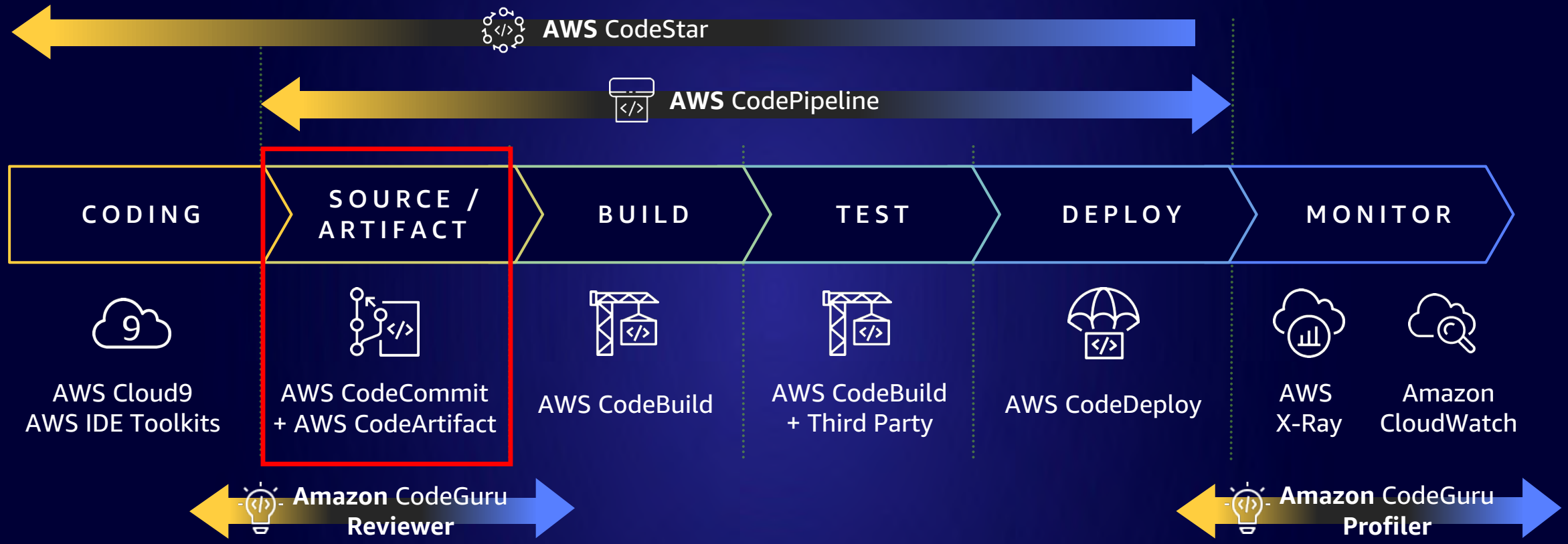


AWS CodeCommit

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス

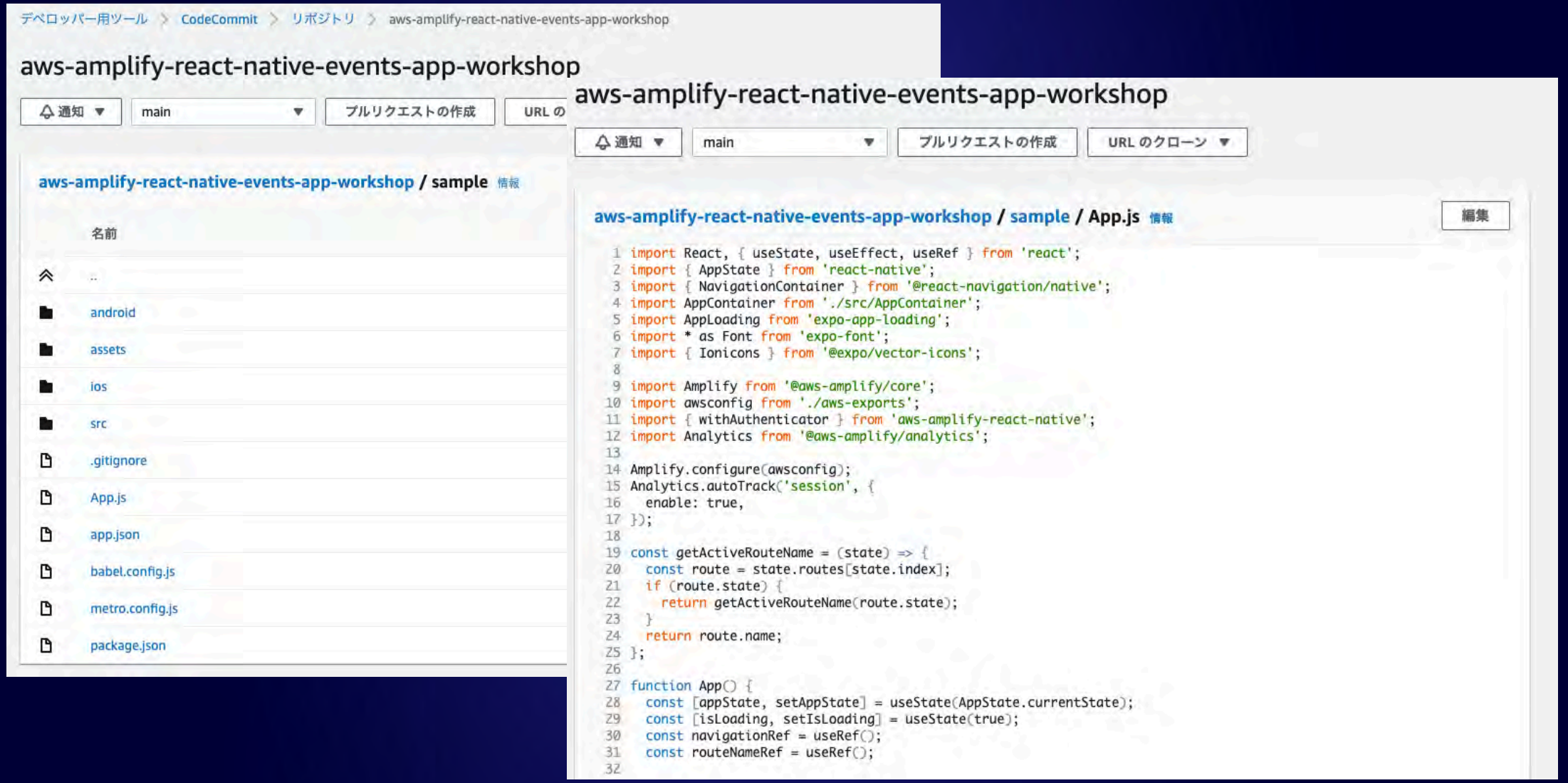
再掲



AWS CodeCommit

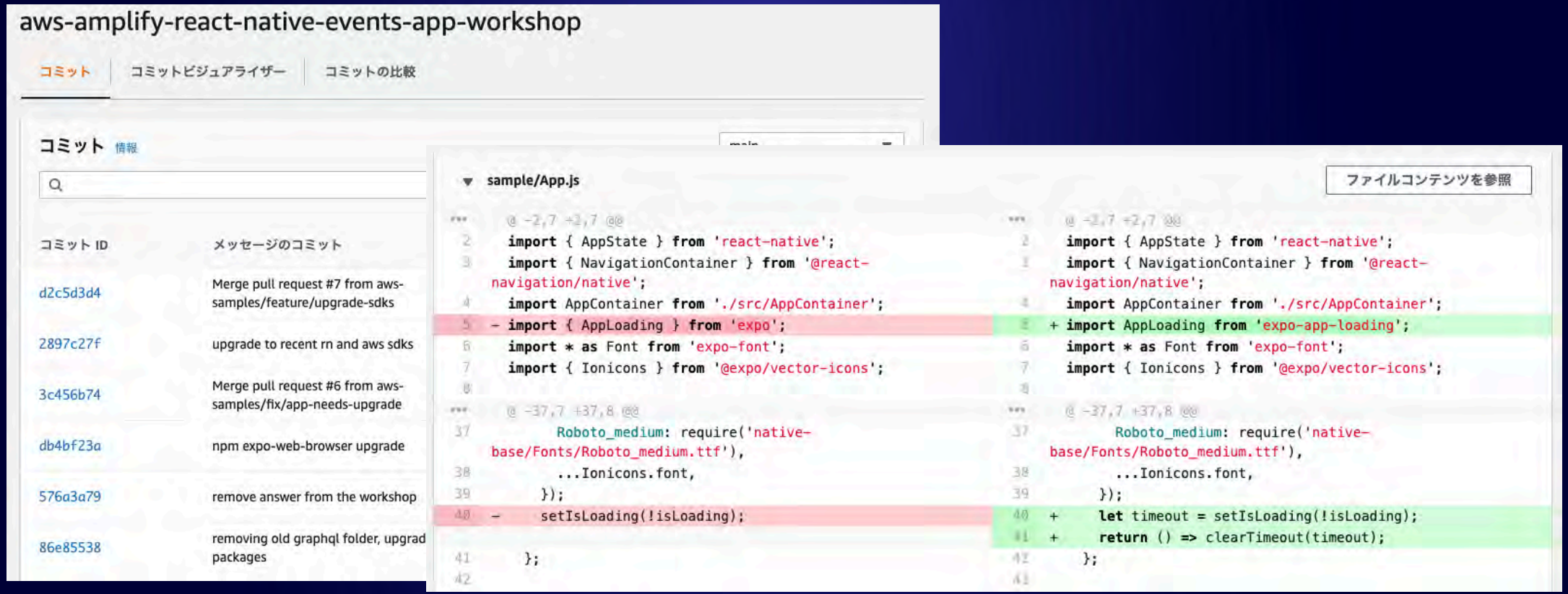
- セキュアでスケーラブルなマネージドGit互換ソース管理
- 標準的な Git ツールが利用可能
- Amazon S3 のスケーラビリティ、可用性、堅牢なストレージ
- リポジトリサイズの上限なし
- コンテンツの直接編集をサポート
- プルリクエスト機能に対応

リポジトリのコンテンツ参照



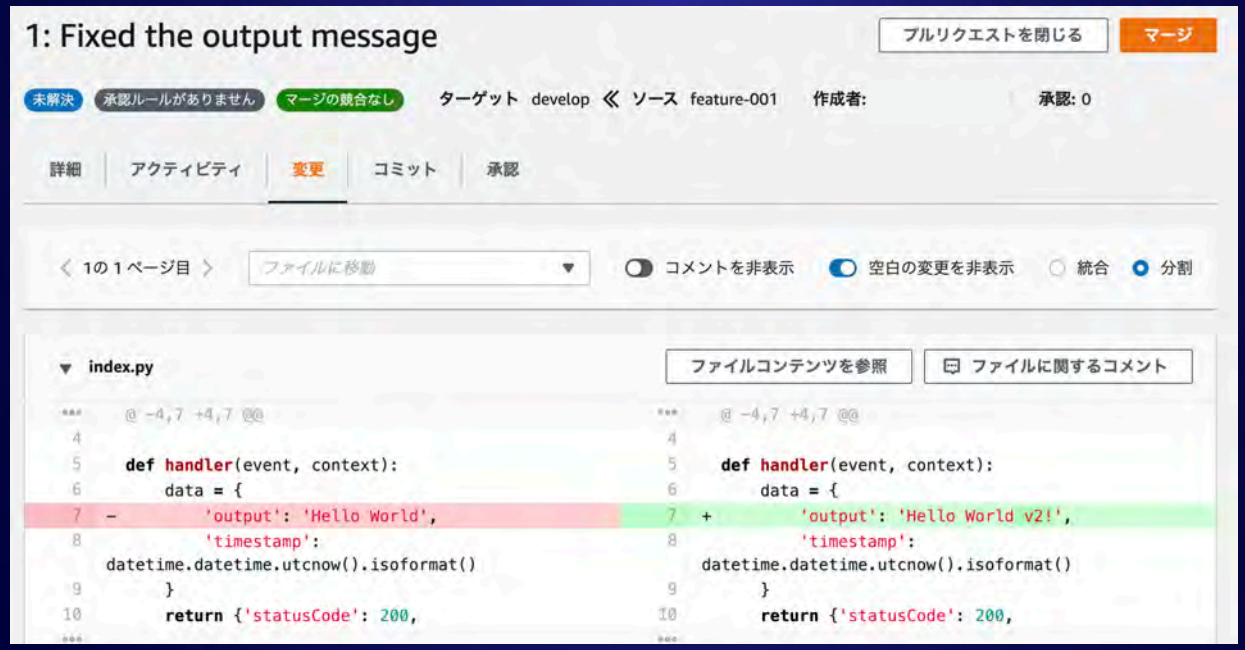
コミット履歴の表示

- ブランチを切り替えて表示可能
- 直前のコミットとの差分を比較できる



プルリクエストの作成

- リポジトリのユーザに対する作業内容の通知
- プルリクエストを起点として、議論やコミュニケーションの機会を提供
- コードレビューや変更箇所の精査ができ、品質改善に繋がる



（参考）AWS CodeArtifact

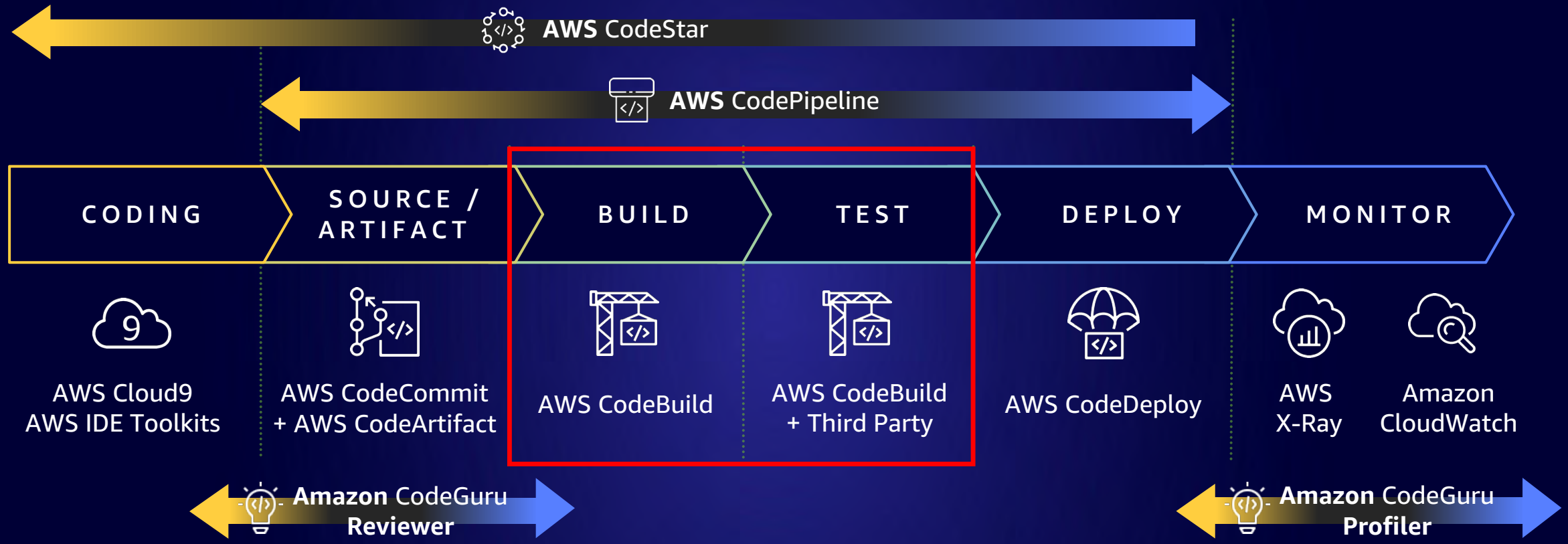
- セキュアでスケーラブルなマネージドアーティファクト管理
- Maven, Gradle, npm, yarn, twine, pip, Nuget(.Net) などの一般的なビルドツール、パッケージマネージャーから利用可能
- npm パブリックリポジトリ、 Maven Central, Python Pacage Index (PyPi), NuGet.org などからパッケージ取得可能
- AWS Key Management Service (KMS) による暗号化をサポート
- リポジトリはポリグロットでサポートされている任意のパケージを含めることができる

AWS CodeBuild

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス

再掲



AWS CodeBuild

- フルマネージドなビルドサービスでソースコードのコンパイル、テスト実行、ソフトウェアパッケージの作成を実行
- 継続的なスケールと同時複数ビルドプロセス
- 一貫したイミュータブルな環境のために個々のビルドを新規 Docker コンテナで実行
- ニーズに応じて Docker イメージを作成することによってカスタムなビルド環境を提供可能

AWS CodeBuild の仕組み

AWS CodeBuild

AWS Management Console

AWS Command Line Interface

AWS Tools and SDKs

AWS CodePipeline

AWS CodeBuild の仕組み

Amazon Simple Storage Service (S3)

ソースコード

AWS CodeBuild

作成

ビルドプロジェクト

ビルド環境

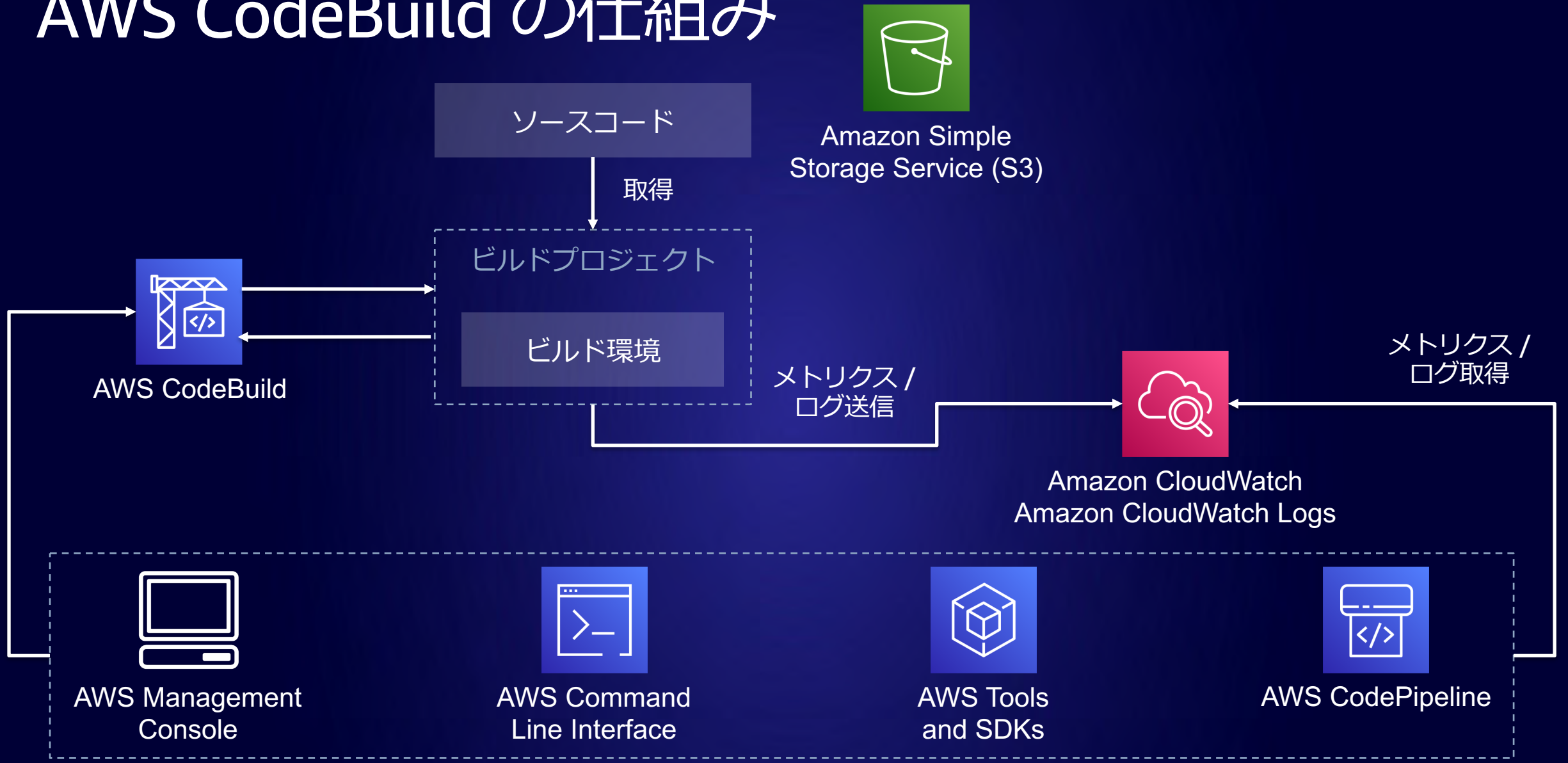
AWS Management Console

AWS Command Line Interface

AWS Tools and SDKs

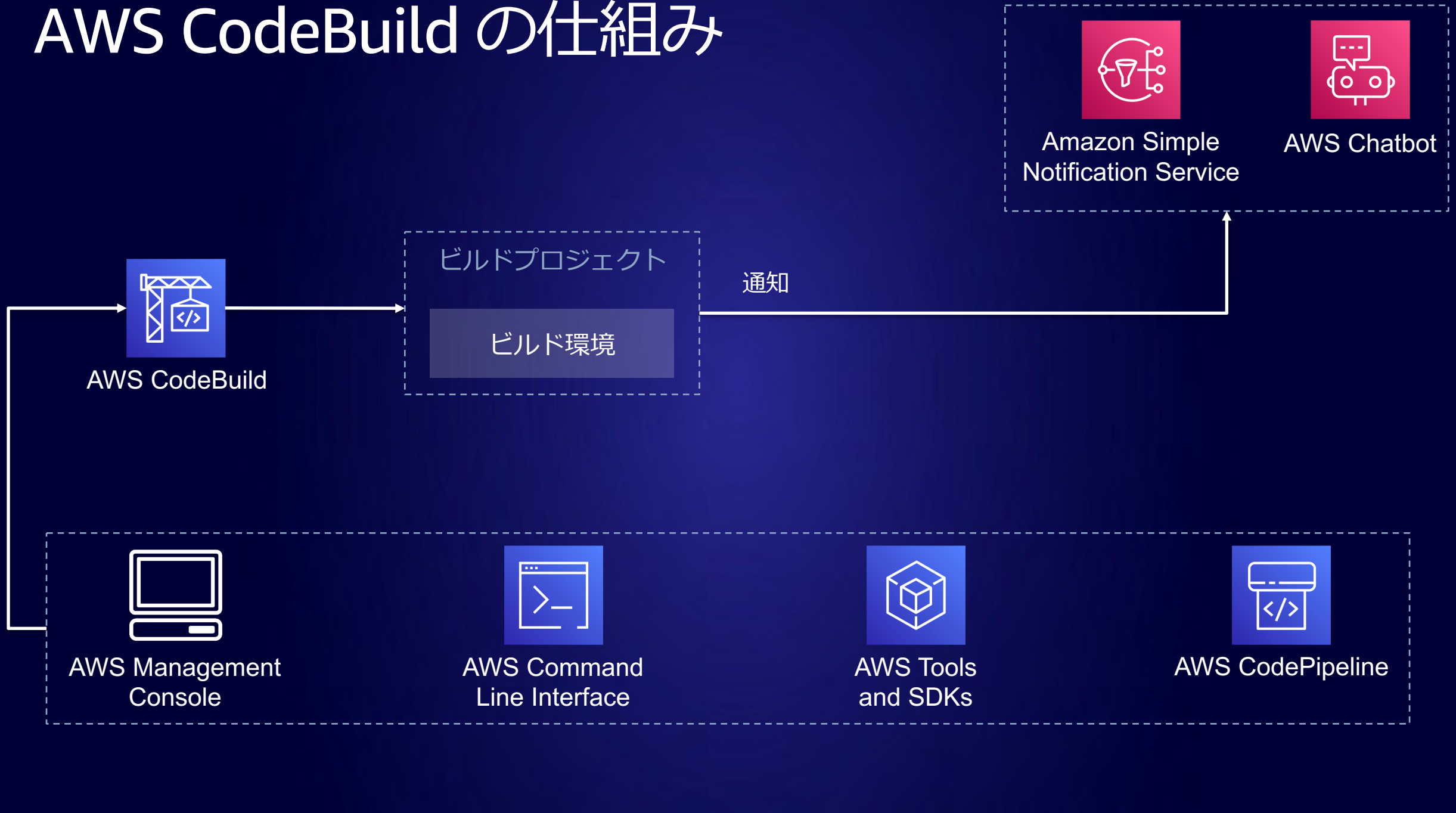
AWS CodePipeline

AWS CodeBuild の仕組み



AWS CodeBuild の仕組み

AWS CodeBuild の仕組み



ソースコードの場所

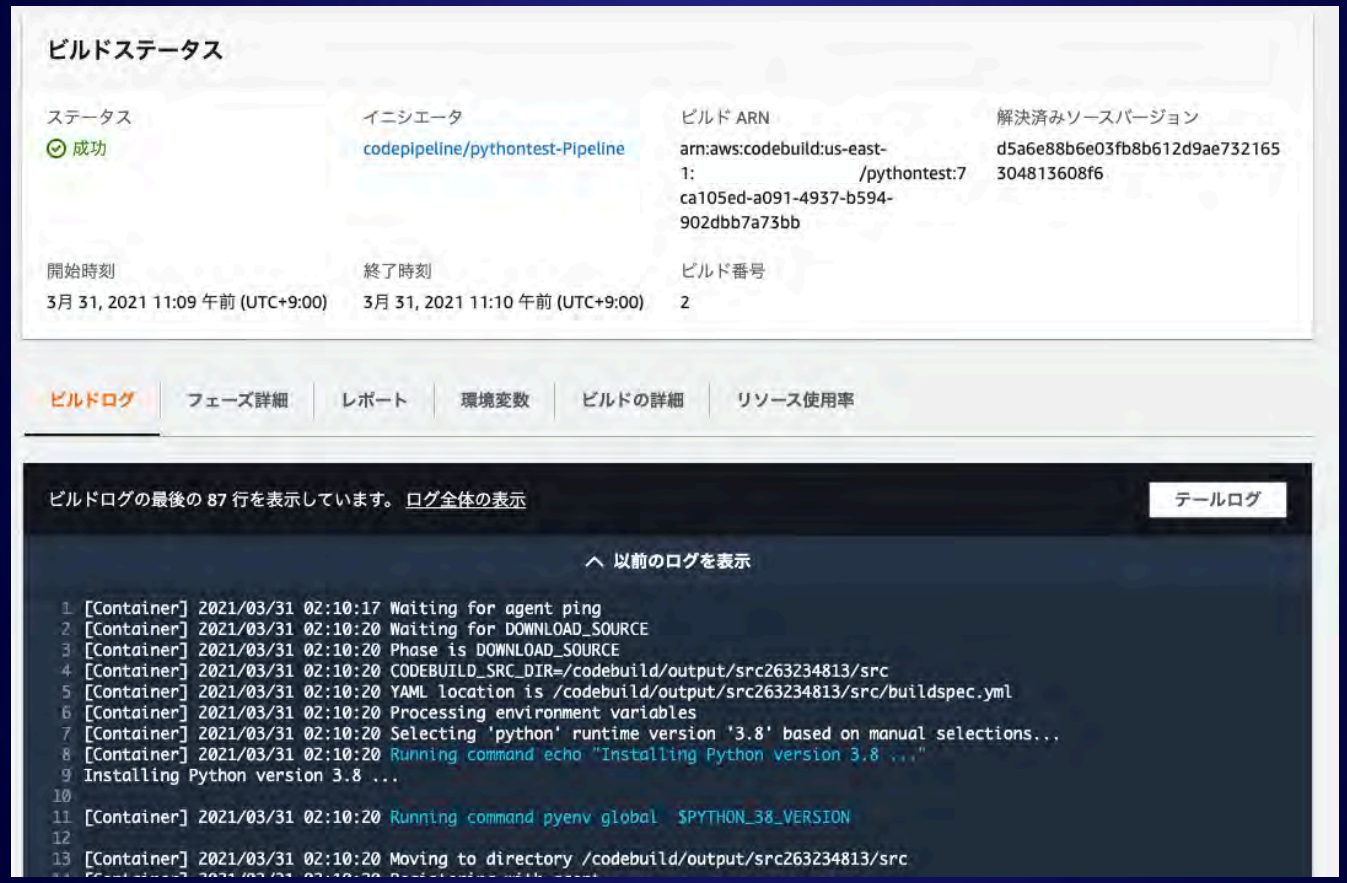
- **AWS CodeCommit**
  - リポジトリを指定する
- **Amazon S3**
  - バケットやフォルダの位置を指定する
- **GitHub**
  - OAuth 等で接続しリポジトリを指定する
- **Bitbucket**
  - OAuth 等で接続しリポジトリを指定する
- **GitHub Enterprise**
  - アクセストークンで接続しリポジトリを指定する

ビルドに使用できるランタイム

- Ubuntu と Amazon Linux 2 プラットフォーム
- Android : 28, 29
- .NET : 3.0, 3.1, 5.0
- Go : 1.12, 1.13, 1.14, 1.15
- Node.js : 8, 10, 12, 14
- Java :openjdk8, openjdk11, corretto8, corretto11
- PHP : 7.3, 7.4, 8.0
- Python : 3.7, 3.8, 3.9
- Ruby : 2.6, 2.7

ビルド履歴の確認

- コンテナ内のログをコンソールから確認可能

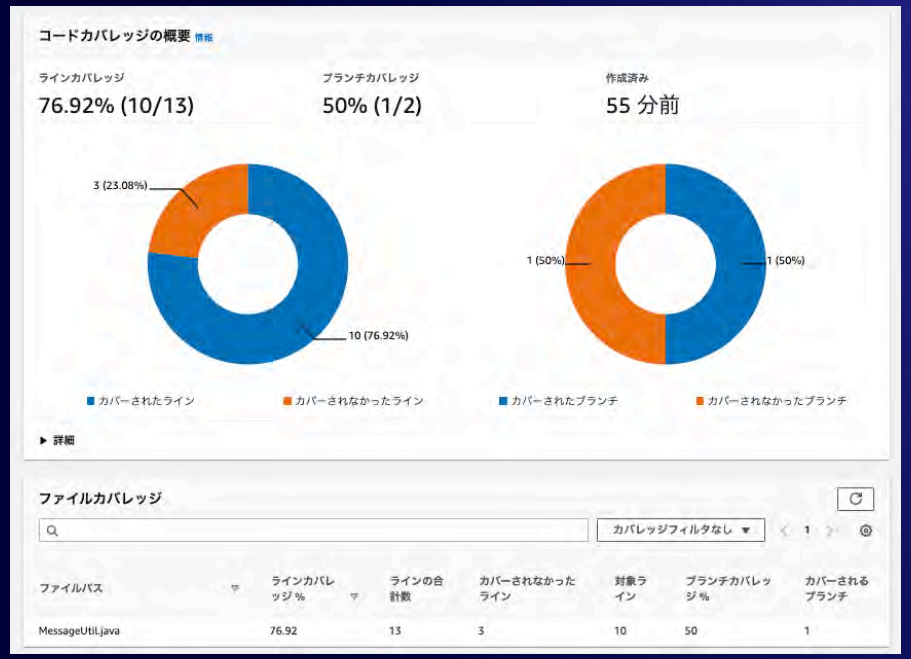


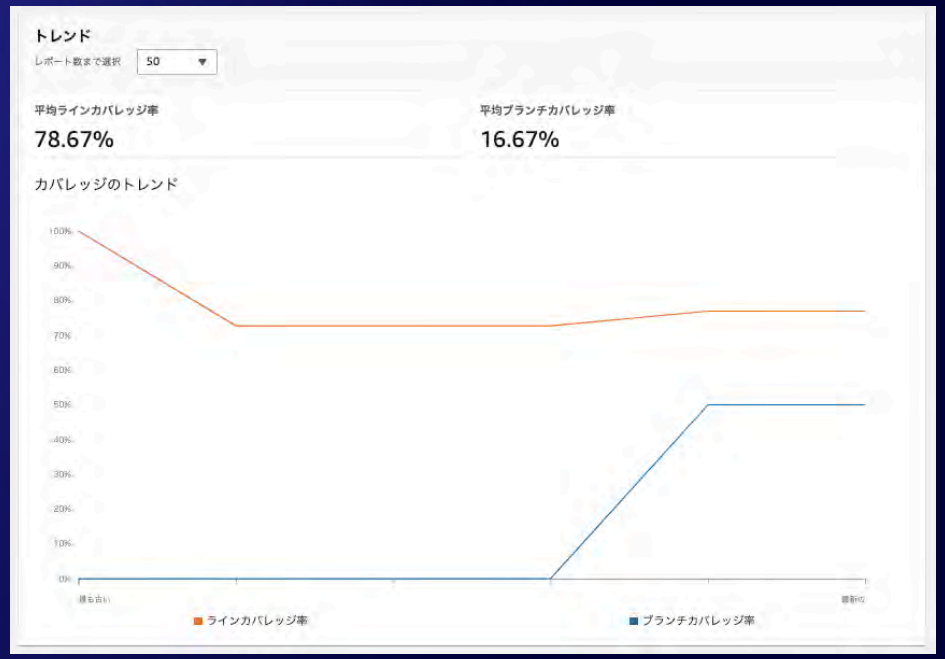
テストレポート

- ビルド時に実行したテストのレポートを作成できる
- 以下のファイル形式をサポート
  - Cucumber JSON, JUnitXML, NUnitXML, NUnit3XML, TestNGXML, Visual Studio TRX

コードカバレッジレポート

- ビルド時に実行したテストのカバレッジレポートを作成できる
- 以下のファイル形式をサポート
  - JaCoCo XML, SimpleCov JSON, Clover XML, Cobertura XML





ステータスの通知

- 通知ルールを設定してビルド状態の通知が可能

Amazon Simple Notification Service

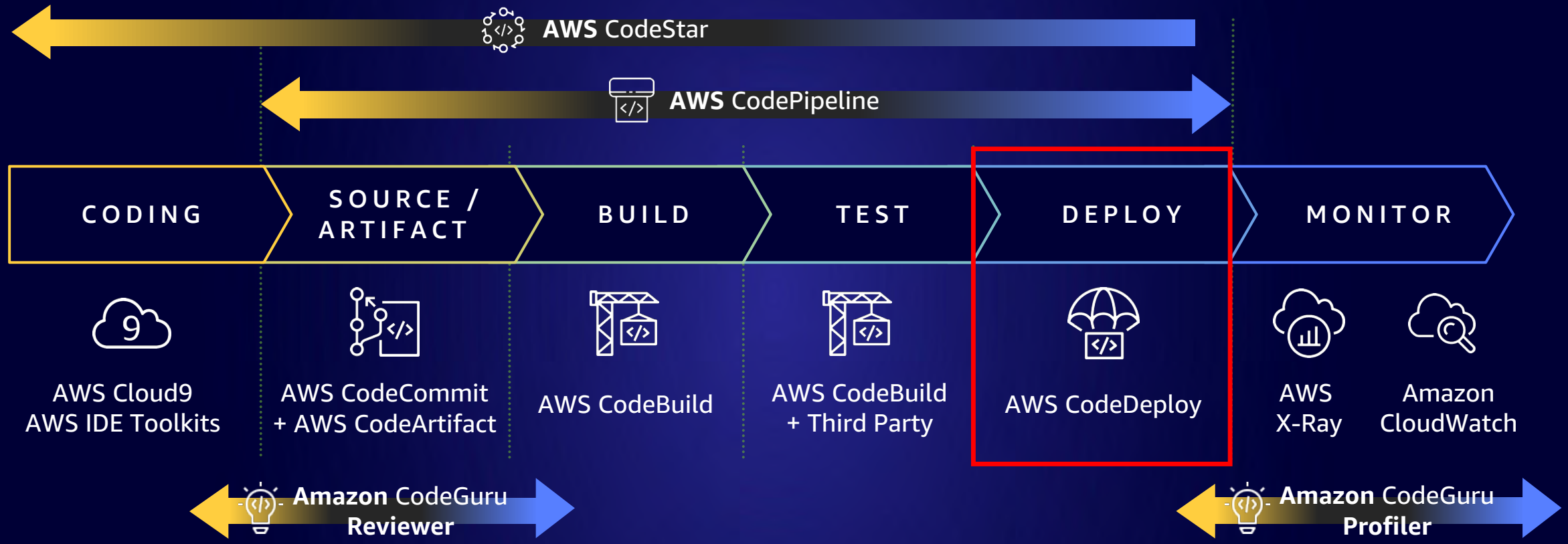
AWS Chatbot

AWS CodeDeploy

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス

再掲

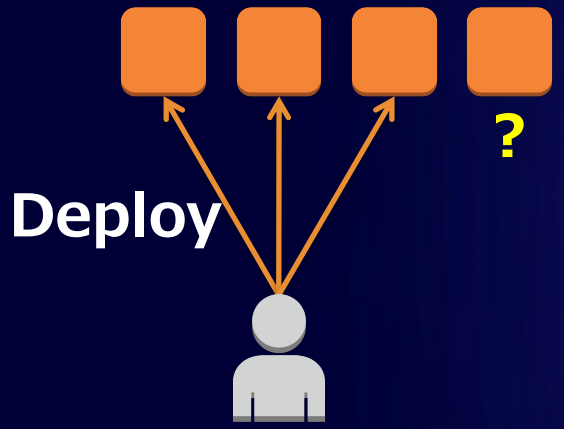


AWS CodeDeploy

- ソフトウェアのデプロイを自動化する、フルマネージド型のサービス
- Amazon EC2、AWS Lambda、コンテナ、オンプレミス サーバーへの自動的なデプロイメント
- アプリケーションの複雑なアップデートの実施
- アプリケーションのデプロイ中のダウンタイムを回避
- エラーを検知すると自動的にロールバックを実行

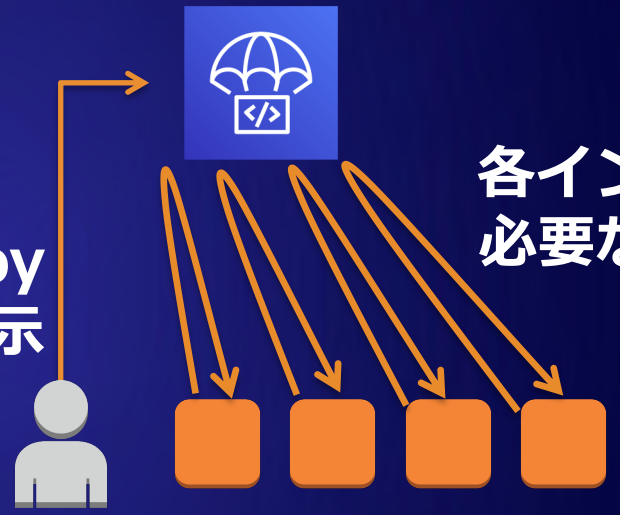
Pull 型のデプロイ手段

× Push型



Deploy先サーバを
知っている必要がある

○ Pull型



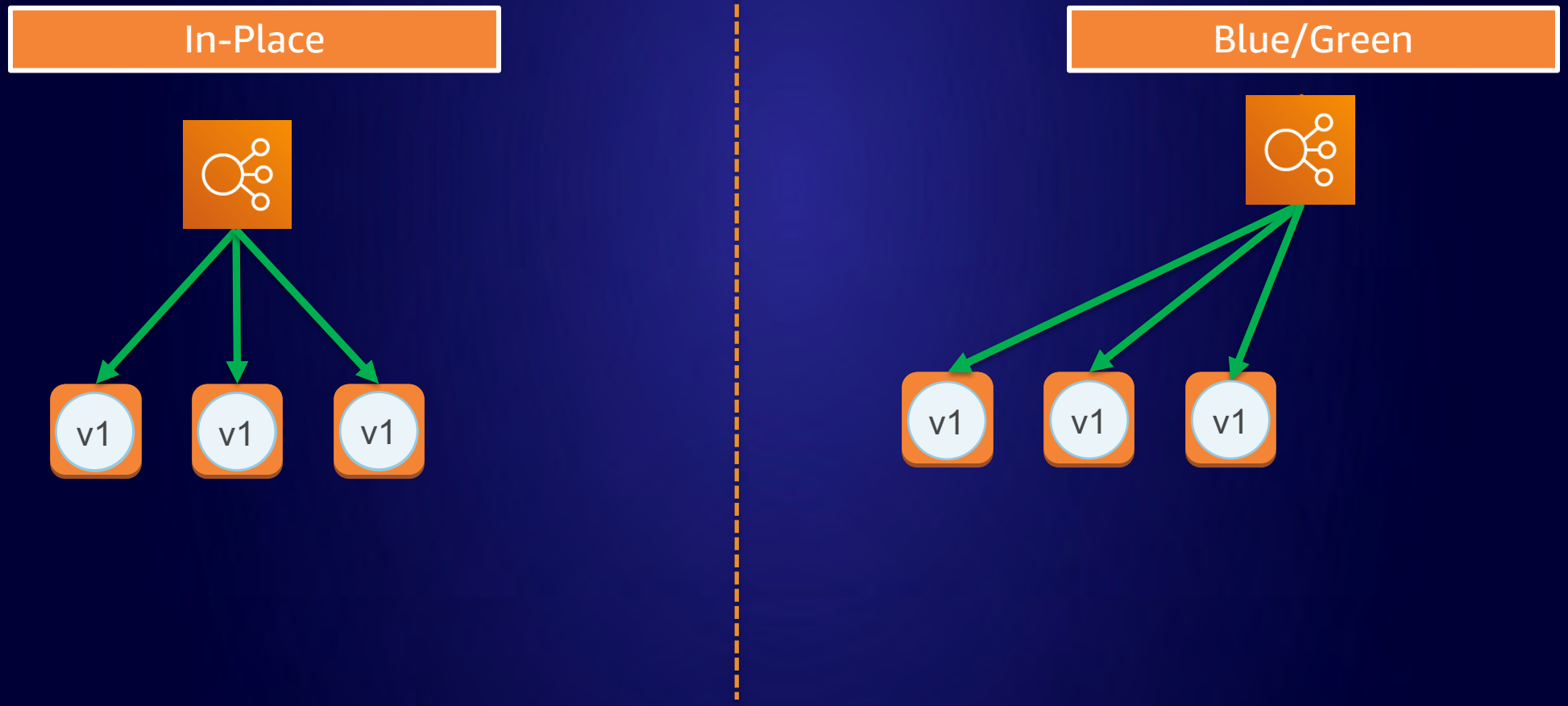
各インスタンスが
必要な変更をPull

どんなサーバが立っているか意識する必要はない

自動化を実現するには、Pull 型のデプロイが推奨される。

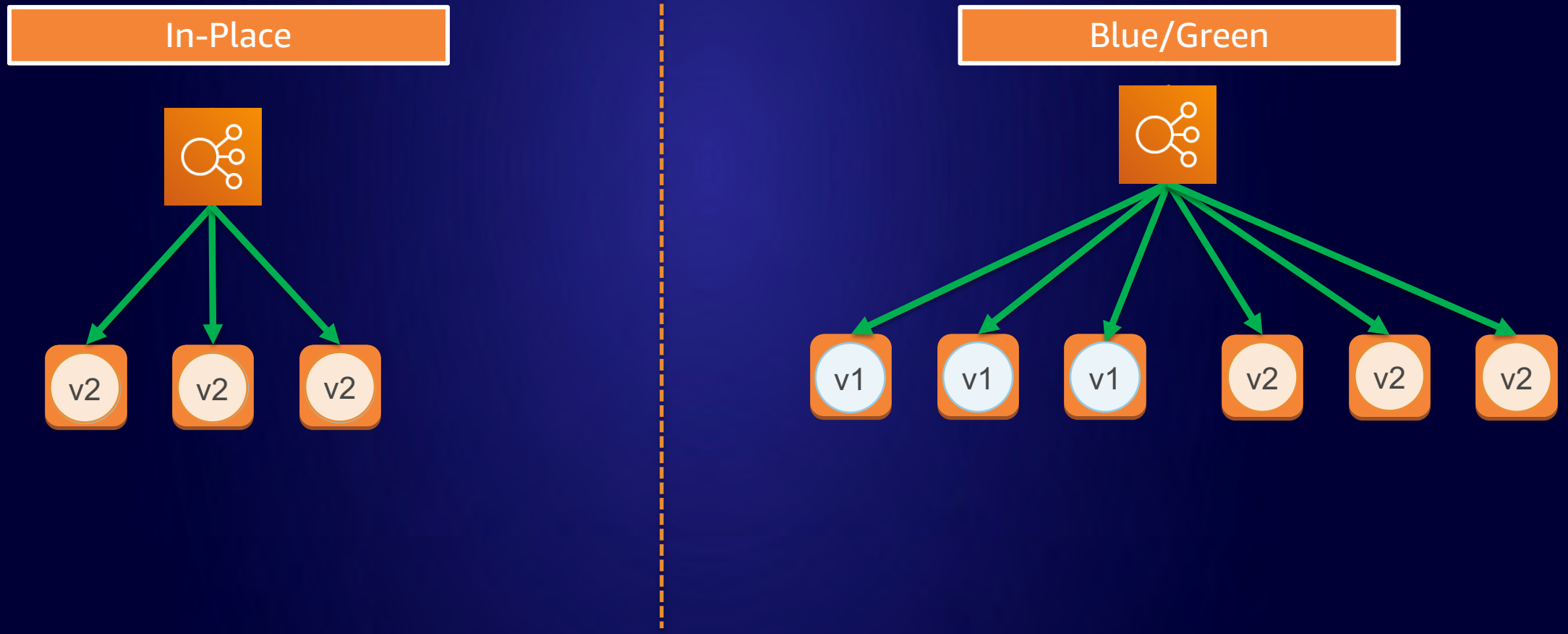
In-Place デプロイと Blue/Green デプロイ

v1 がデプロイされた状態



In-Place デプロイと Blue/Green デプロイ

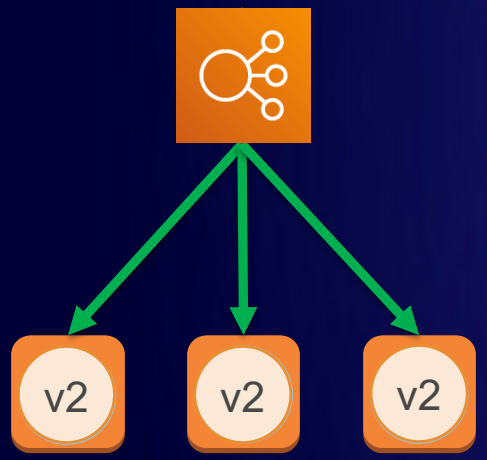
v2 をデプロイ



In-Place デプロイと Blue/Green デプロイ

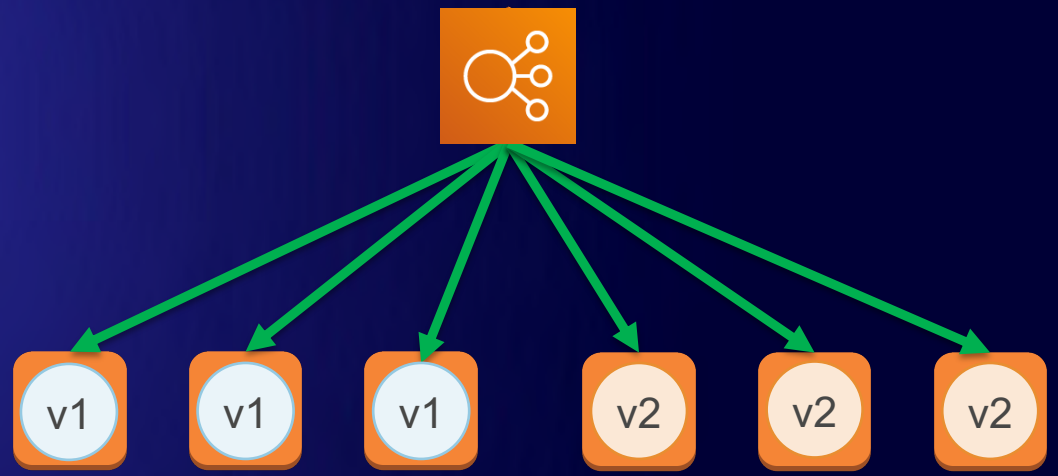
EC2 の場合、In-Place もしくは Blue/Green
Lambda, ECS の場合は Blue/Green

In-Place



既存ノードのアセットを更新
(1 台ずつ行う場合は
ローリングデプロイとも言う)

Blue/Green



新規にノードを構築し、デプロイ・テスト後
リクエストの振り分け先を変更

EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0

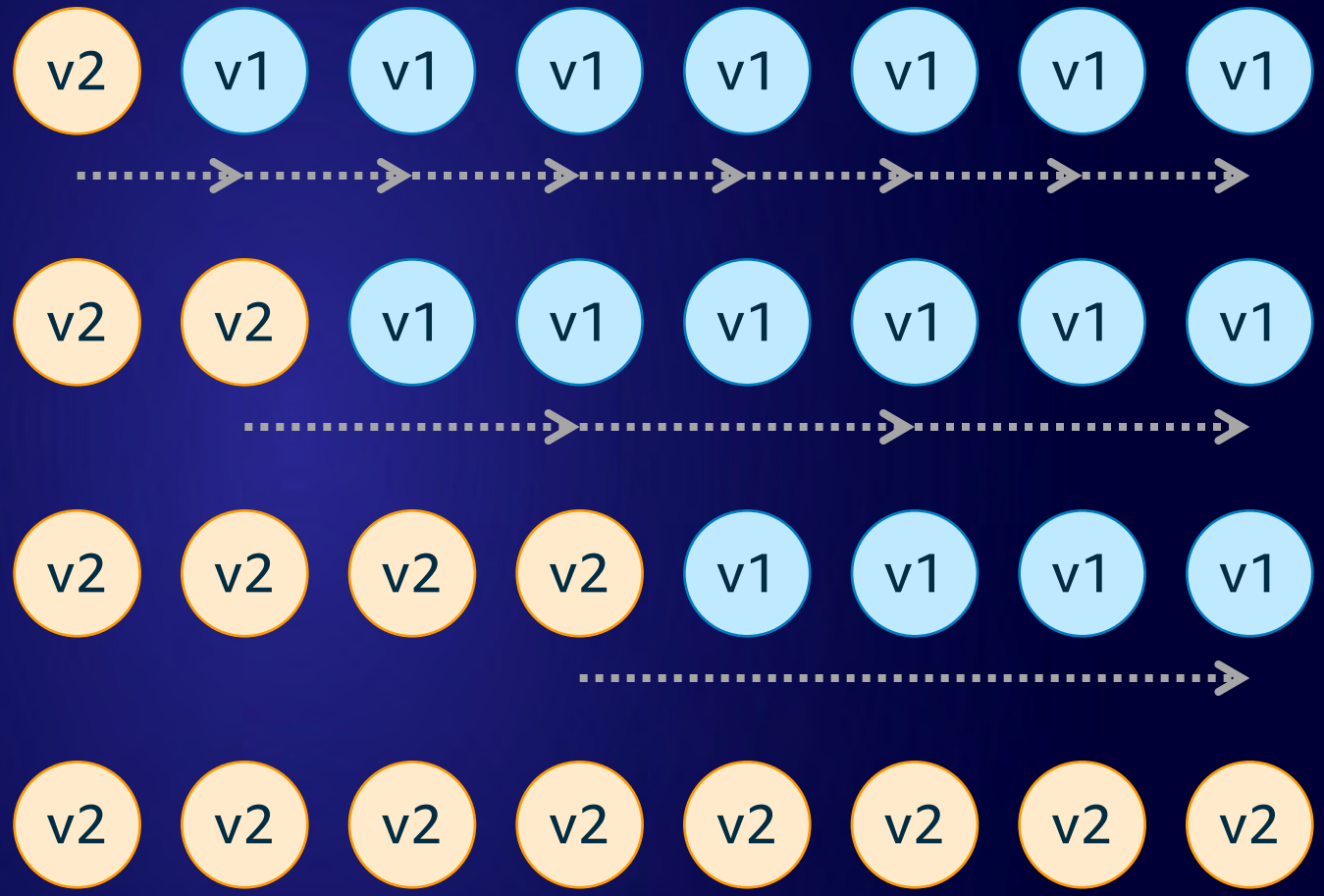
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



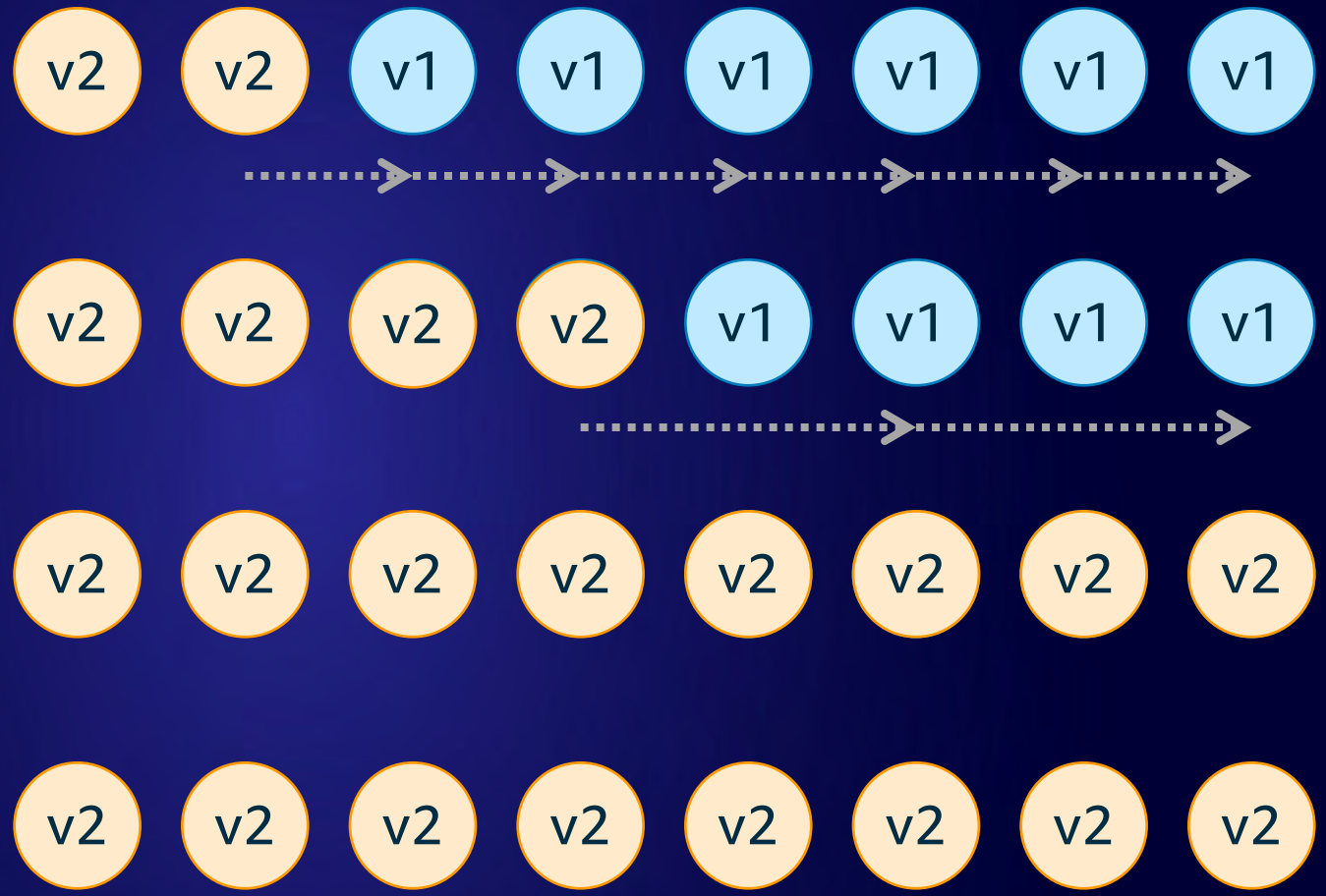
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



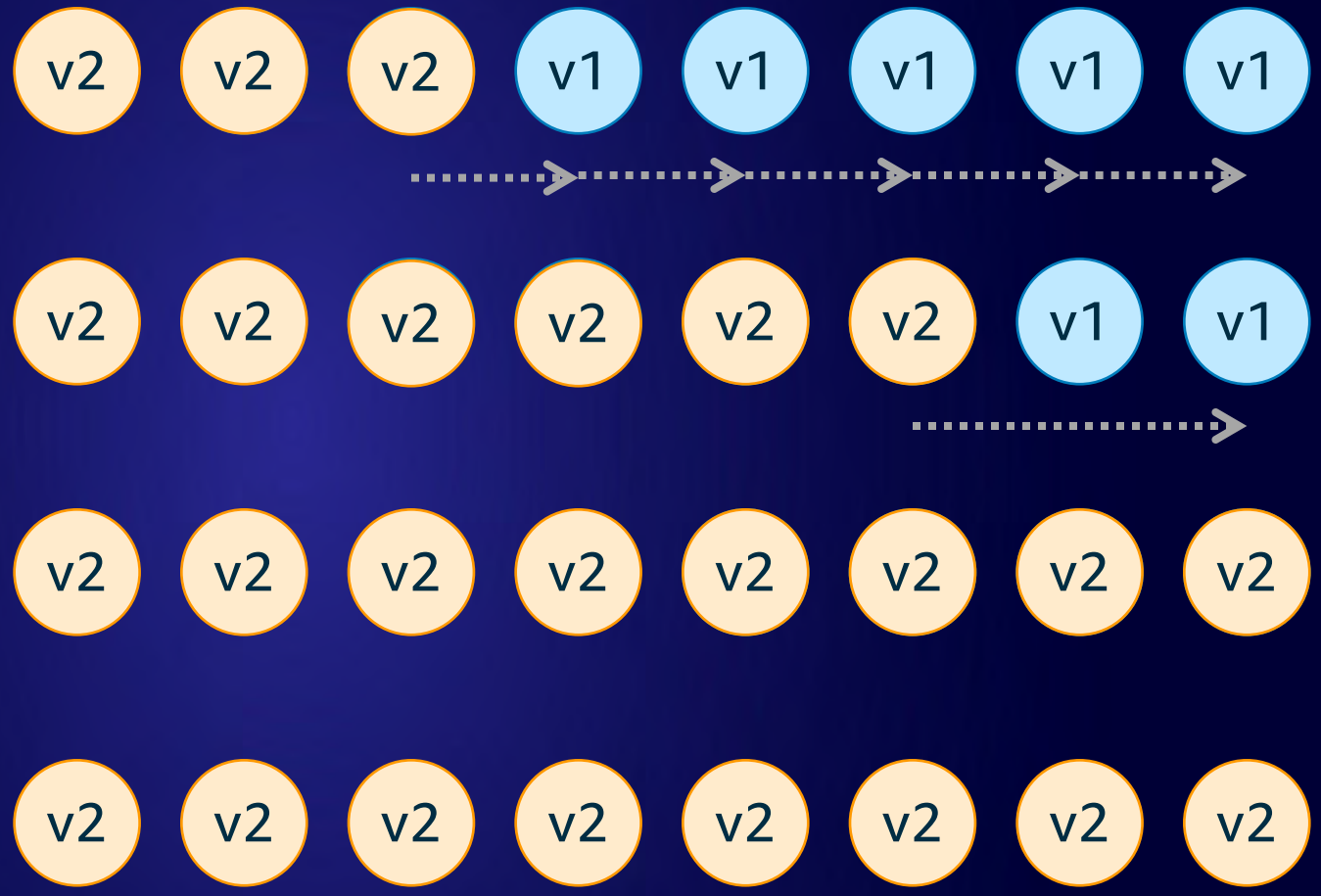
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



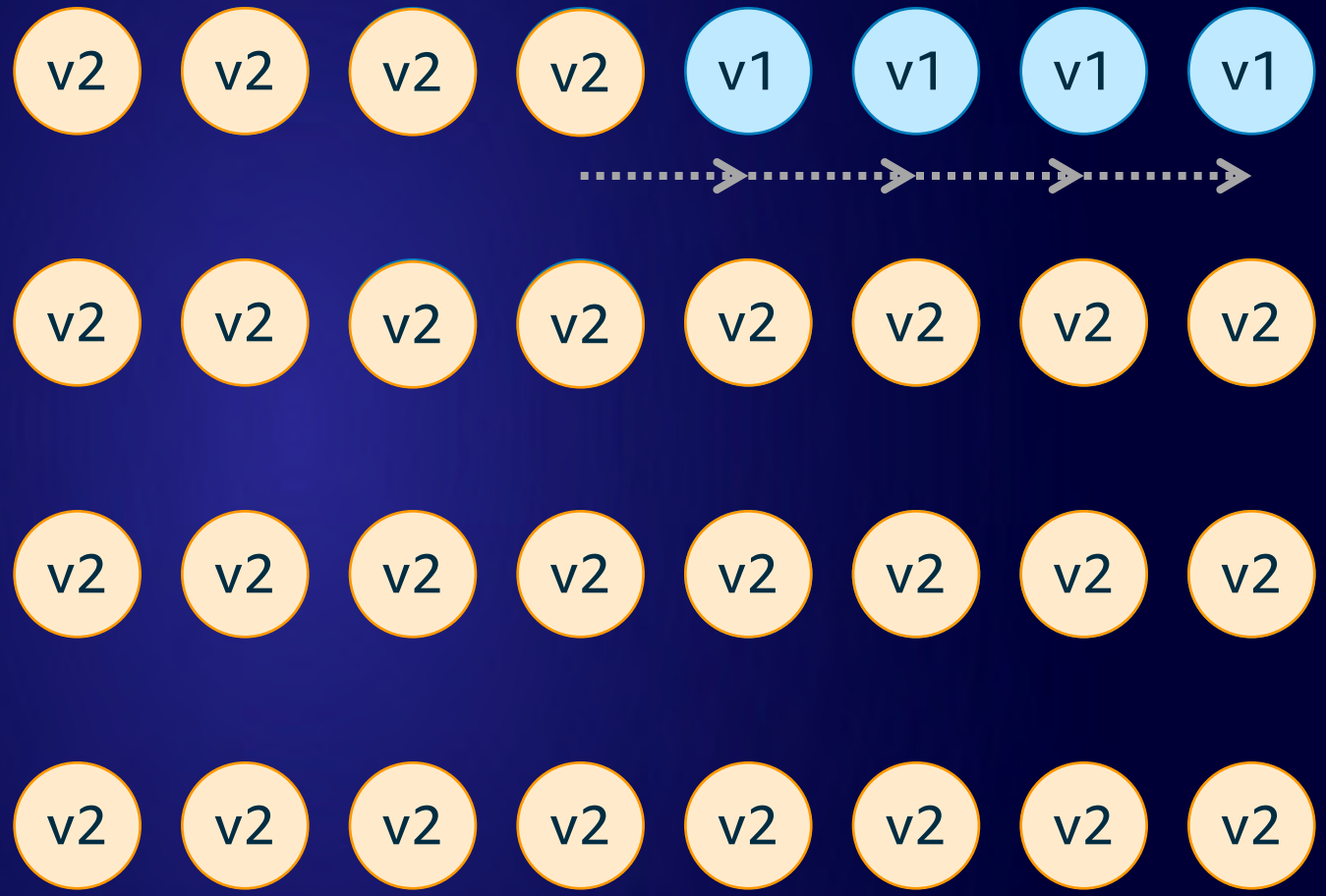
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



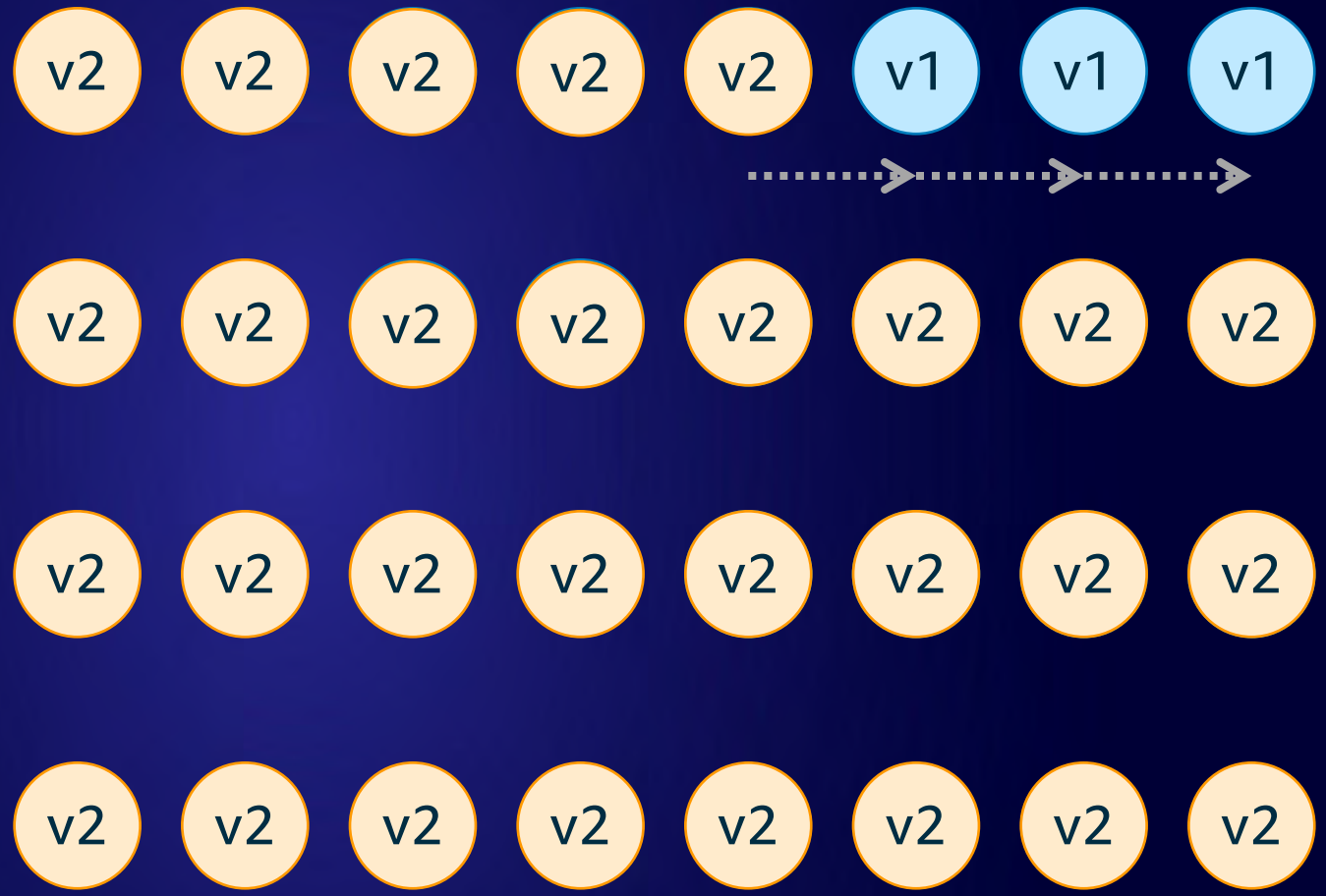
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



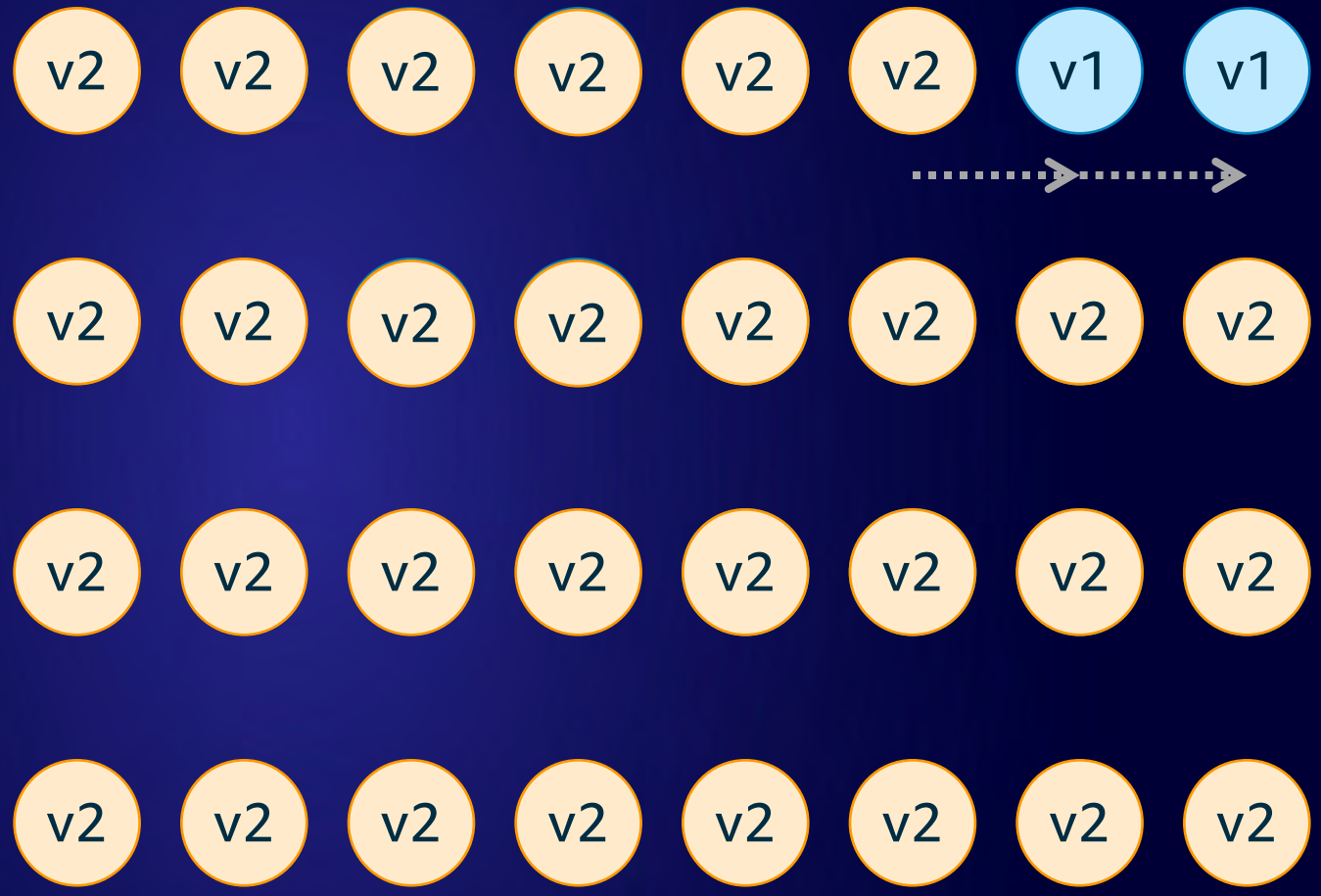
EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0



EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

All-at-once
Min. healthy hosts = 0

EC2 への In-Place デプロイイメージ

One-at-a-time
Min. healthy hosts = 99%

[Custom]
Min. healthy hosts = 75%

Half-at-a-time
Min. healthy hosts = 50%

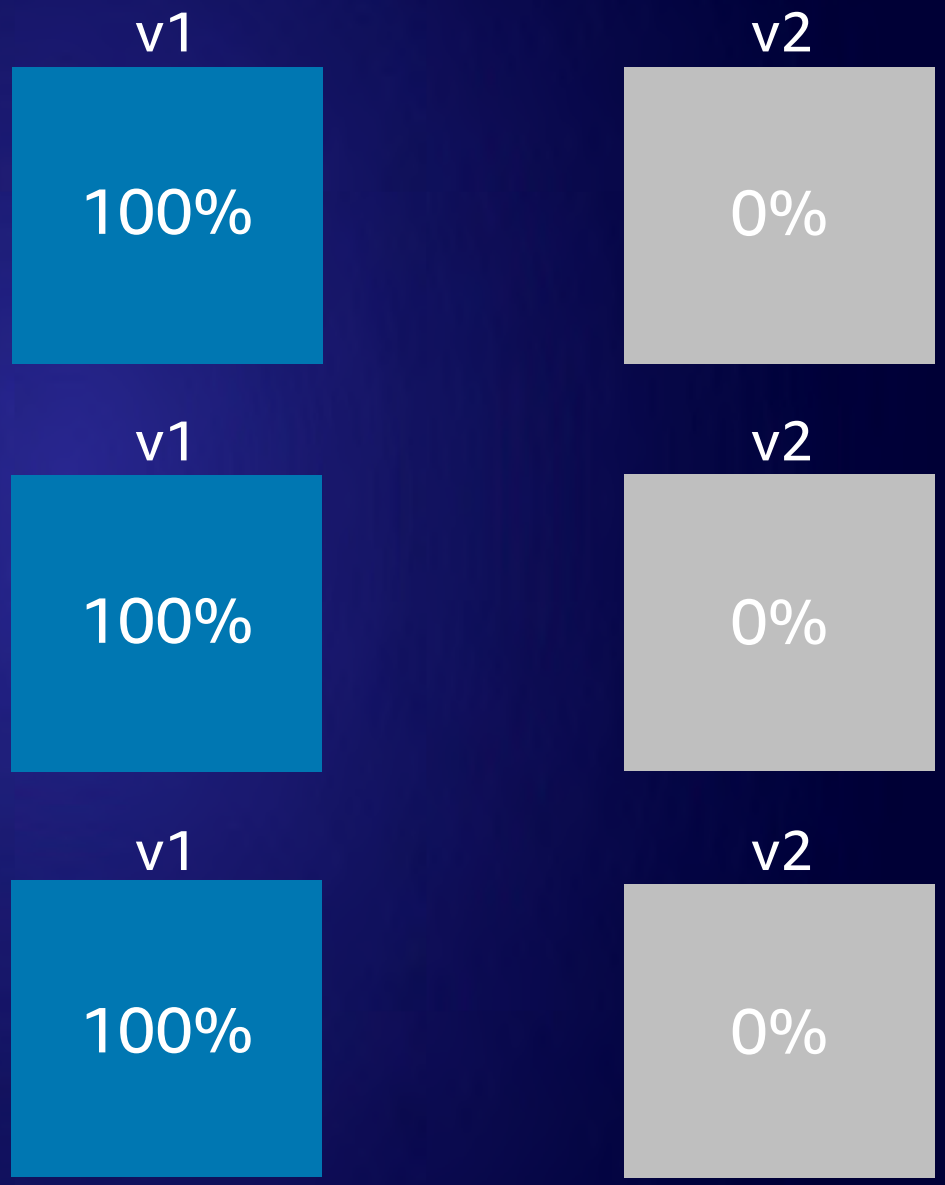
All-at-once
Min. healthy hosts = 0

Lambda, ECS へのデプロイ イメージ

Linear
Step 25%, Interval 10min

Canary
Step 25%, Interval 10min

All-at-once



Lambda, ECS へのデプロイ イメージ

Linear
Step 25%, Interval 10min

Canary
Step 25%, Interval 10min

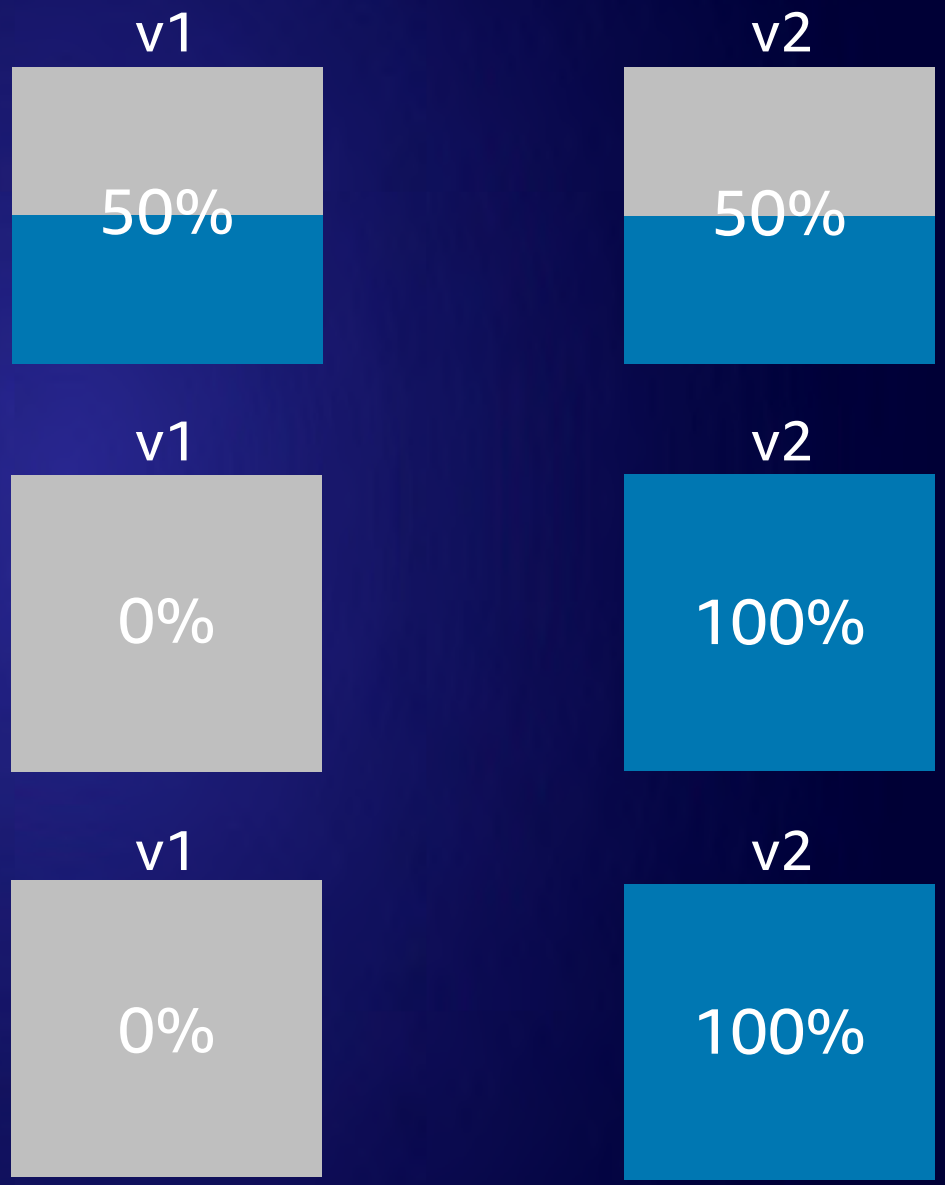
All-at-once

Lambda, ECS へのデプロイ イメージ

Linear
Step 25%, Interval 10min

Canary
Step 25%, Interval 10min

All-at-once



Lambda, ECS へのデプロイ イメージ

Linear
Step 25%, Interval 10min

Canary
Step 25%, Interval 10min

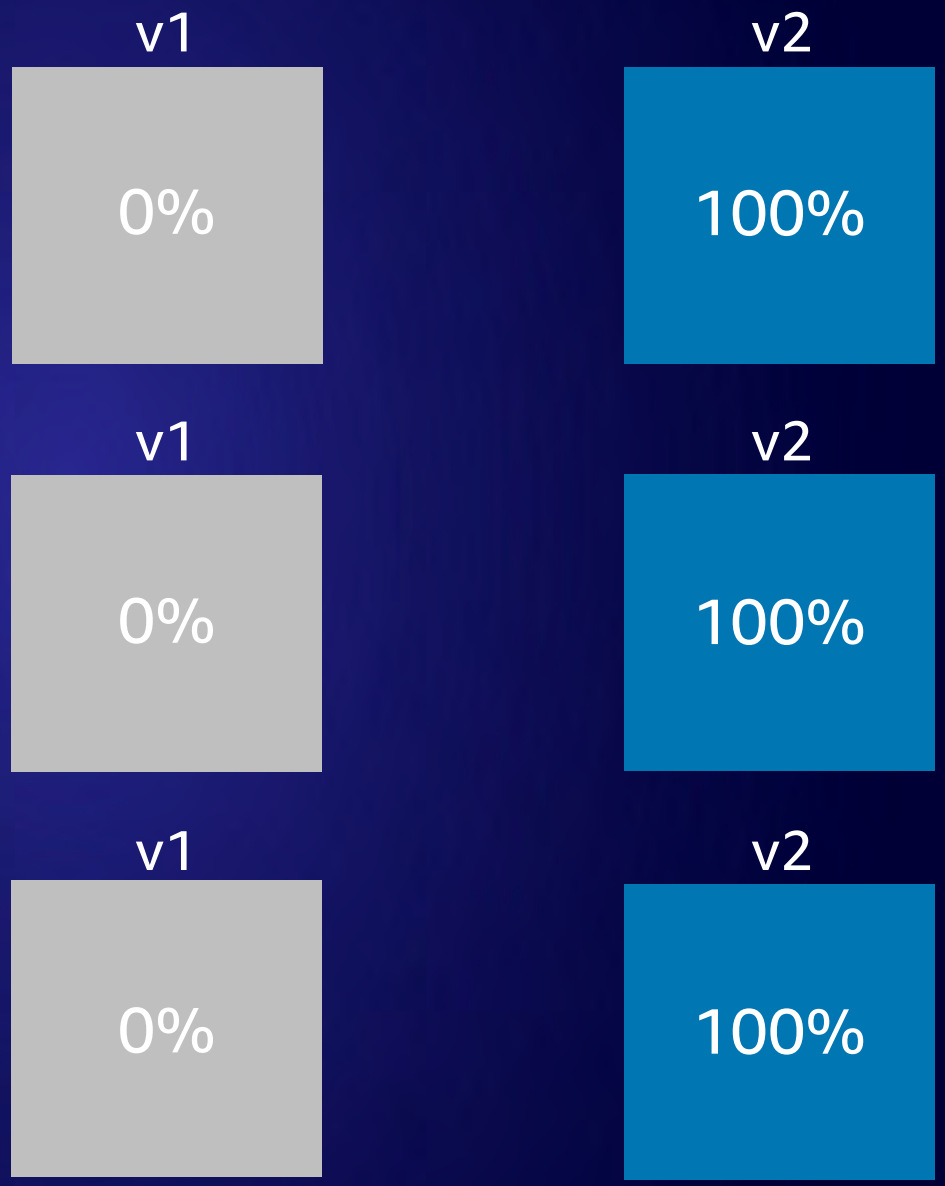
All-at-once

Lambda, ECS へのデプロイ イメージ

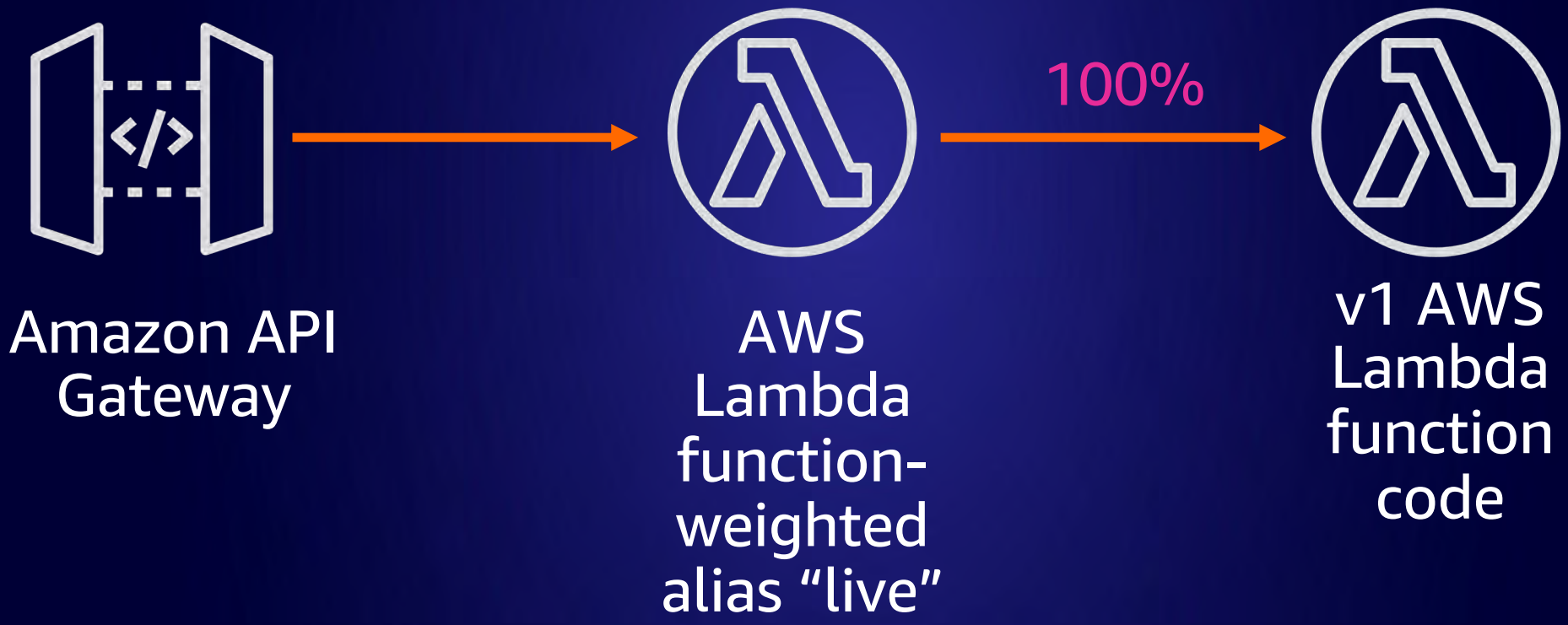
Linear
Step 25%, Interval 10min

Canary
Step 25%, Interval 10min

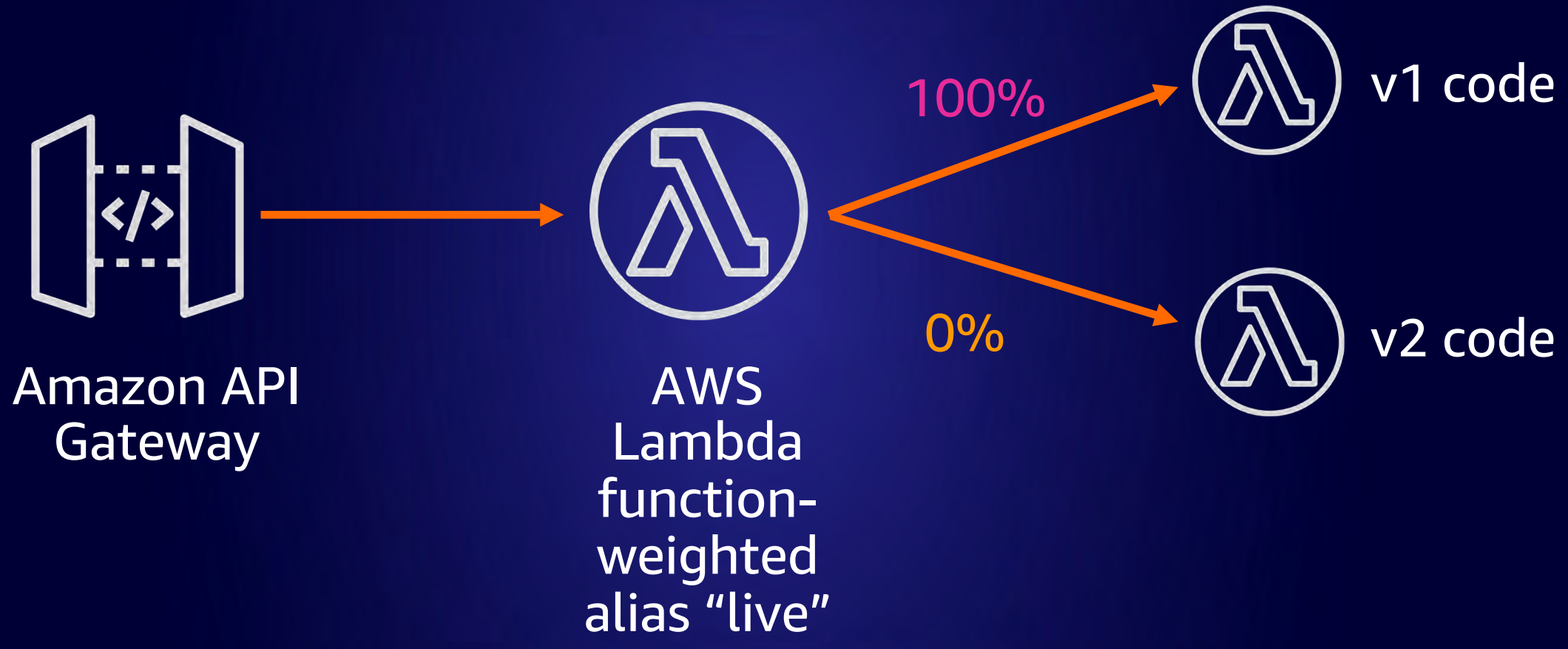
All-at-once



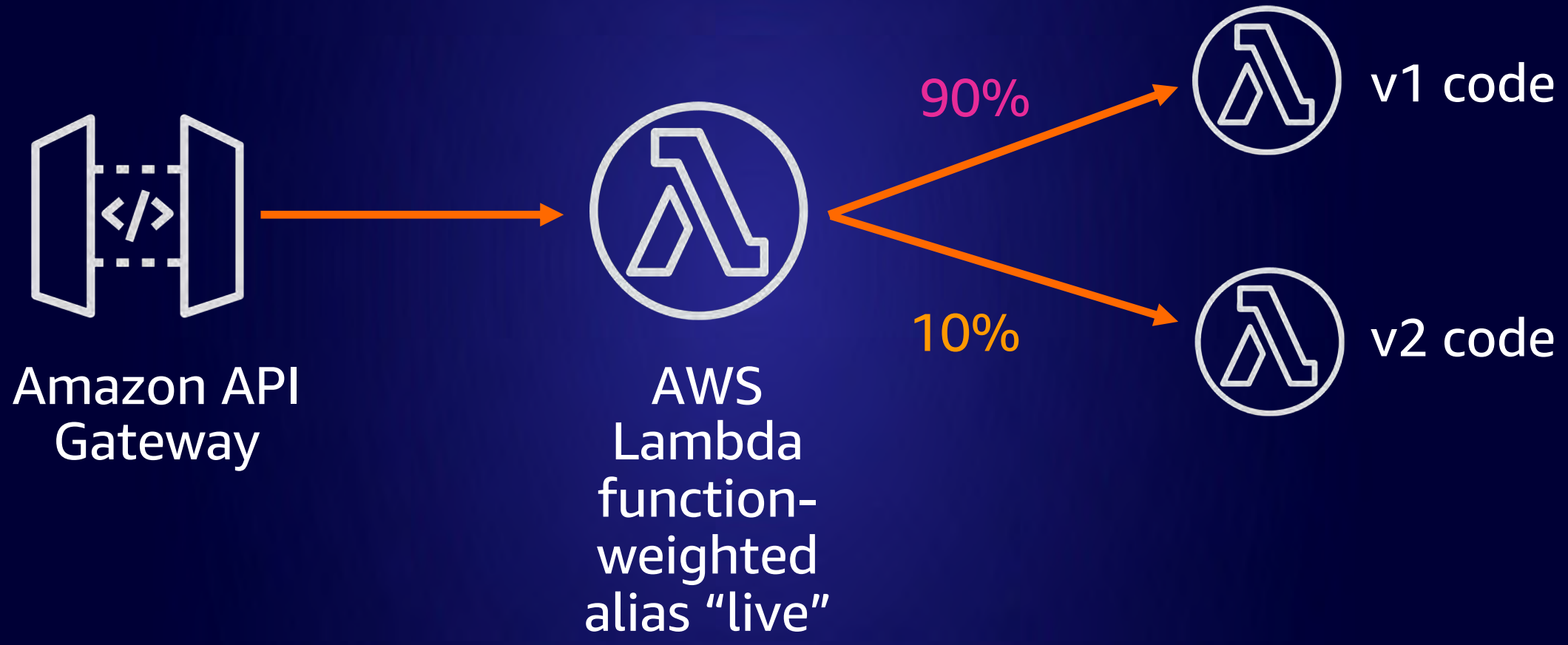
Lambda カナリアデプロイイメージ



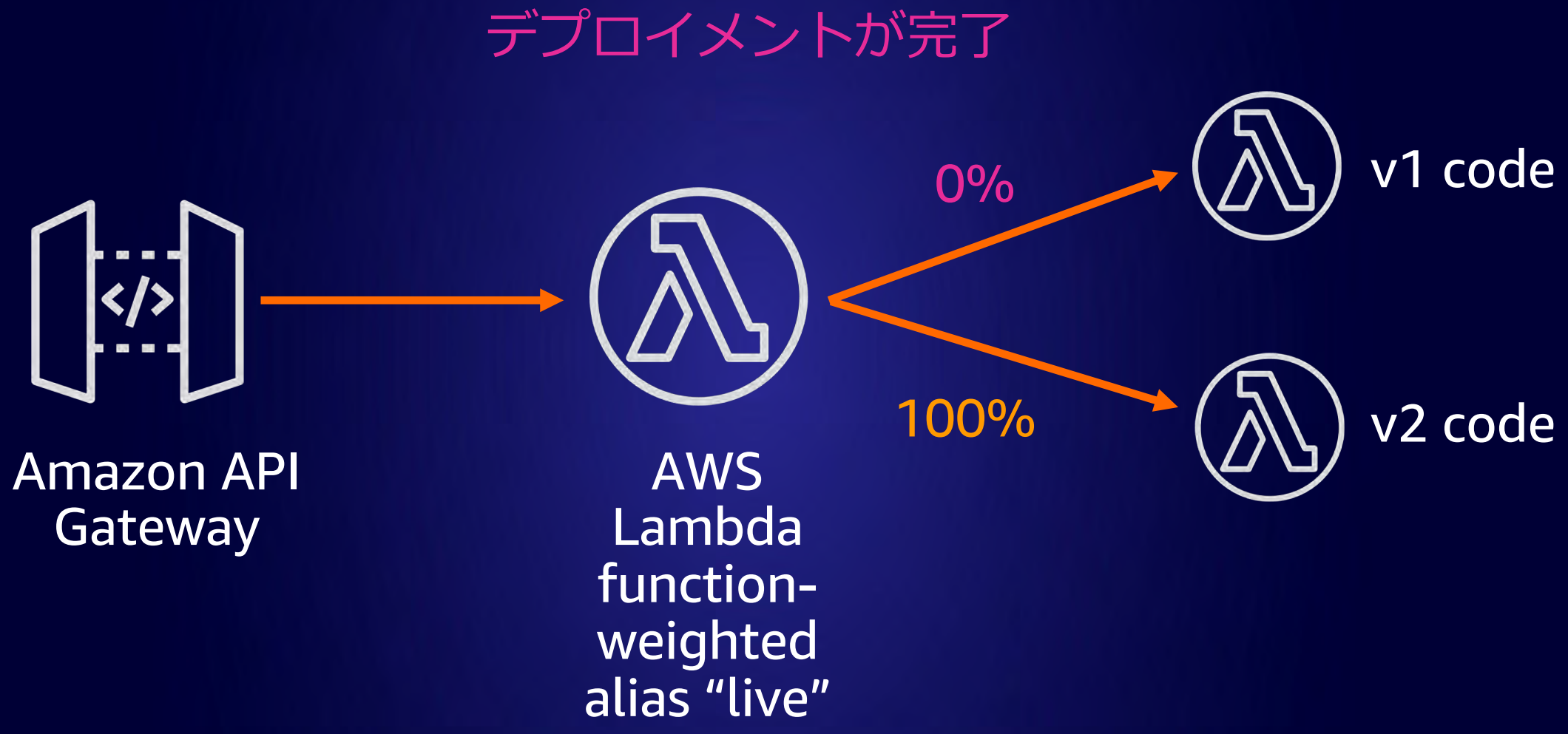
Lambda カナリアデプロイイメージ



Lambda カナリアデプロイイメージ

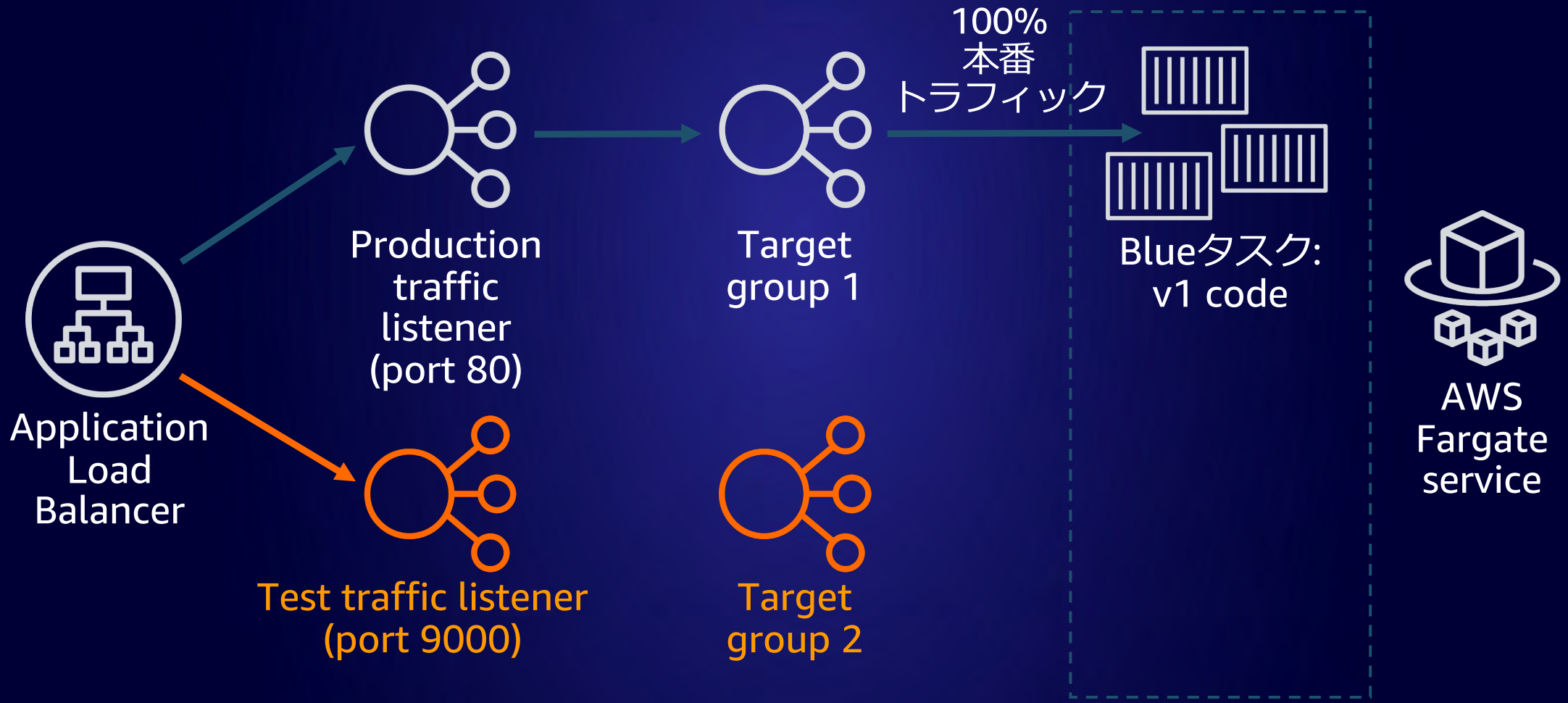


Lambda カナリアデプロイイメージ

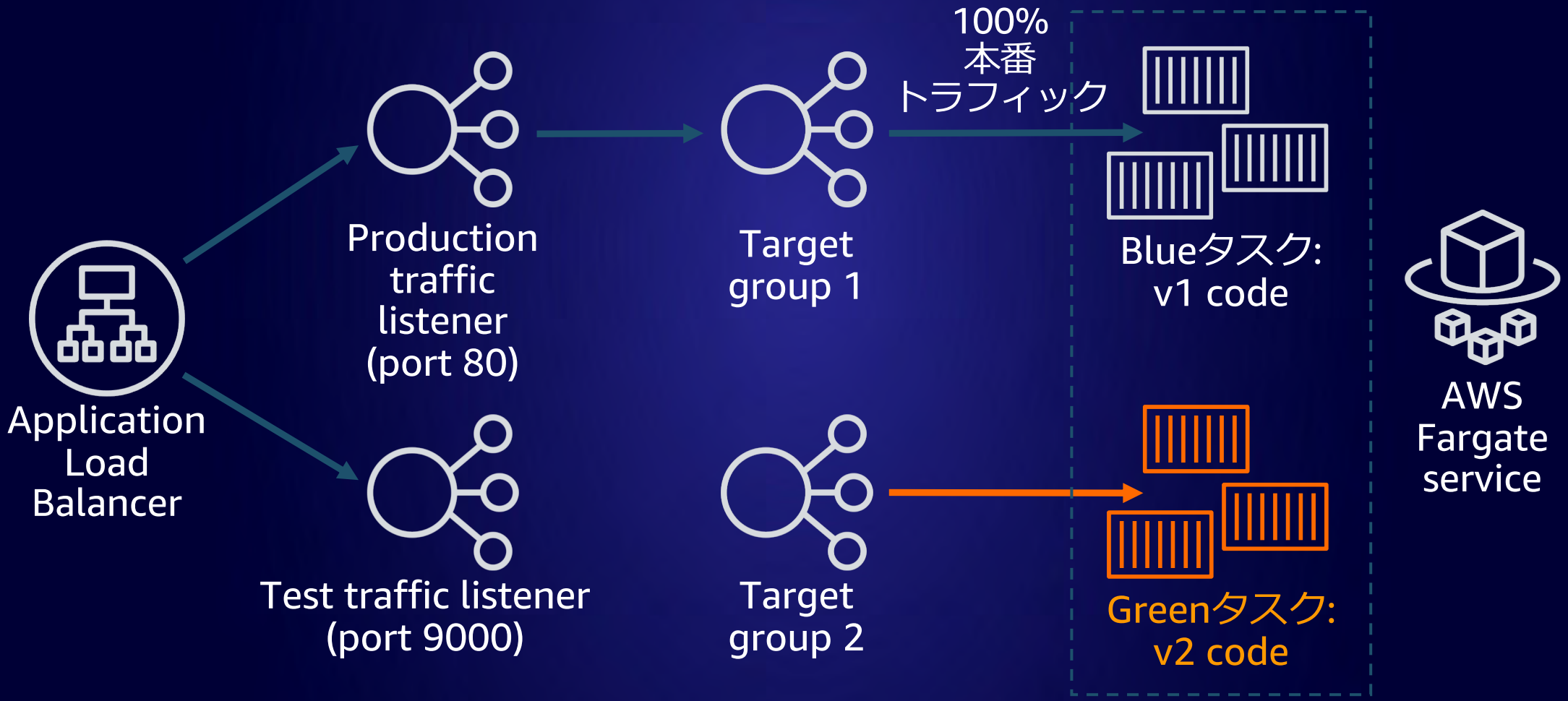


ECS Blue/Green デプロイイメージ

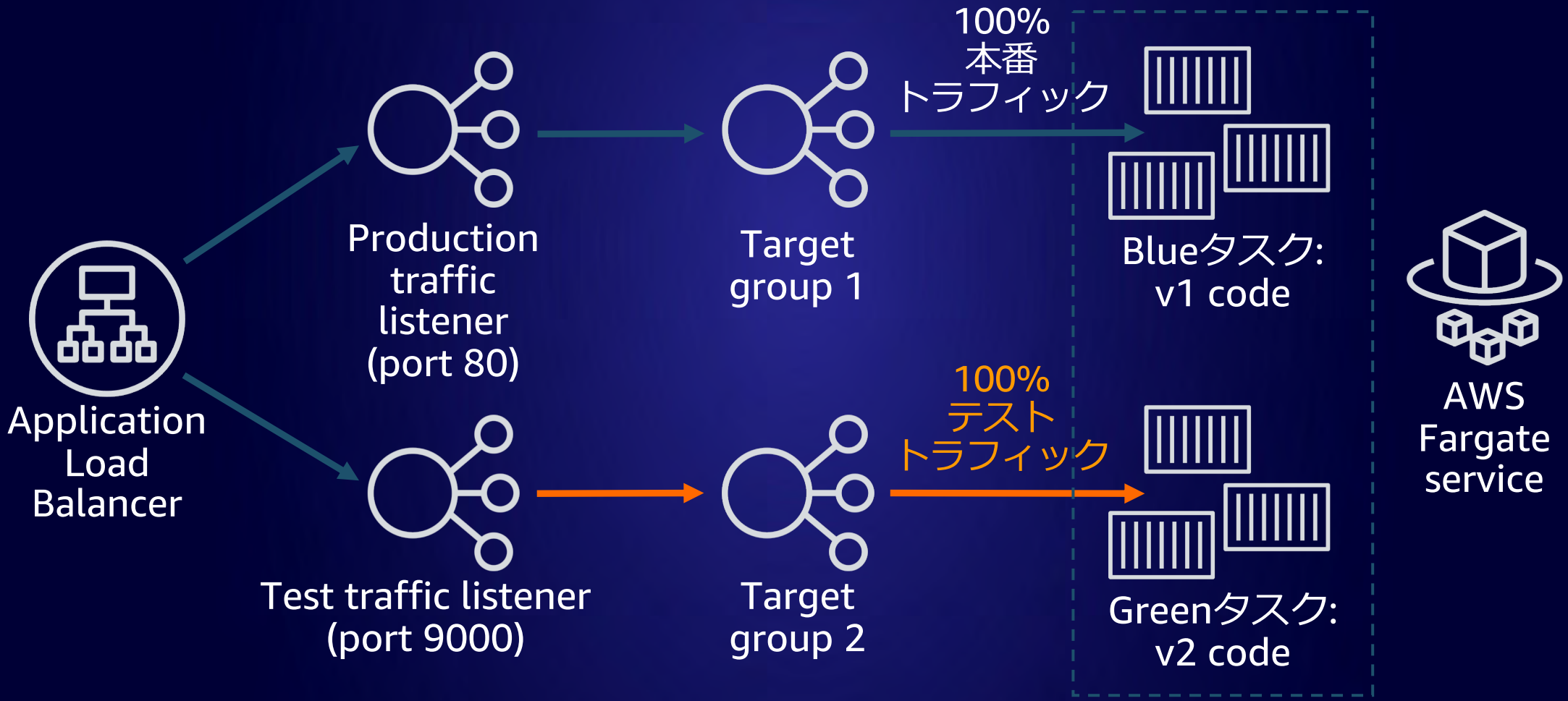
ECS Blue/Green デプロイイメージ



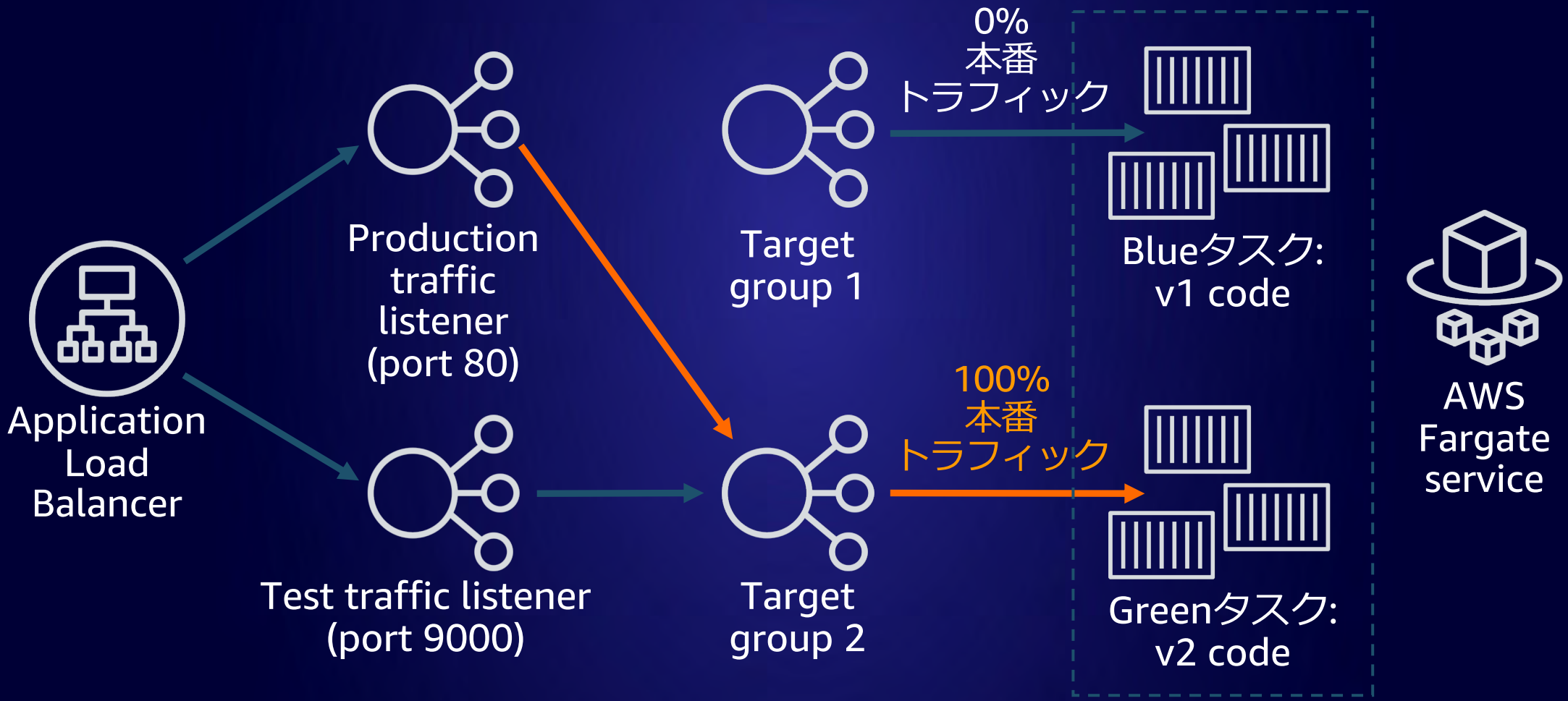
ECS Blue/Green デプロイイメージ



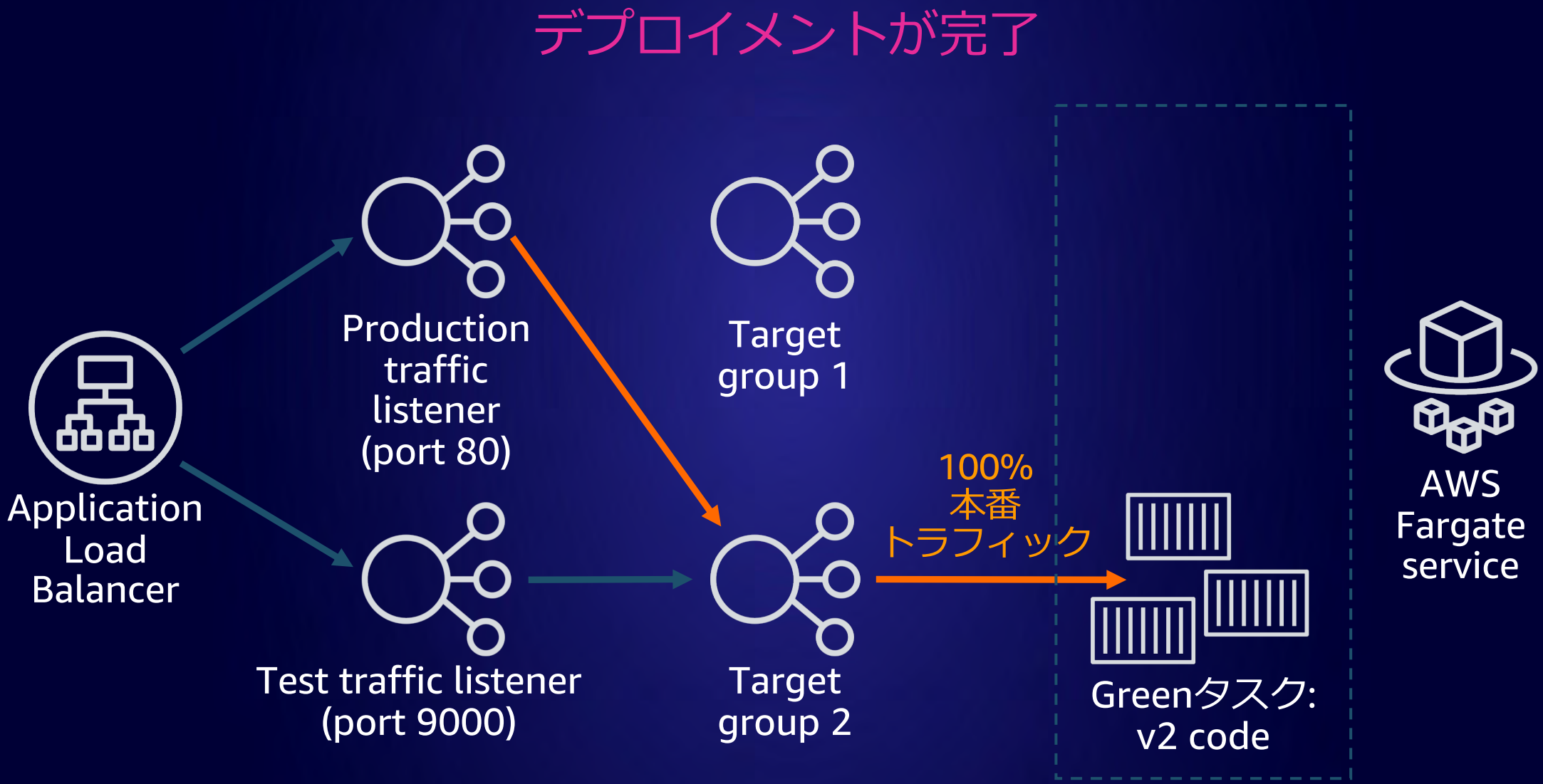
ECS Blue/Green デプロイイメージ



ECS Blue/Green デプロイイメージ



ECS Blue/Green デプロイイメージ

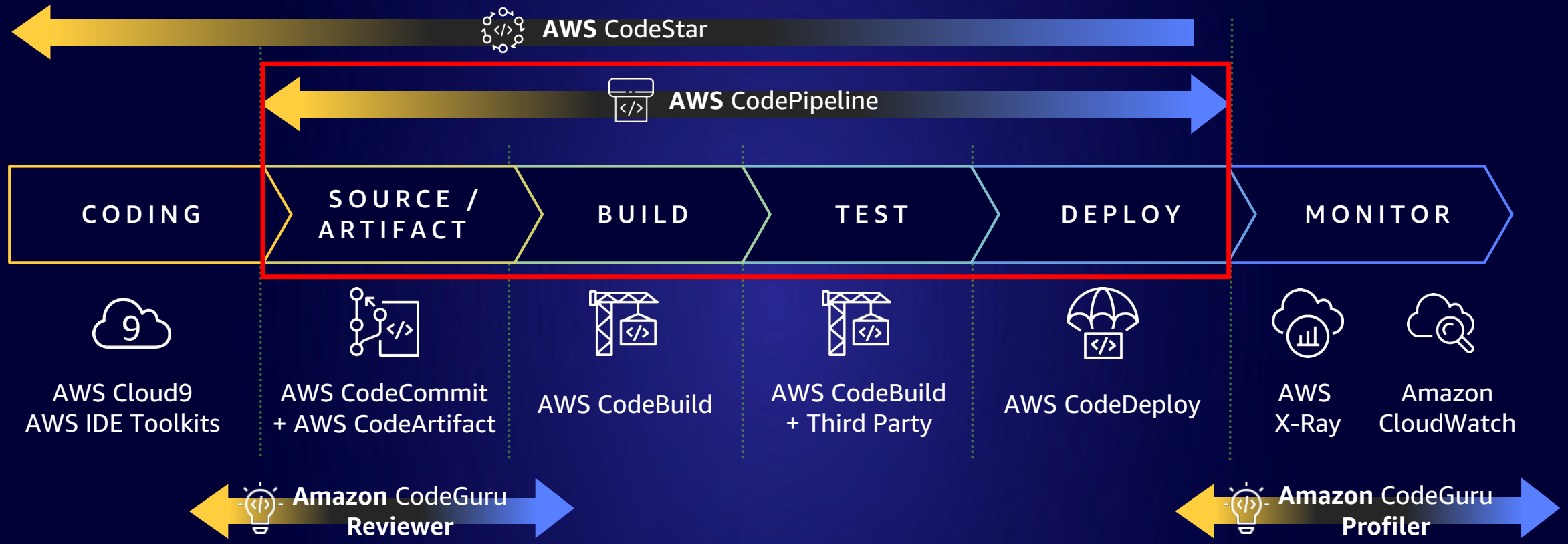


AWS CodePipeline

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス

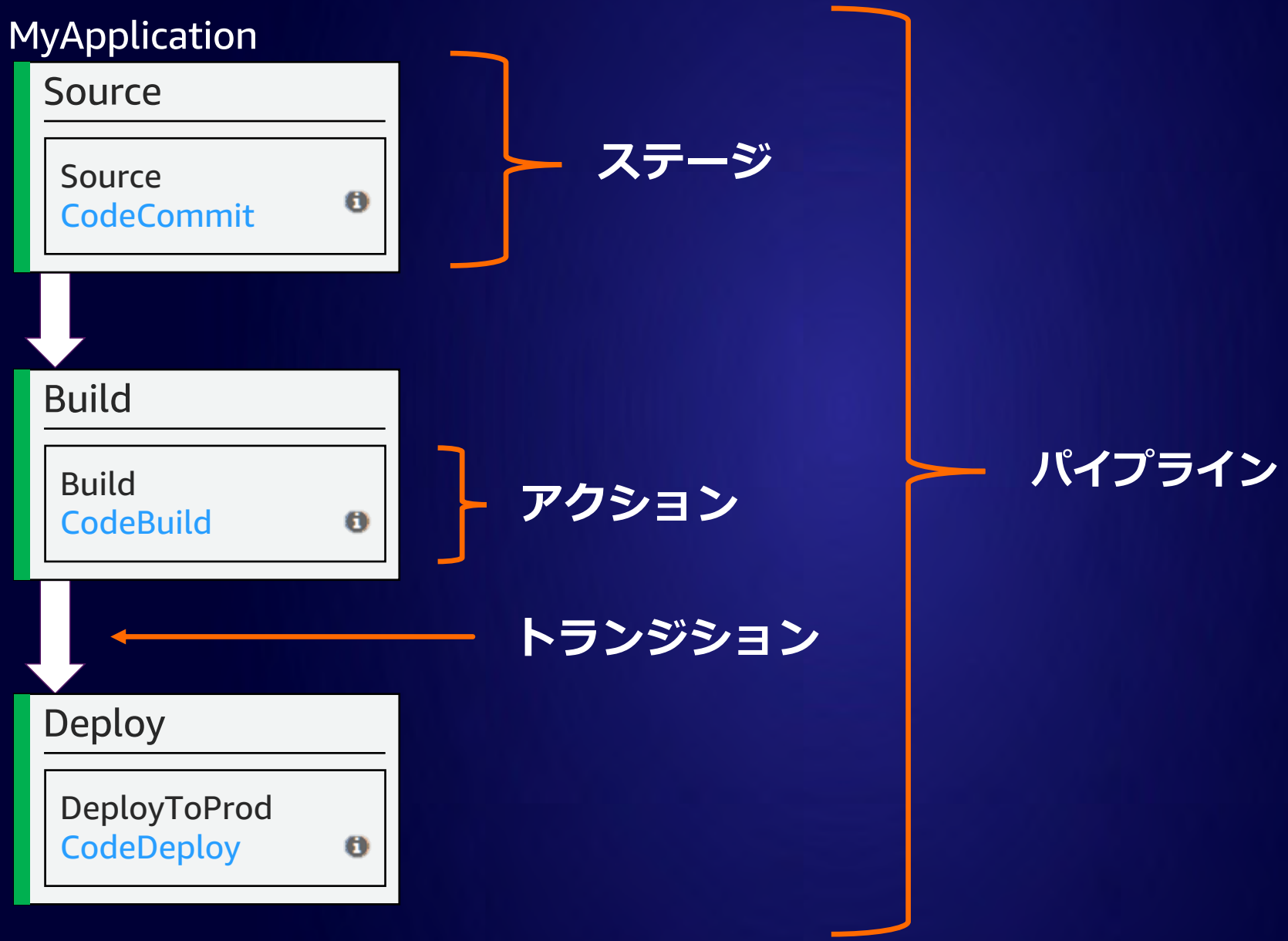
再掲



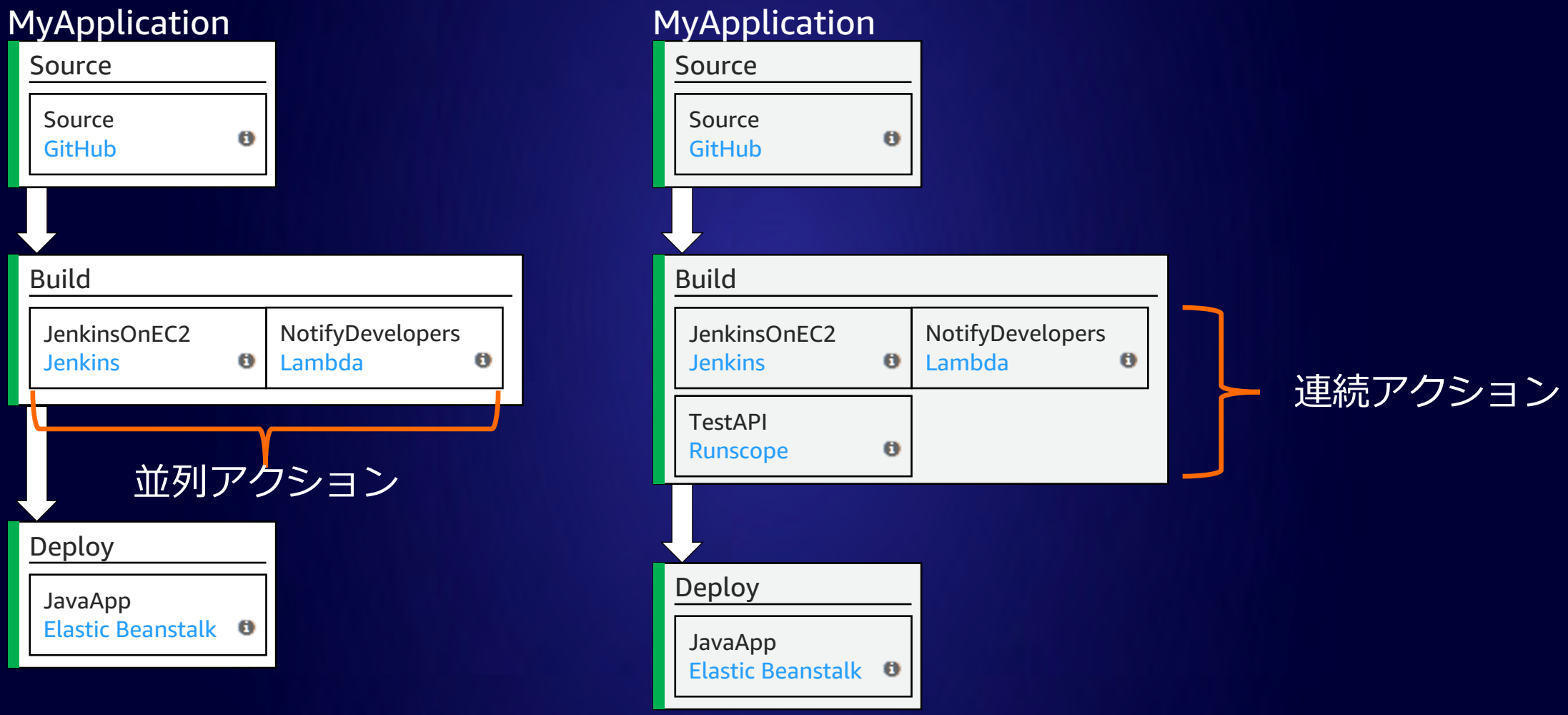
AWS CodePipeline

- ソフトウェアリリースプロセスを自動化する
  フルマネージドサービス
- コードが変更されるたびにワークフローに従って、
  ビルド、テスト、デプロイを実施
- リリースプロセスをモデル化
- AWS サービスやサードパーティーツールと統合

CodePipeline の全体像



並列アクションと連続アクション



アクションプロバイダー

CodePipeline の各ステージで以下のようなサービスと結合できる。

ソース

- AWS CodeCommit
- Amazon ECR
- Amazon S3
- GitHub
- GitHub Enterprise
- Bitbucket

ビルド

- AWS CodeBuild
- Jenkins
- CloudBees
- TeamCity

テスト

- AWS CodeBuild
- AWS Device Farm
- BlazeMeter
- Ghost Inspector
- Micro Focus StomRunner
- Nouvola
- Runscope

承認

- Manual

デプロイ

- AWS CloudFormation
- AWS CodeDeploy
- AWS Elastic Beanstalk
- AWS Service Catalog
- Alexa Skills Kit
- Amazon ECS
- Amazon ECS (Blue/Green)
- Amazon S3

アクションプロバイダー

CodePipeline の各ステージで以下のようなサービスと結合できる。

ソース

- AWS CodeCommit
- Amazon ECR
- Amazon S3
- GitHub
- GitHub Enterprise
- Bitbucket

ビルド

- AWS CodeBuild
- Jenkins
- CloudBees
- TeamCity

テスト

- AWS CodeBuild
- AWS Device Farm
- BlazeMeter
- Ghost Inspector
- Micro Focus StomRunner
- Nouvola
- Runscope

承認

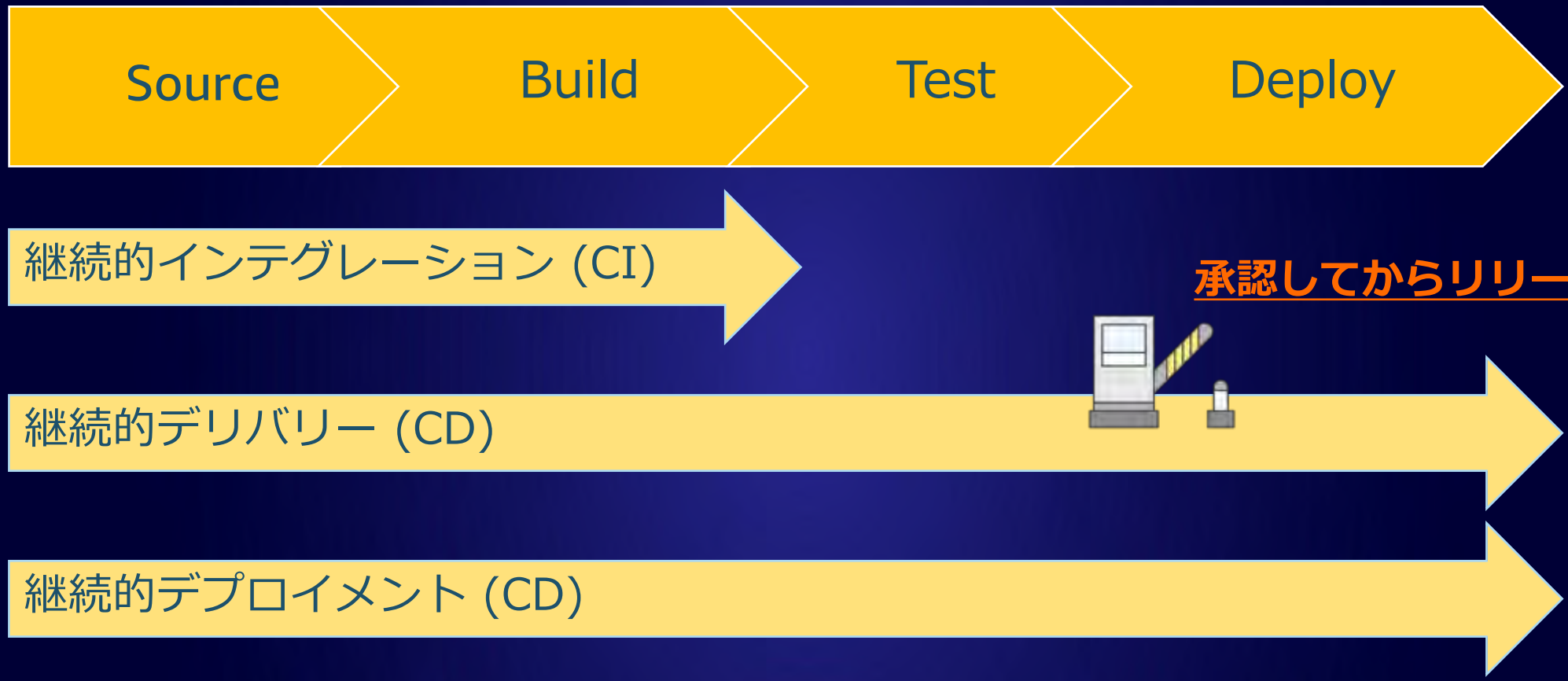
- Manual

デプロイ

- AWS CloudFormation
- AWS CodeDeploy
- AWS Elastic Beanstalk
- AWS Service Catalog
- Alexa Skills Kit
- Amazon ECS
- Amazon ECS (Blue/Green)
- Amazon S3

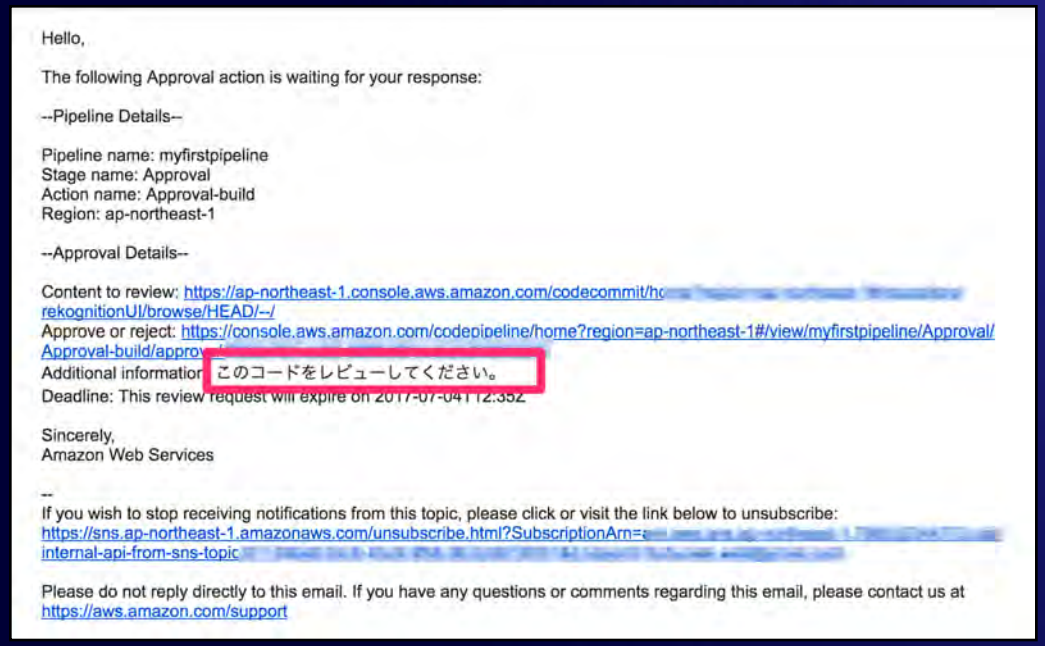
リリースプロセスのレベル

再掲



承認アクション

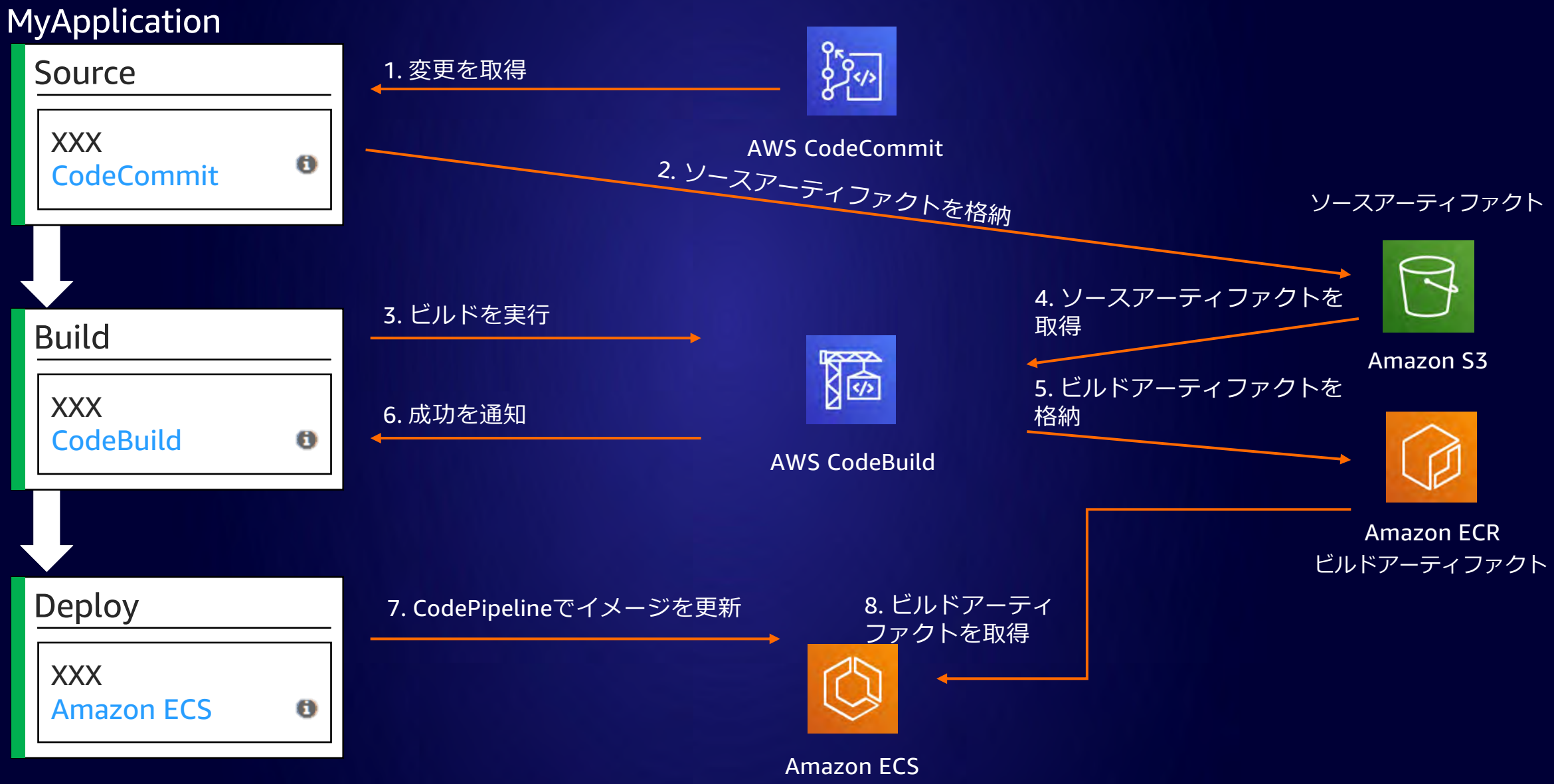
- 承認されたら次のステージへ
- SNS トピックをあわせて指定することでメールで通知
- 承認アクションの例
  - 次のステージに進む前にコードレビューや変更管理レビューを依頼
  - 手動の品質管理テストを最新バージョンのアプリケーションに実施を依頼
  - リリース前にビルドされた成果物の統合確認を依頼



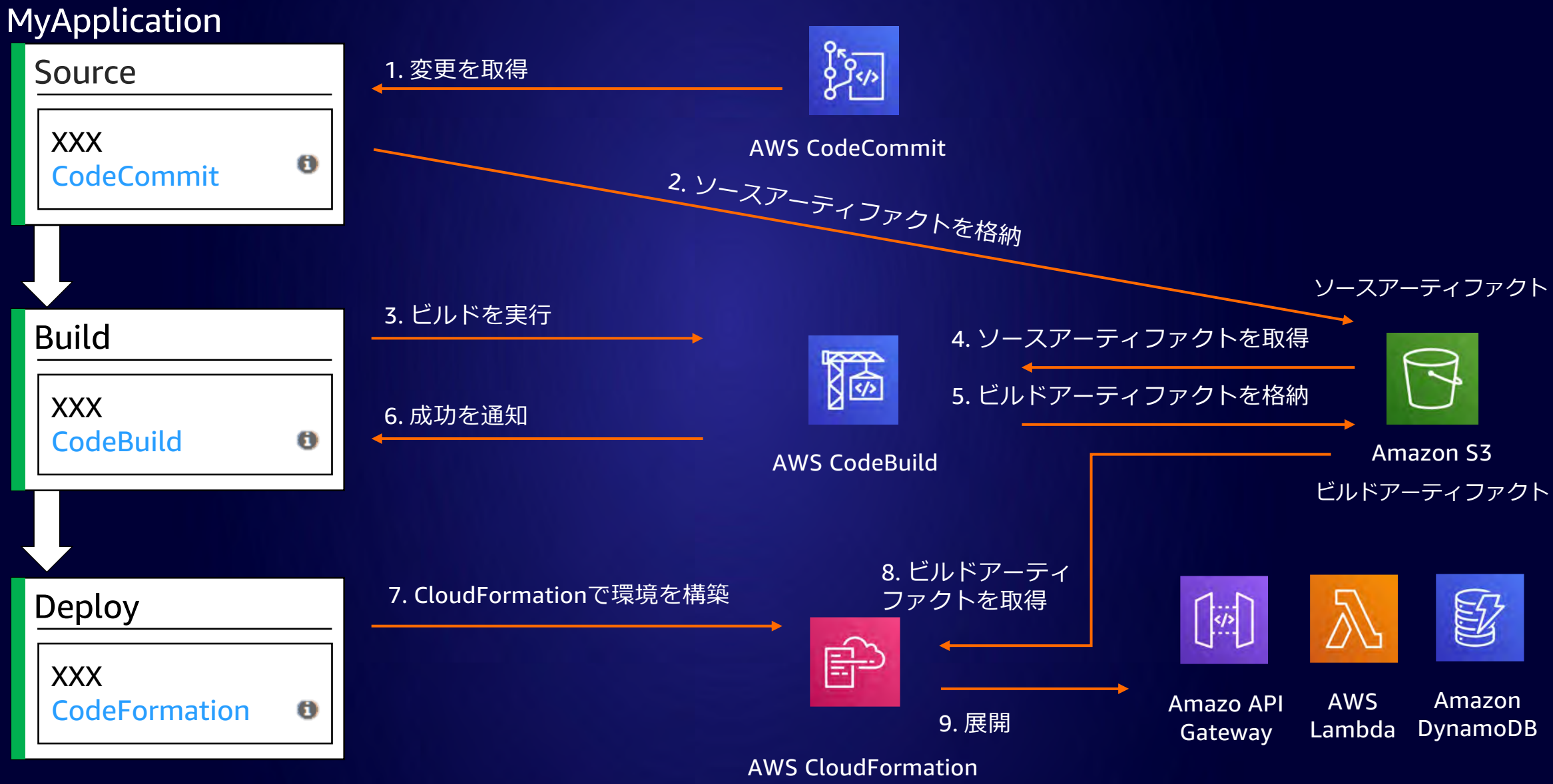
構成例：承認アクションを追加したパイプライン

- 統合環境にデプロイ後に手動でテスト
- その結果を元に、リリースの続行、中止を判断

構成例: Amazon ECS での CI/CD パイプライン



構成例: サーバーレス での CI/CD パイプライン



CI/CD はじめの一歩

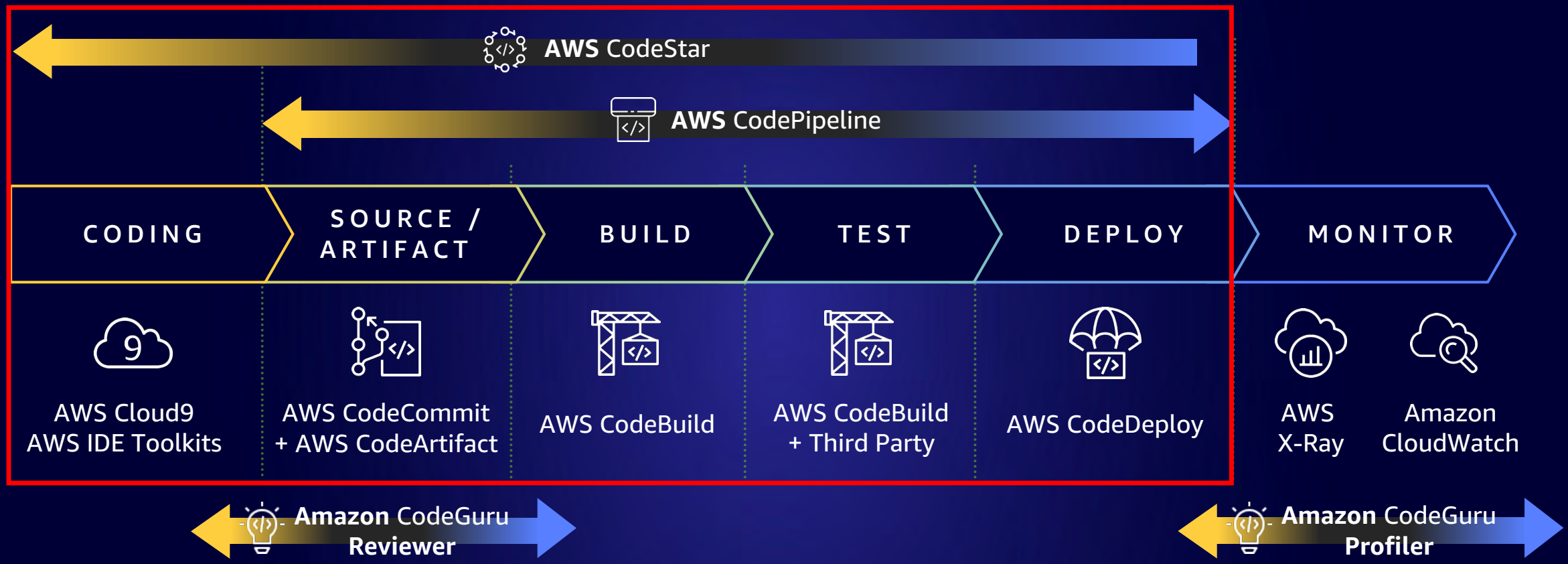
© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS CodeStar

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

CI/CD に関連する AWS サービス

再掲



MODEL　AWS CloudFormation　AWS Cloud Development Kit　AWS SAM

AWS CodeStar

- 様々なプロジェクトテンプレート
- AWS 上での開発をわずか数分で開始
- チームをまたがった開発をセキュアに
- テンプレートから構築した環境をカスタマイズ可能

豊富なプロジェクトテンプレート

AWS サービス

- AWS Elastic Beanstalk
- Amazon EC2
- AWS Lambda

アプリケーションタイプ

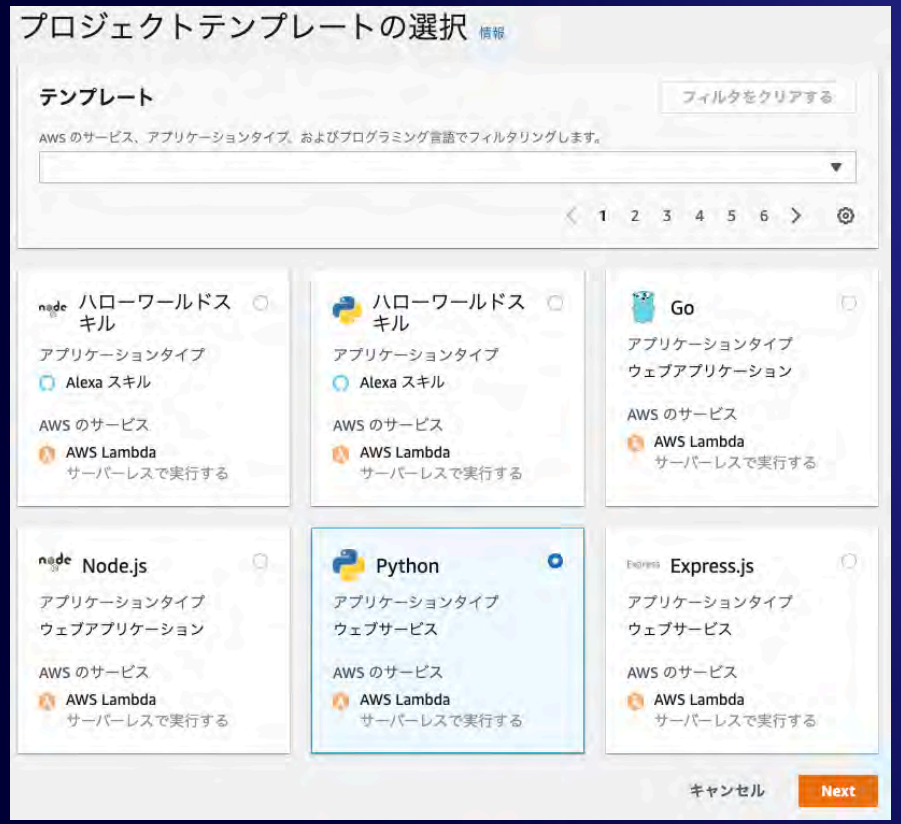
- ウェブアプリケーション
- ウェブサービス
- 静的ウェブサイト
- Alexa スキル
- AWS Config ルール

プログラミング言語

- C#
- Go
- HTML 5
- Java
- Node.js
- PHP

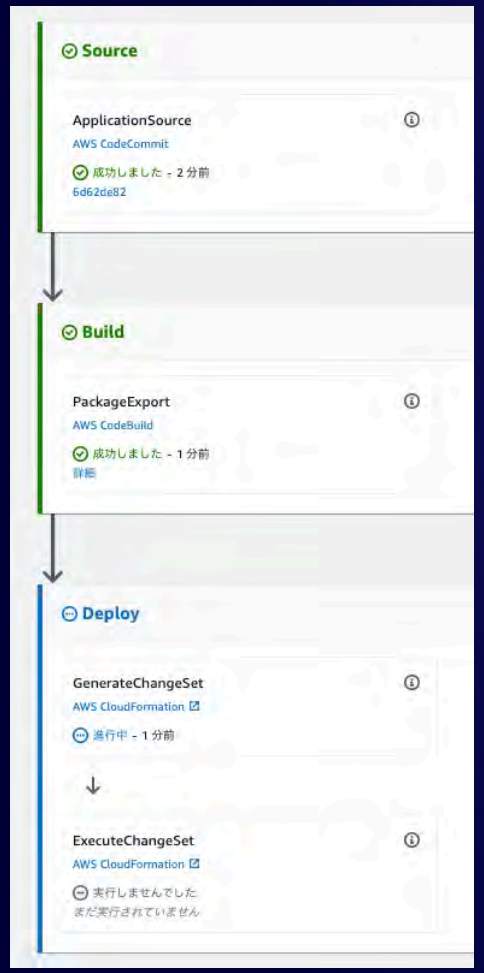
CodeStar プロジェクトの作成

- テンプレートを選択
- プロジェクト名を入力、リポジトリを選択

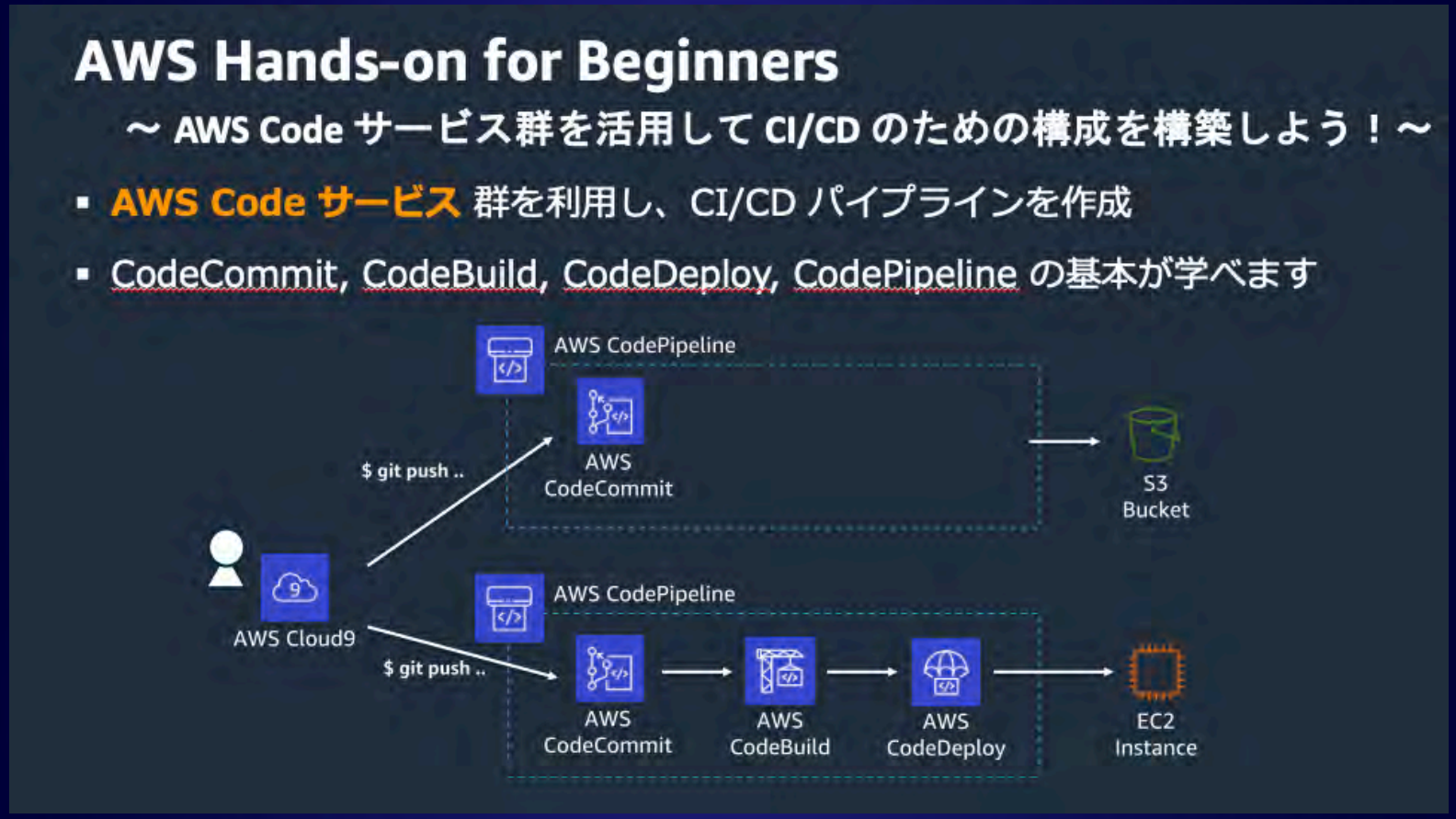


自動生成された環境

- サンプルコードを変更することでパイプラインを実行可能



AWS Hands-on for Beginners



https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/

まとめ

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

まとめ

- 変化に素早く対応するには、継続的インテグレーション / 継続的デリバリー (CI/CD) が必須
- AWS CodeStar で CI/CD 環境を簡単に構築、体験できる
- Code シリーズは単体での利用も可能
- 自動化できる作業は自動化し、より価値を生む作業に開発リソースの投入を！

参考資料

- AWS サービス別資料
  - Developer Tools セクション
  - https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/#dev-tools
- AWS ハンズオン資料
  - AWS Code サービス群を活用して、CI/CD のための構成を構築しよう！
  - https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-hands-on/

関連セッション

- AWS-02
  - Amazon CodeGuru 機械学習で実現する
    コードレビュー自動化とアプリケーションパフォーマンス最適化

- AWS-31
  - AWS で始める Infrastructure as Code

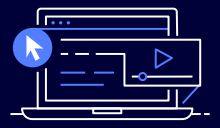
Thank you!

松本 雅博
アマゾン ウェブ サービス ジャパン株式会社
技術統括本部 西日本ソリューション部
ソリューションアーキテクト

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS トレーニングと認定

AWS クラウドをキャリアに活用してください



デジタルトレーニング

クラウドのスキルを構築する無料のオンデマンドコースを探索する

クラスルームトレーニング

エキスパートインストラクターによるトレーニングに参加する

AWS 認定の取得

業界で認められている認定を取得する

教育プログラム

AWS のスキルと経験を持つ人材に出会える

エンタープライズリソース

学習ニーズ分析とAWSランプアップガイドを活用する

詳細はこちら　https://aws.amazon.com/jp/training/

© 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved.