

AWS Builders Online Series

T3-1

はじめてのコンテナワークロード - AWS でのコンテナ活用の 第一歩

多田 慎也

アマゾン ウェブ サービス ジャパン合同会社
技術統括本部 西日本ソリューショングループ
ソリューションアーキテクト



自己紹介

名前：多田 慎也 (Tada Shinya)

所属：アマゾン ウェブ サービス ジャパン合同会社
技術統括本部 西日本ソリューショングループ
ソリューションアーキテクト



経歴：インフラエンジニア@Sler

好きなAWSサービス： Amazon ECS , AWS Fargate 

対象者、本セッションで学べること

- ・ 対象者
 - ・ コンテナに興味がある
 - ・ これからコンテナワークロードの構築を始める方
- ・ 本セッションで学べること
 - ・ なぜモダナイゼーションが必要か
 - ・ モダナイゼーションを進める上でのコンテナ活用のメリット
 - ・ AWS コンテナサービスを活用することで解決できる課題

関連セッションについて

- 3つの関連セッションで構成しています。
- 本セッションでは、モダナイゼーションにおいてなぜコンテナを利用するのか、コンテナの基本的な概念とメリットを解説した後、AWS 上のコンテナサービスの概要を説明します。

本セッションで学べること

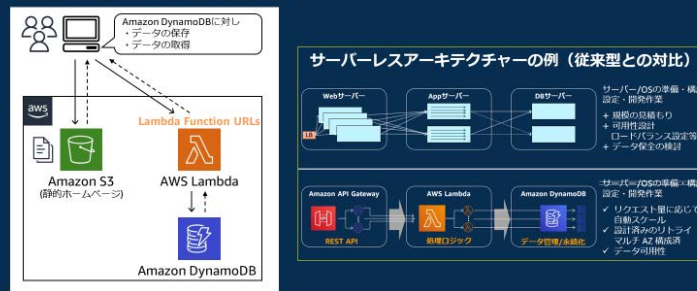
- 本セッションで学べること
 - なぜモダナイゼーションが必要か
 - モダナイゼーションを進める上でのコンテナ活用のメリット
 - AWS コンテナサービスを活用することで解決できる課題



T3-1: はじめてのコンテナワークロード - AWS でのコンテナ活用の第一歩

本セッションで学べること

- サーバーレスとは何か
- サーバーレスな Web アプリケーション



T3-2: はじめてのサーバーレス - AWS でのサーバーレスアプリケーション開発

本セッションで学べること

- オブザーバビリティとは何か、なぜ必要か
- AWS でオブザーバビリティを実現する方法



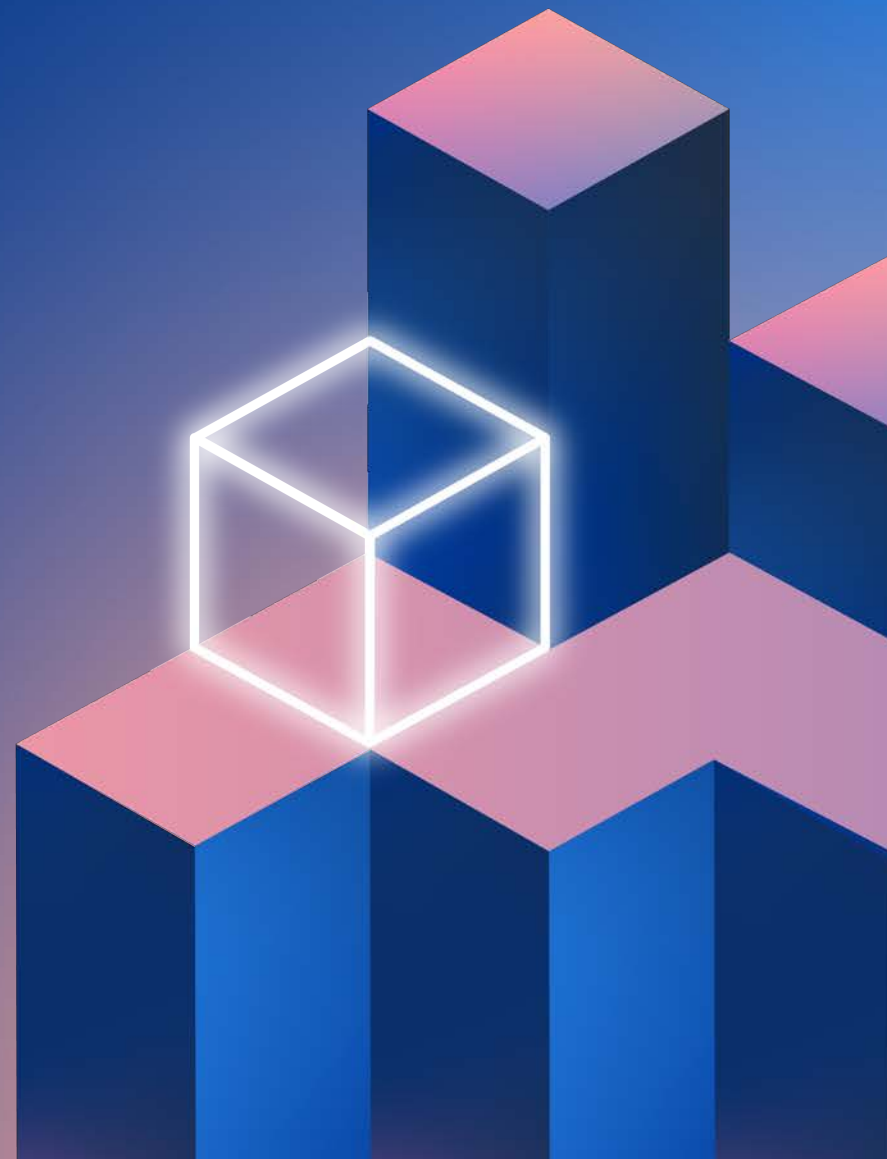
T3-3: AWS ではじめるオブザーバビリティ - システムのどこで・何が・なぜ起こってるのかを理解する

アジェンダ

- なぜモダナイゼーションが必要か
- なぜコンテナを利用するのか
- AWS コンテナサービス紹介
- コンテナサービスの始め方



なぜ モダナイゼーションが必要か



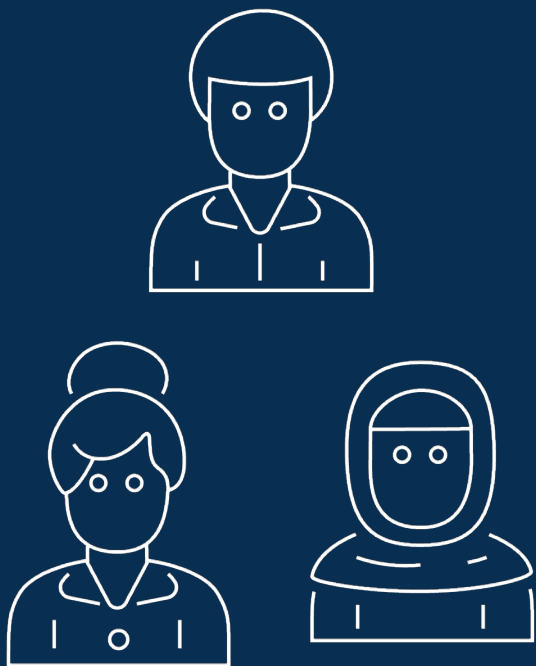
モダンアプリケーションとは

モダンアプリケーションは、最新のテクノロジー、アーキテクチャ、ソフトウェア配信プラクティス、運用プロセスを組み合わせたアプリケーションです。チーム全体の価値実現の速度、頻度、整合性、安全性を高めることができます。通常、モダンアプリケーションでは疎結合化された分散型テクノロジーを活用して、イベント駆動型のサーバーレスコンポーネントを重視することで面倒な単純作業を削減できるので、チームがより多くの時間を顧客のための価値創造に充てられるようになります。

引用 - モダンアプリケーション FAQ

<https://aws.amazon.com/jp/modern-apps/faqs/>

ITシステムを取り巻く環境の変化



ユーザーの多様化



手段の多様化



社会や環境の変化

モダナイゼーションの遅れが与えるビジネスインパクト



製品／機能の
リリースサイクルの
長期化

競争力の低下
ビジネス機会損失



非効率な運用、
コストオーバーヘッド

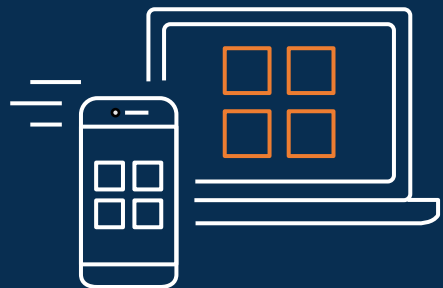
生産性の低下
差別化を産まない
スキルへの投資



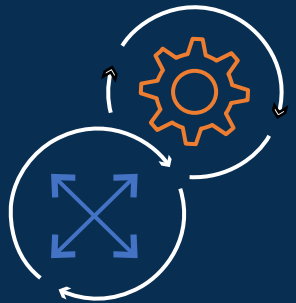
セキュリティ／
コンプライアンス対策の
遅れ

顧客からの信頼損失
レピュテーションリスク
の増加

モダンアプリケーションに求められる特性



アジリティ
(高速な市場投入)



パフォーマンス
スケーラビリティ



セキュリティ
レジリエンシー



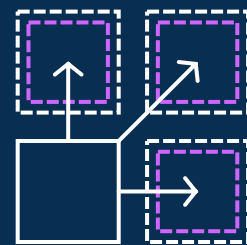
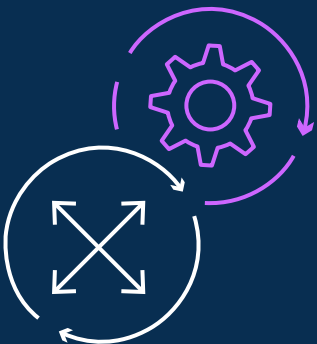
最適なTCO
高い生産性



変化に対応する
継続性

モダナイゼーションの3つの柱

様々な要素を目的合わせて取捨選択する



People



アジャイル, スクラム,
DevOps, SRE

Process



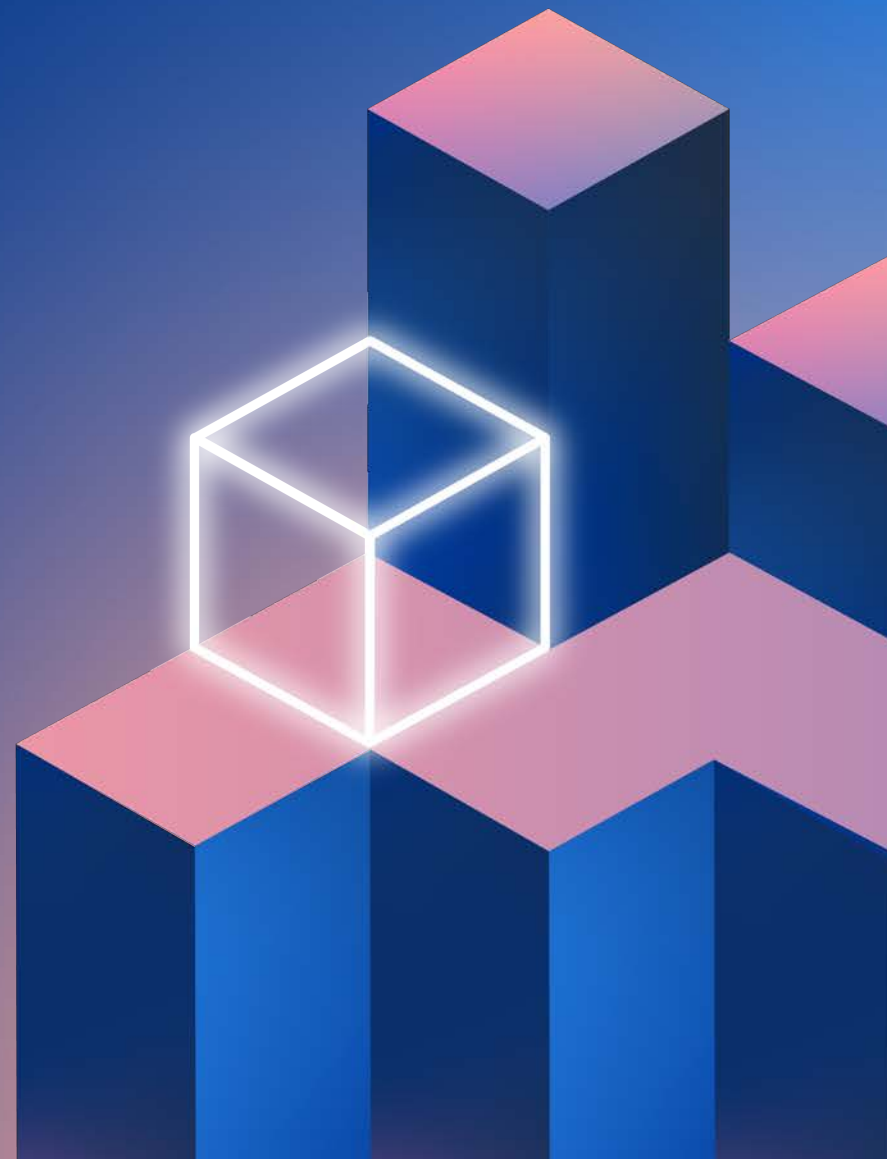
CI/CD, IaC,
オブザーバビリティ

Technology



コンテナ, サーバーレス
マイクロサービス

なぜ コンテナを利用するのか

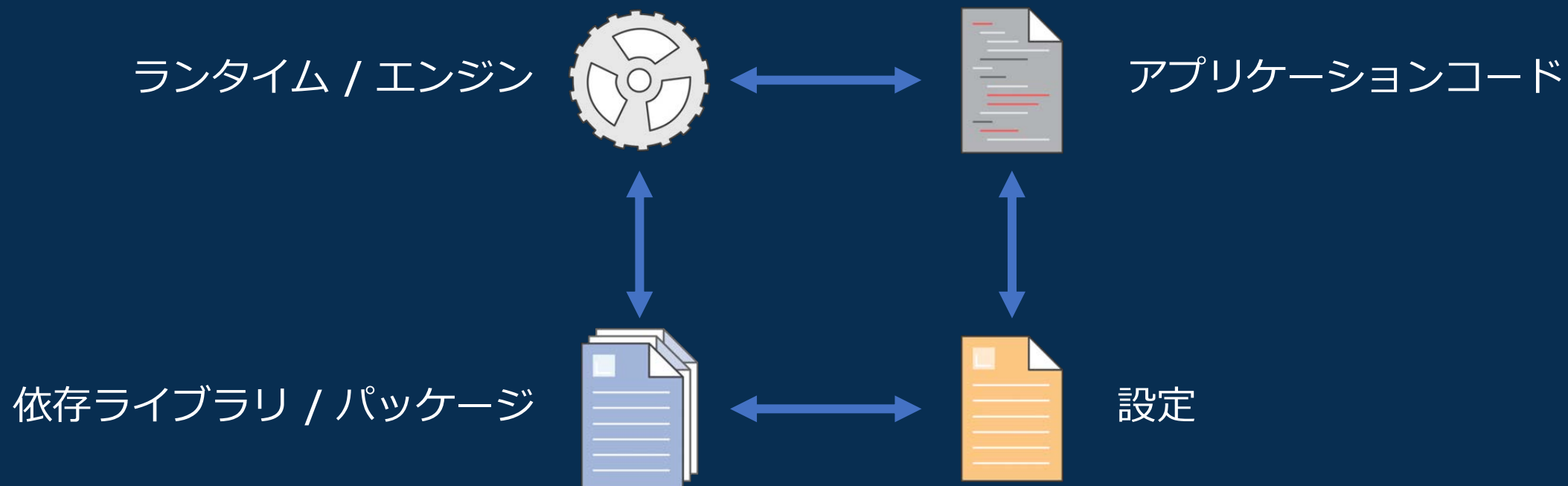


なぜコンテナ利用するのか

コンテナによる**可搬性**

コンテナランタイムさえ動いていればどこでも動く

アプリケーションを構成するコンポーネント



異なる複数の環境



ローカルラップトップ



ステージング / QA



本番

異なる複数の環境

ランタイム/エンジン



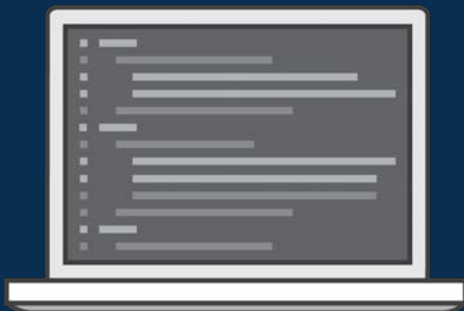
v6.0.0



v7.0.0



v4.0.0



ローカルラップトップ

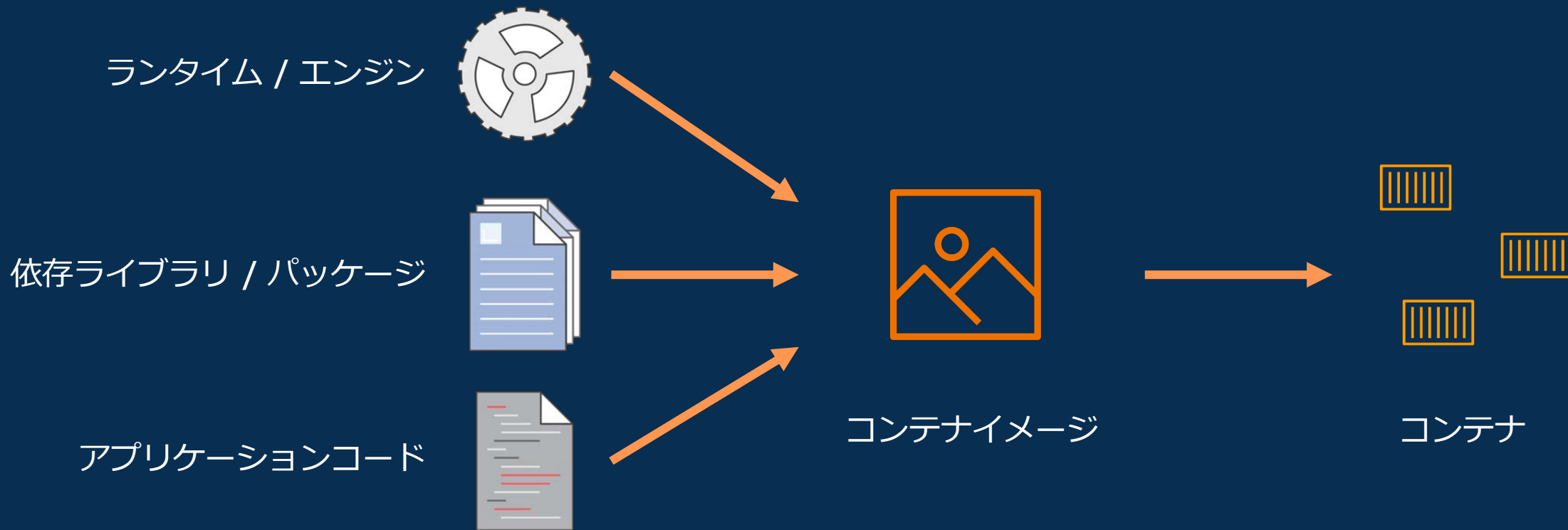


ステージング / QA

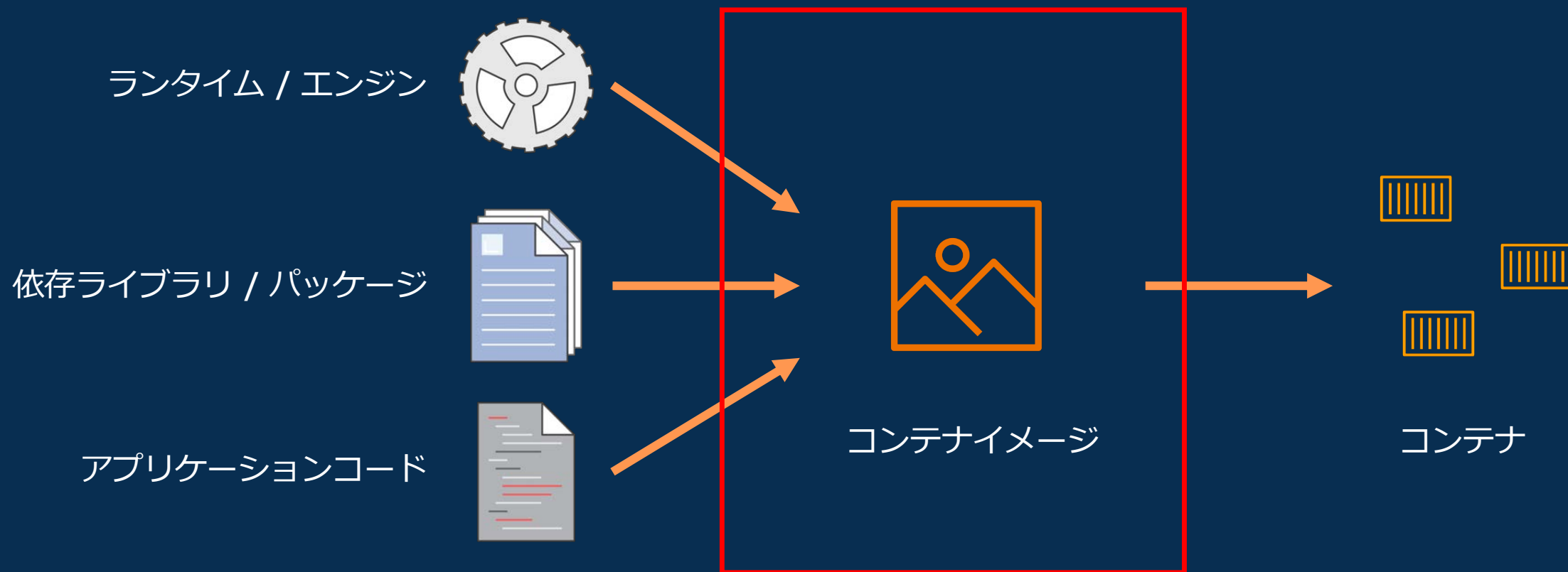


本番

「コンテナ」という解決策



「コンテナ」という解決策



コンテナイメージの管理方法

Dockerfile

```
FROM node:16.20.0-alpine3.17

RUN apk update && apk upgrade

WORKDIR /usr/src/app

COPY package.json ./

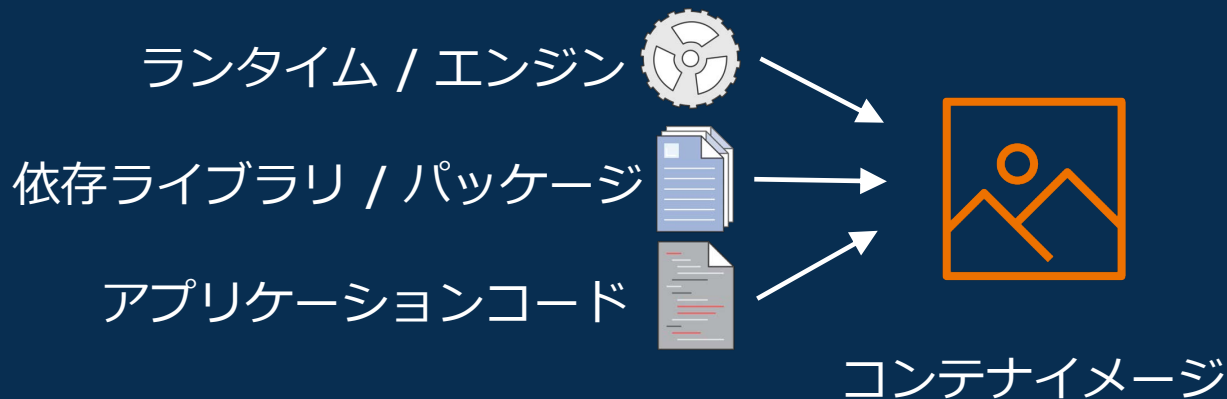
RUN npm install

COPY --chown=node:node . .

EXPOSE 8080

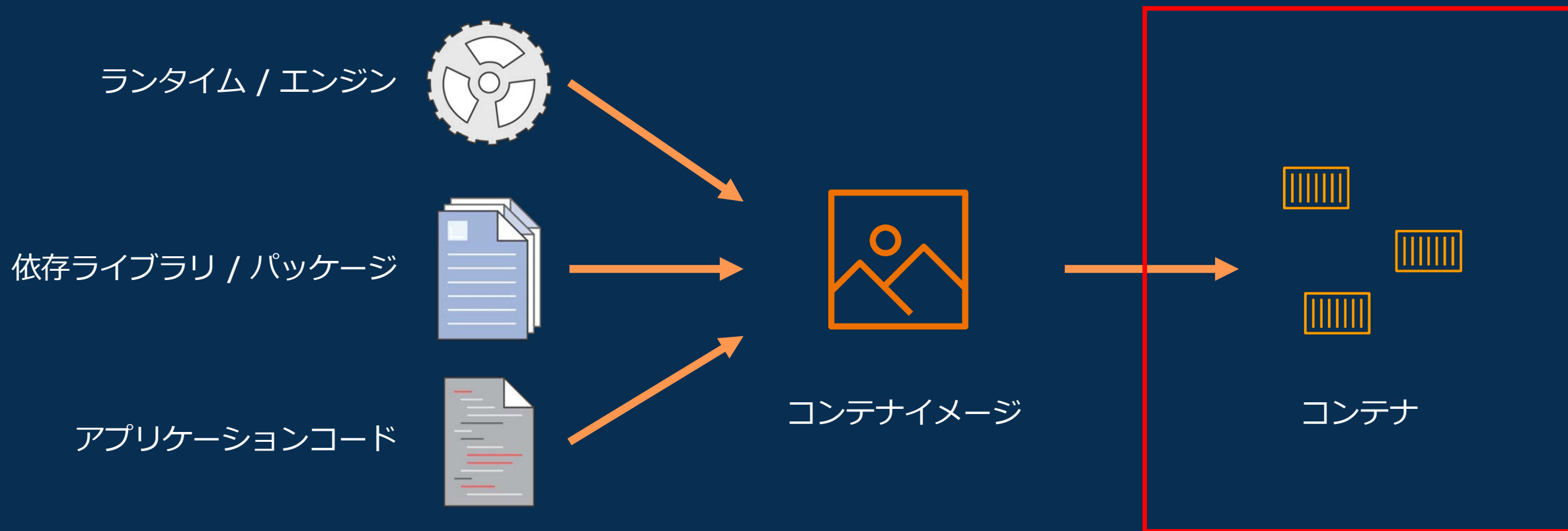
USER node

CMD [ "npm", "start" ]
```

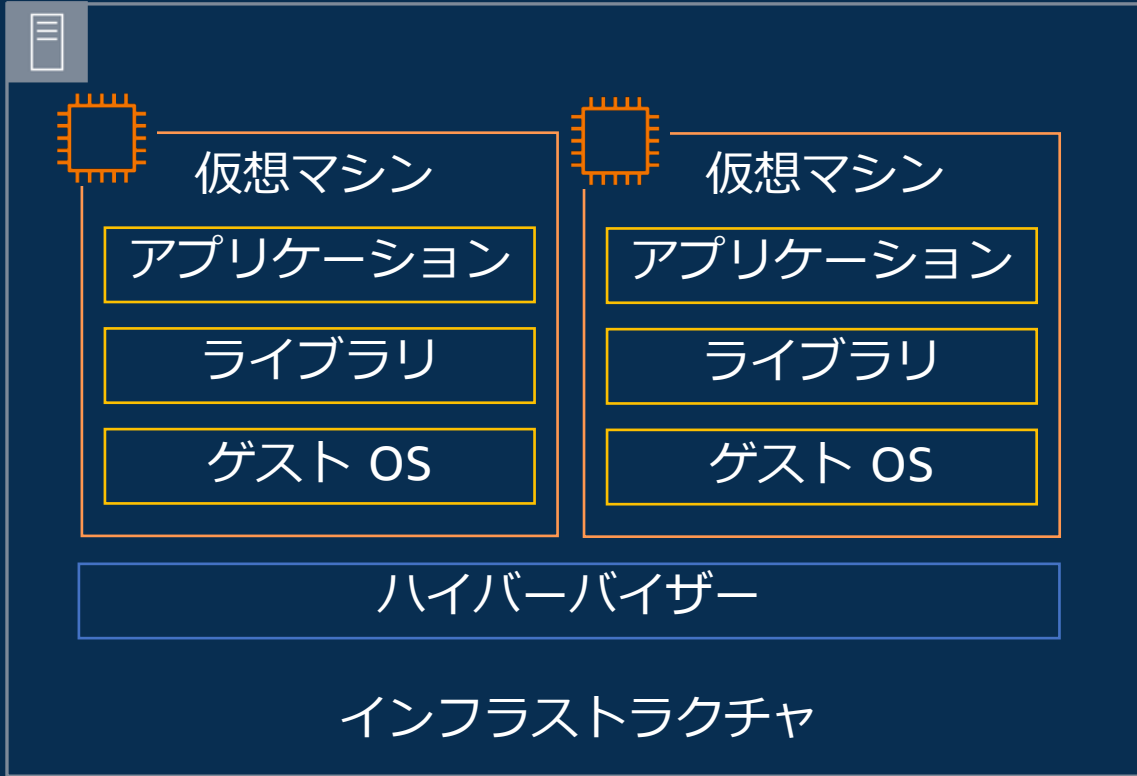


- Dockerfile という形でコンテナイメージの設計書をコードで記述可能
- 再現性のあるビルドが可能になる

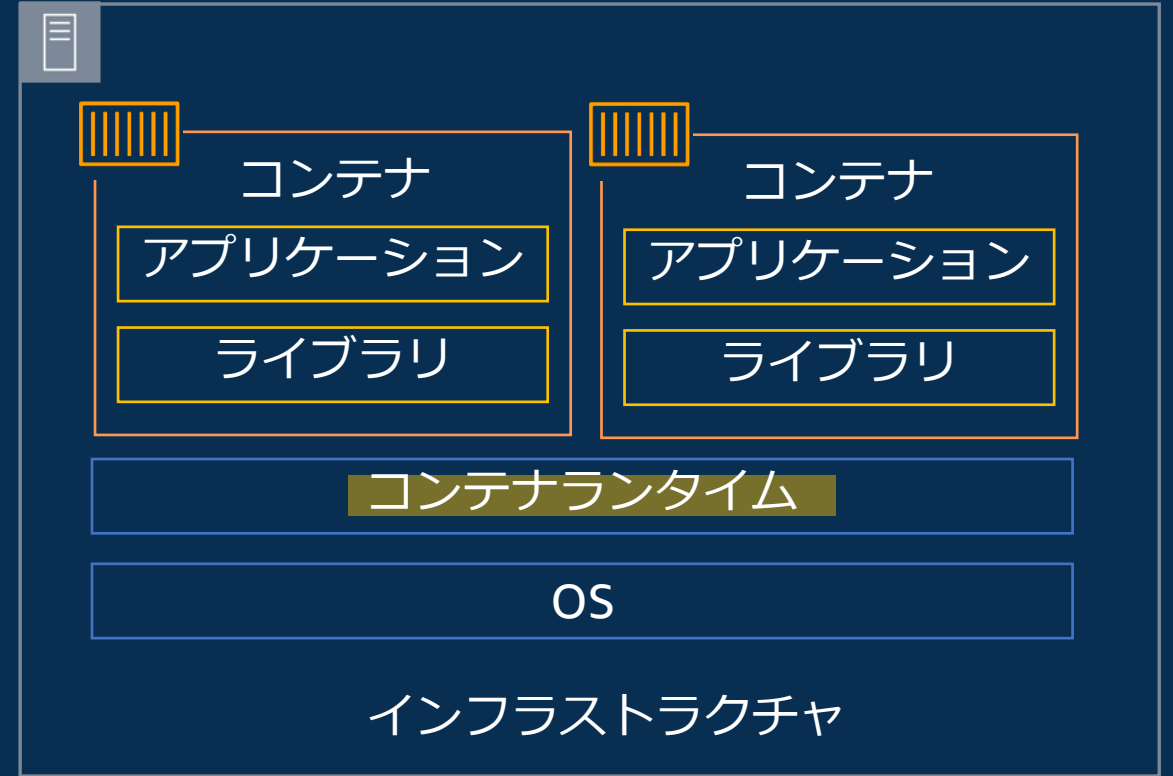
「コンテナ」という解決策



仮想マシンとコンテナ

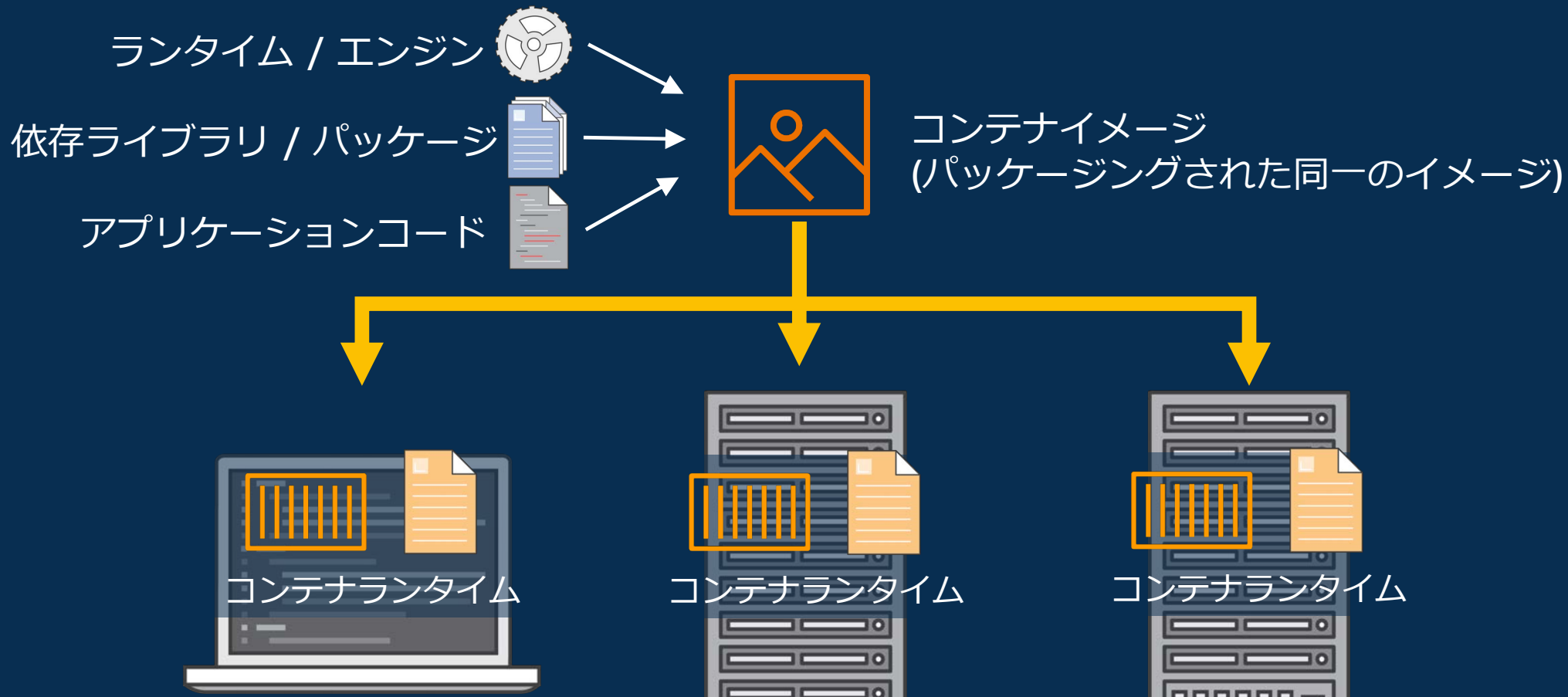


仮想マシンは
仮想マシンごとにゲストOSが必要



コンテナはOSから見ると1つのプロセス
コンテナランタイムさえあれば動く

コンテナと設定をデプロイする

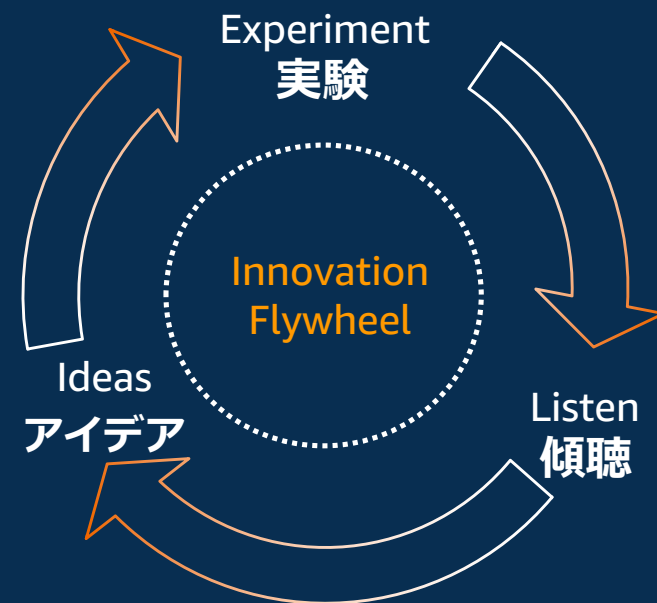


ランタイム/エンジン、ライブラリ、アプリケーションコードに差異がないため
デリバリにおける手戻りが少なくなり、素早く価値を届けることができる

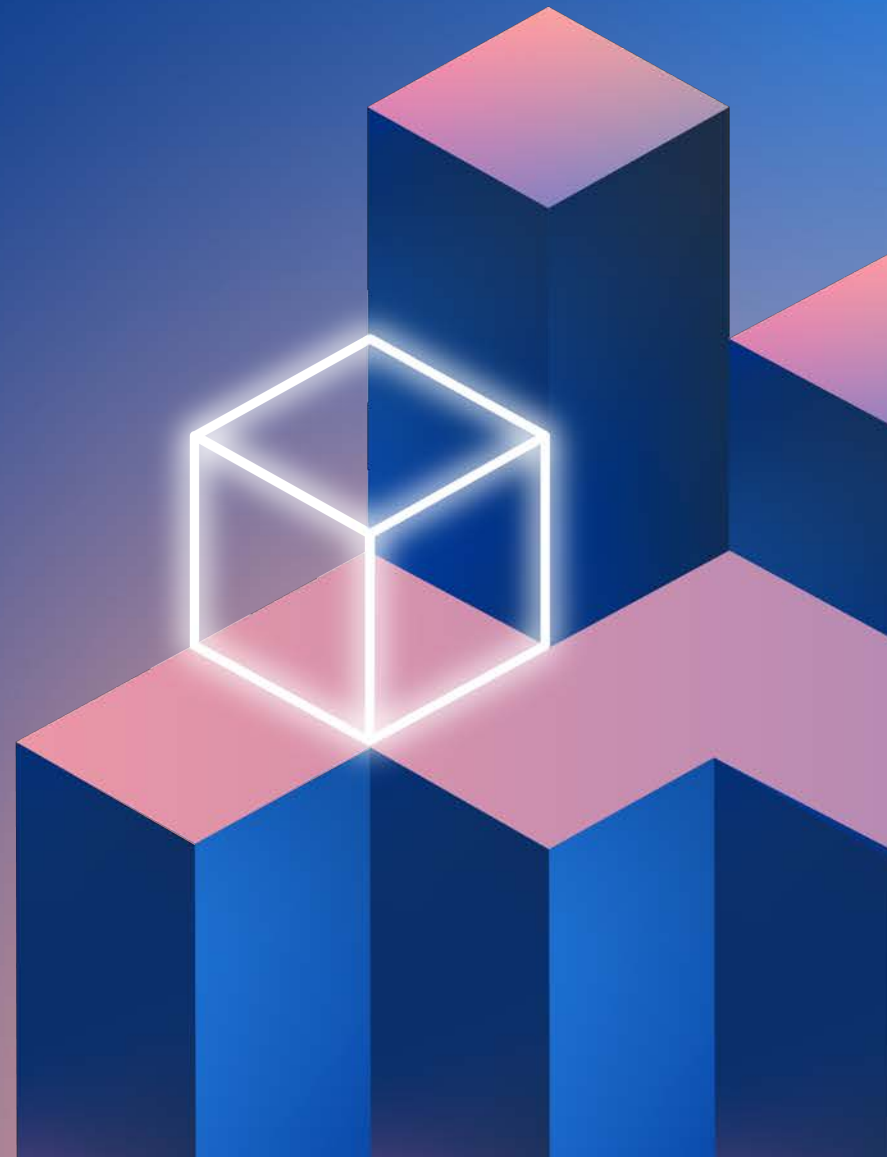
モダナイゼーションにむけて

コンテナ利用 ≠ モダナイゼーション

コンテナの可搬性の高さを活かし、
変化に迅速に対応し、
イノベーションを促進することが重要



AWS コンテナサービス紹介



紹介するコンテナサービス

オーケストレーション



Amazon ECS



Amazon EKS

レジストリ



Amazon ECR

コンピューティング



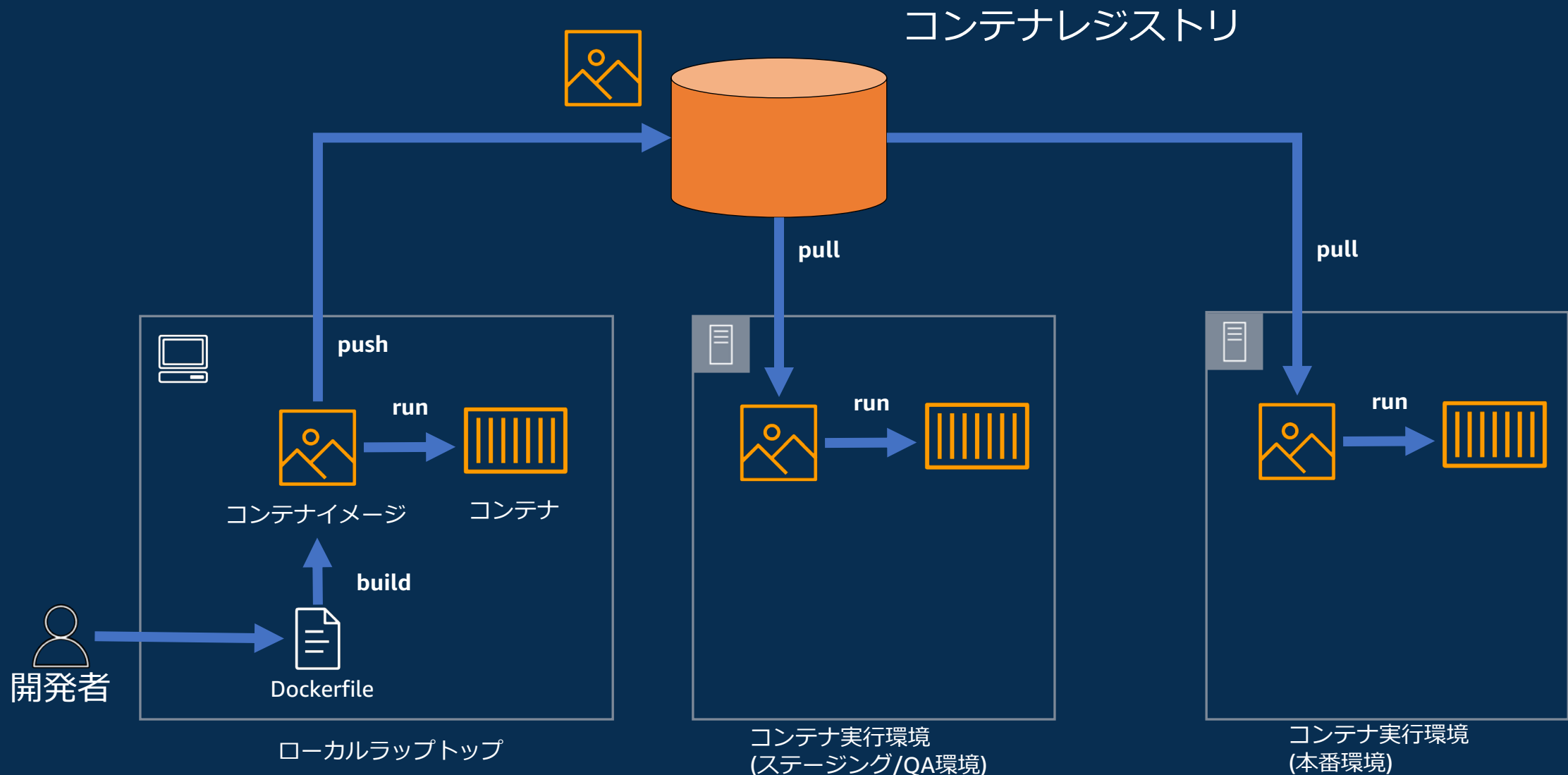
AWS Fargate

フルマネージド

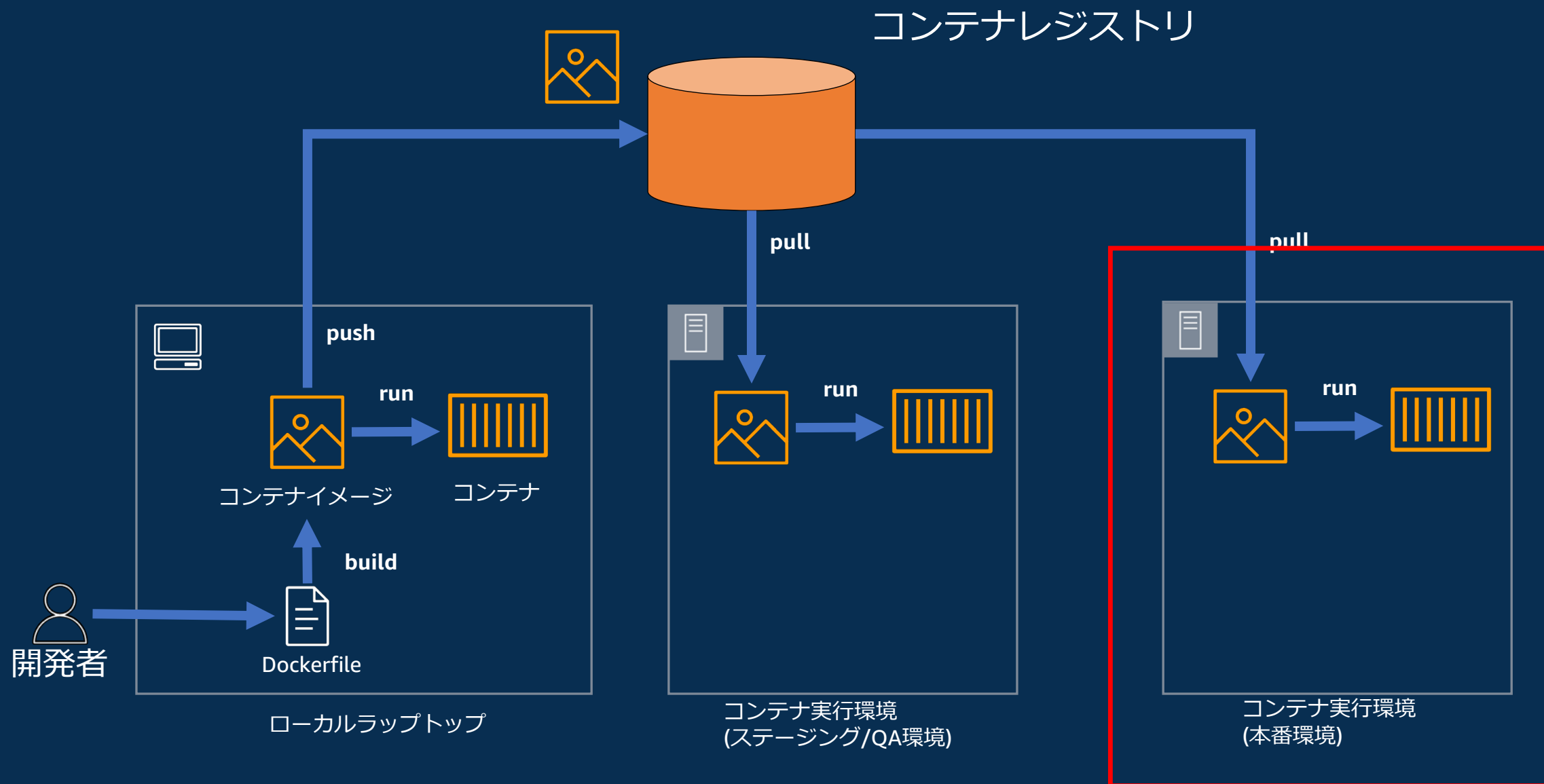


AWS App Runner

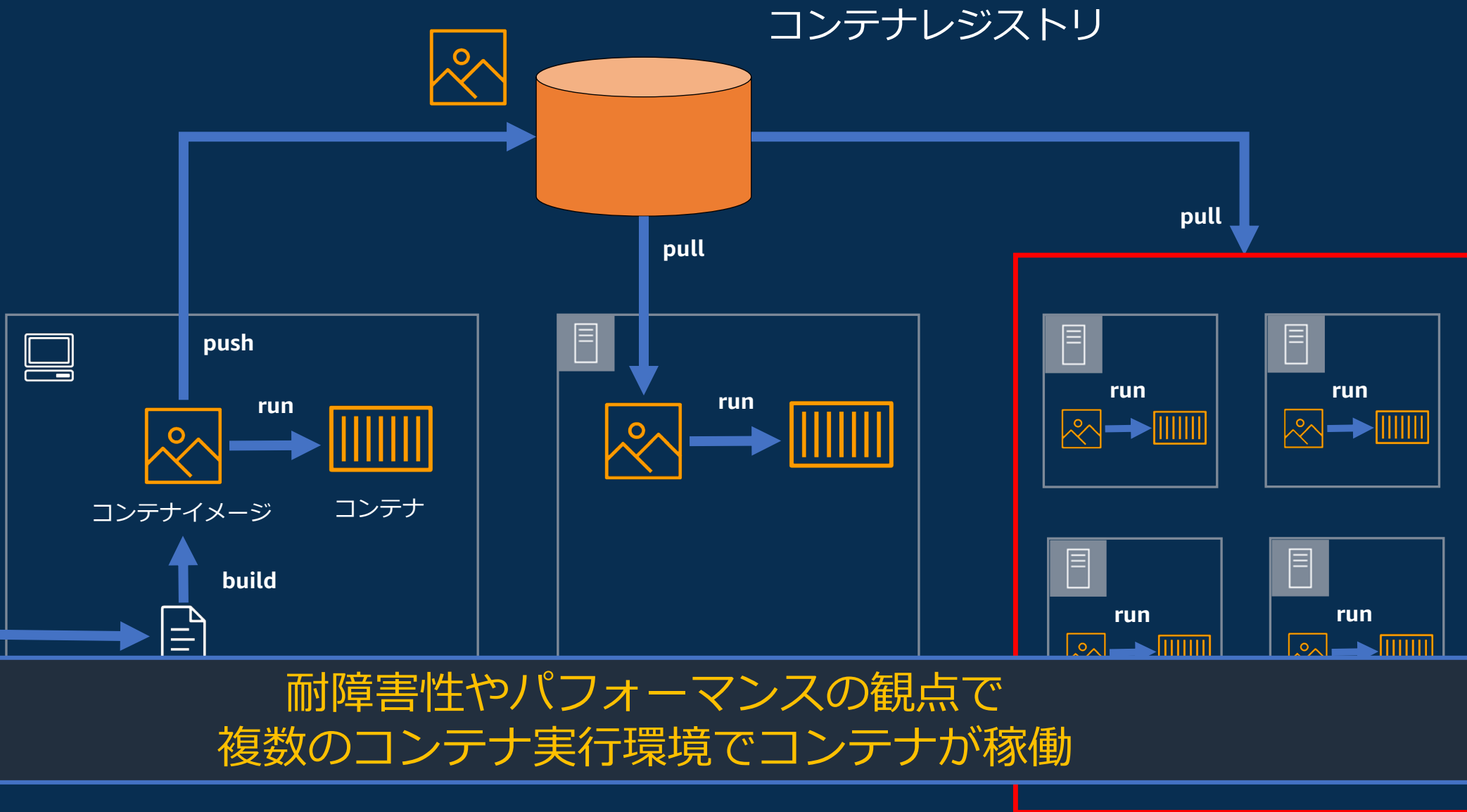
コンテナ実行の基本的なフロー



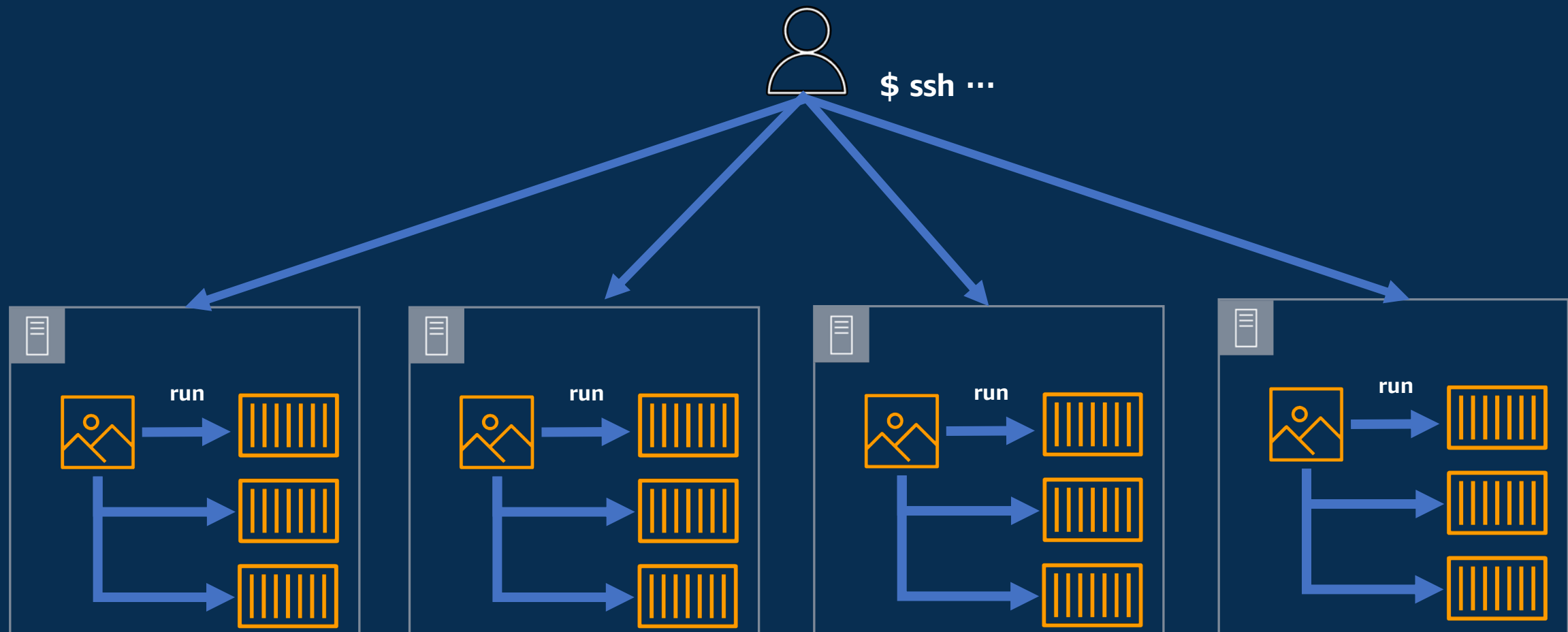
コンテナ実行の基本的なフロー



コンテナ実行の基本的なフロー

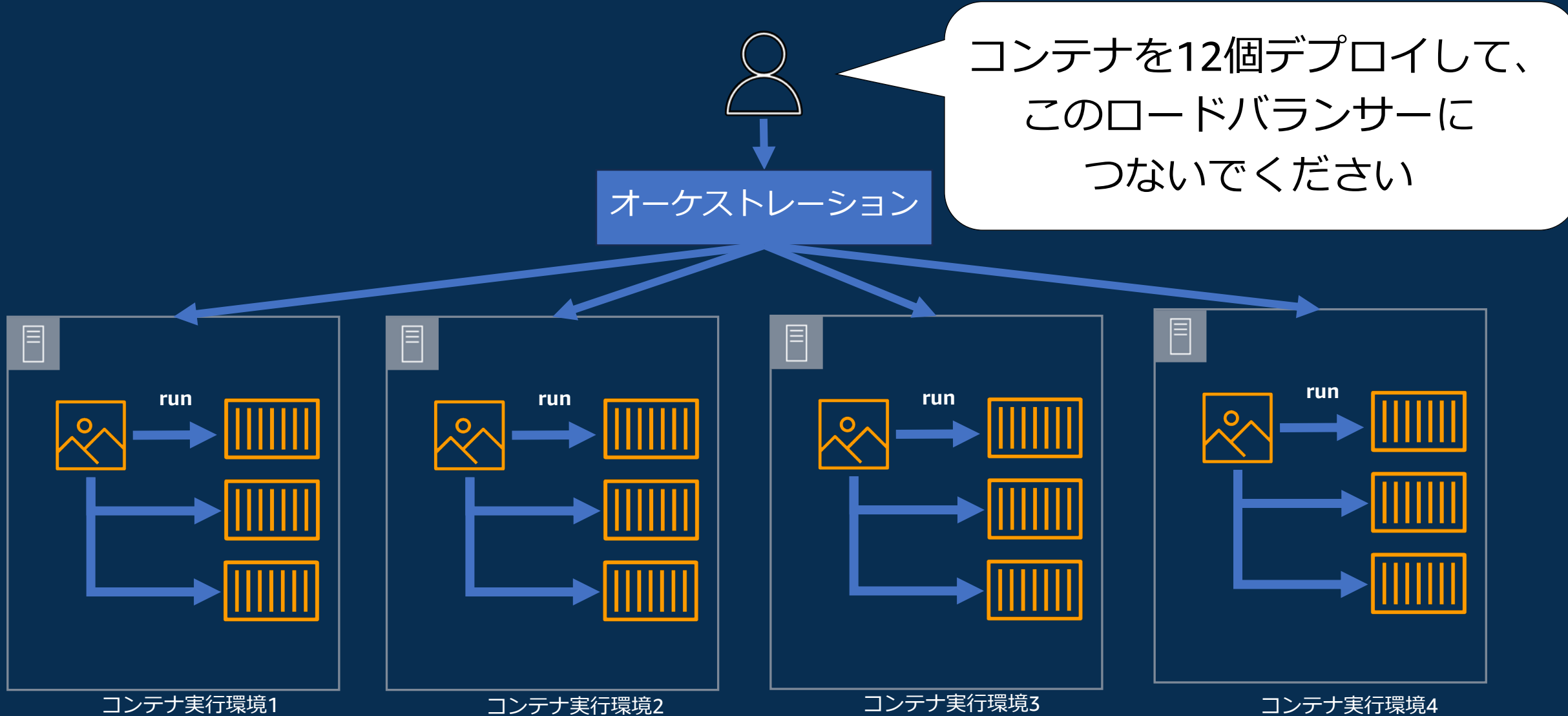


コンテナ実行の基本的なフロー



手作業でのコンテナイメージダウンロードと実行は
非効率かつミスオペレーションを招く

オーケストレーションの利用



Amazon ECS / Amazon EKS

コンテナオーケストレーションサービス



Amazon Elastic Container Service
(Amazon ECS)

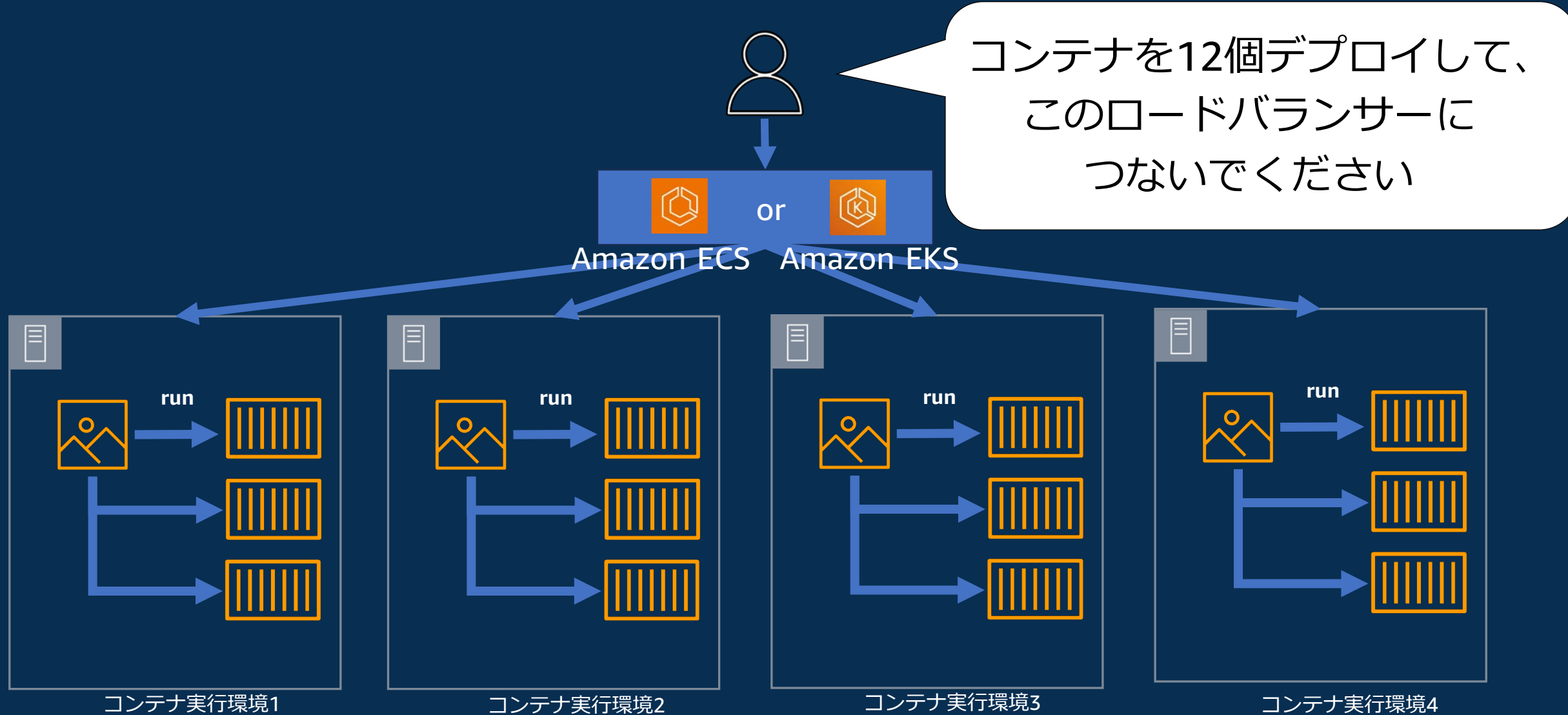
Powerful simplicity



Amazon Elastic Kubernetes Service
(Amazon EKS)

Open flexibility

オーケストレーションの利用



紹介するコンテナサービス

✓ オークストレーション

コンテナのデプロイ、スケジューリング、
スケーリング



Amazon ECS



Amazon EKS

レジストリ



Amazon ECR

コンピューティング



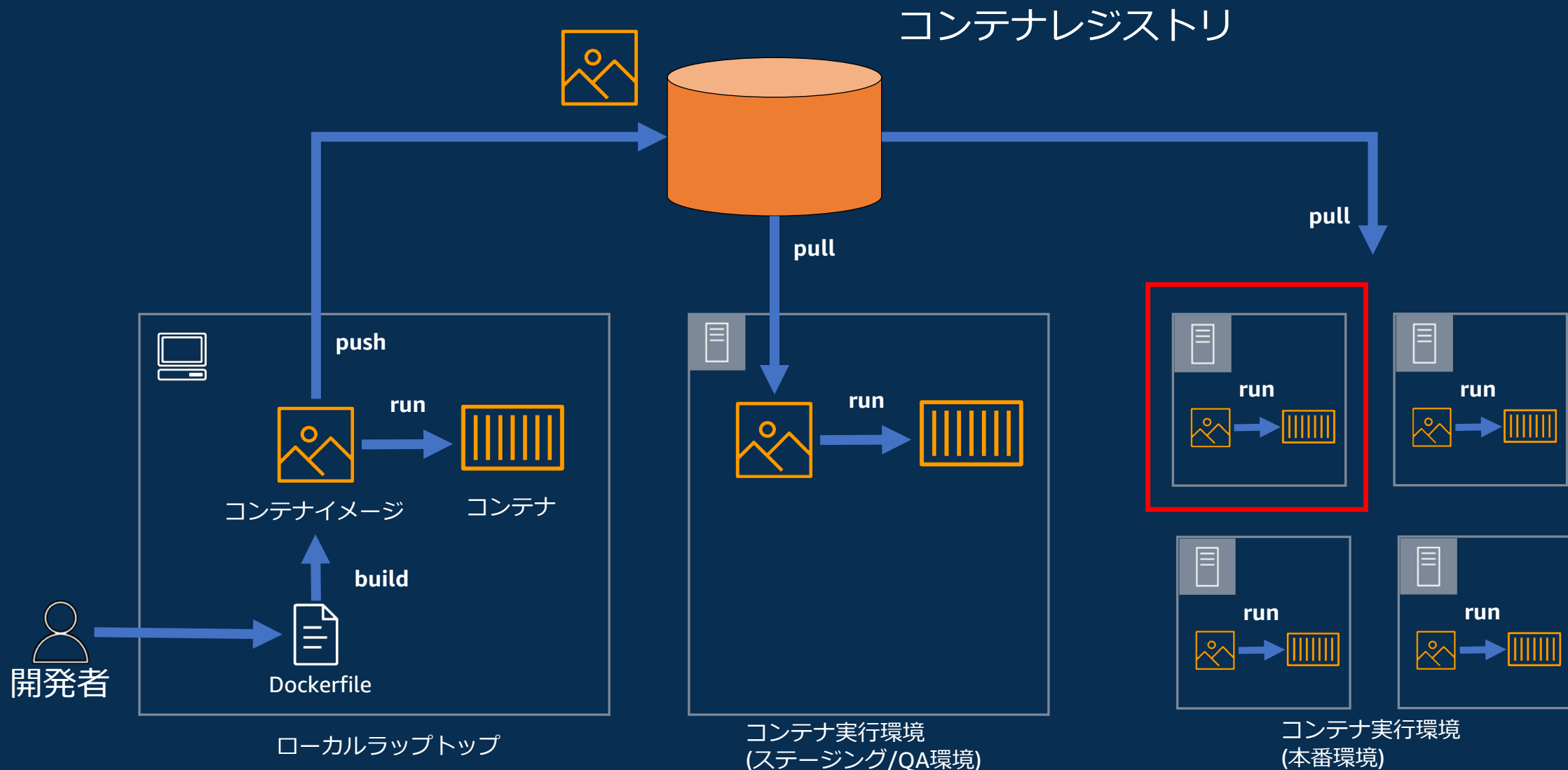
AWS Fargate

フルマネージド



AWS App Runner

コンテナ実行の基本的なフロー



コンテナ実行環境の準備

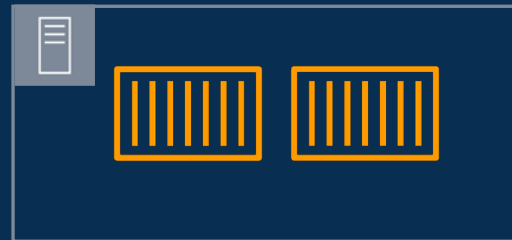
コンテナ実行環境



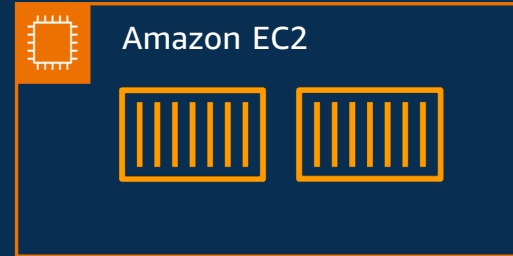
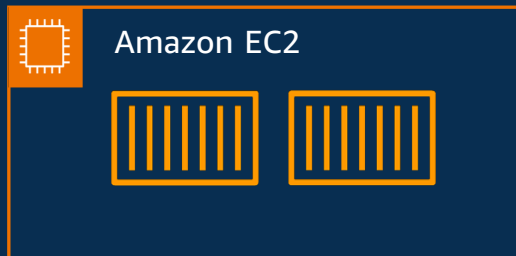
コンテナ実行環境



コンテナ実行環境



自分で用意したAmazon EC2でコンテナを実行することは可能



EC2 のサイジング
EC2 のスケーリング
OS の設定/パッチ適用

管理するレイヤーが増加すると
設計も運用も複雑化

AWS Fargate



AWS Fargate

コンテナ実行環境をサーバーレスに実現

- AWS マネージドで、EC2 インスタンスのプロビジョニング、スケール、管理などが不要
- 仮想マシンを意識しない
シームレスなスケーリングを実現
- コンテナの起動時間・使用リソースに応じた従量課金

コンテナ実行環境の管理が不要

ECS on EC2

(コンテナを仮想サーバー上で動作)



アプリケーションコンテナ

ホストのスケーリング

コンテナエージェント設定

ホスト OS / ライブラリ設定

お客様が管理するレイヤー

ECS on Fargate

(コンテナをサーバーレスで動作)



アプリケーションコンテナ

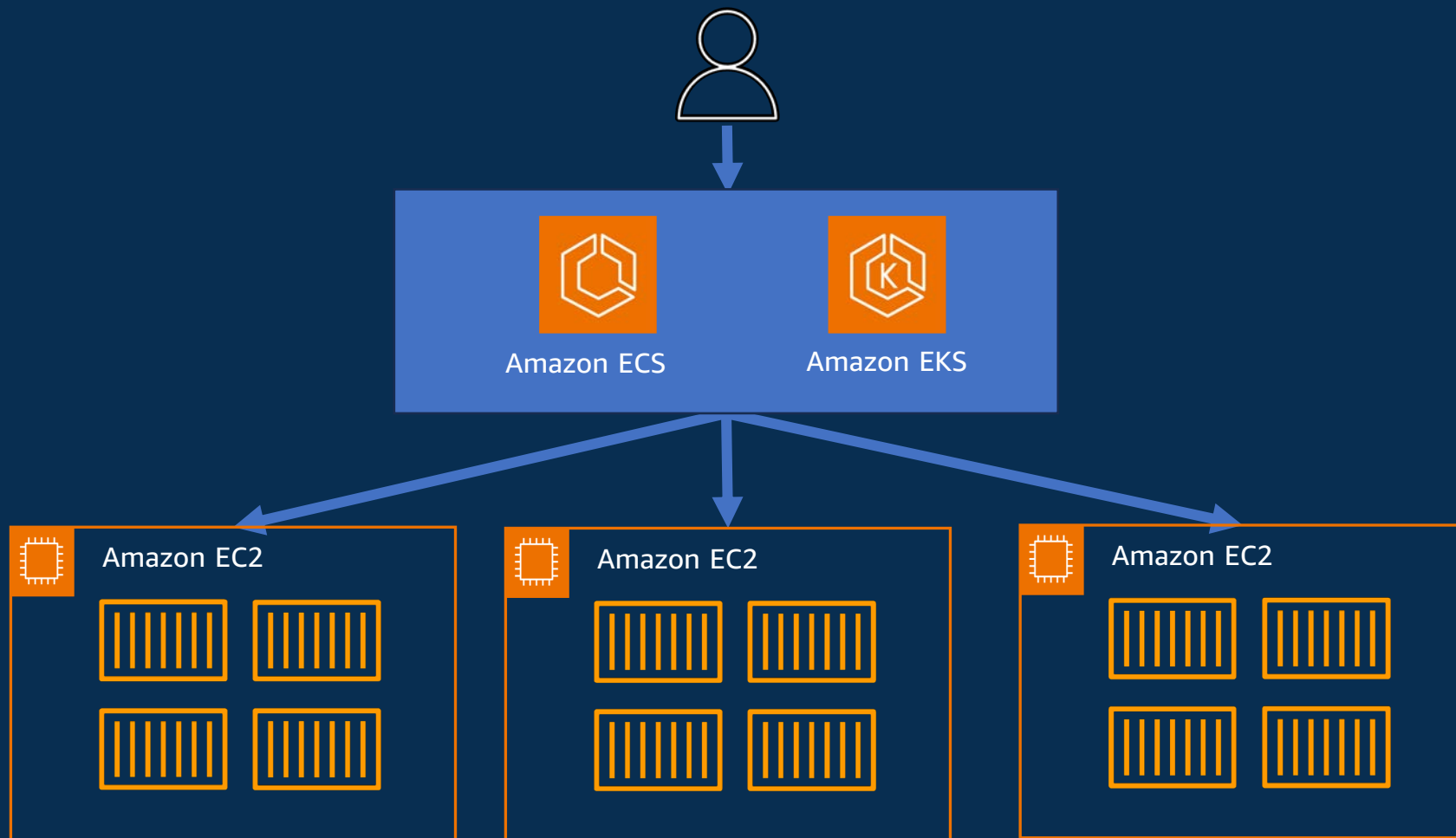
ホストのスケーリング

コンテナエージェント設定

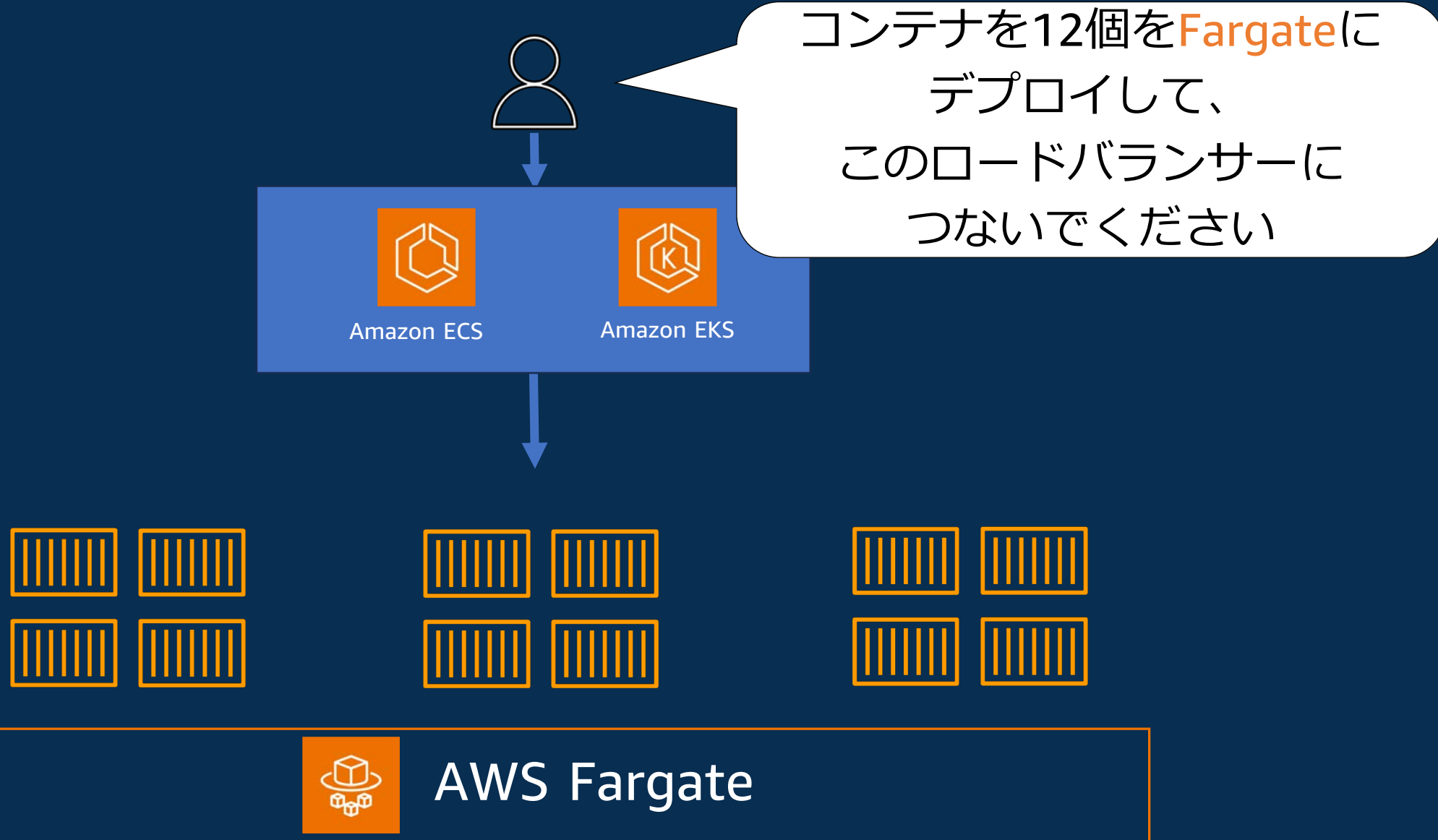
ホスト OS / ライブラリ設定

AWSが提供するレイヤー

Fargateを利用したコンテナ実行



Fargateを利用したコンテナ実行



紹介するコンテナサービス

✓ オーケストレーション

コンテナのデプロイ、スケジューリング、
スケーリング



Amazon ECS



Amazon EKS

レジストリ



Amazon ECR

✓ コンピューティング

サーバーレスなコンテナ実行環境



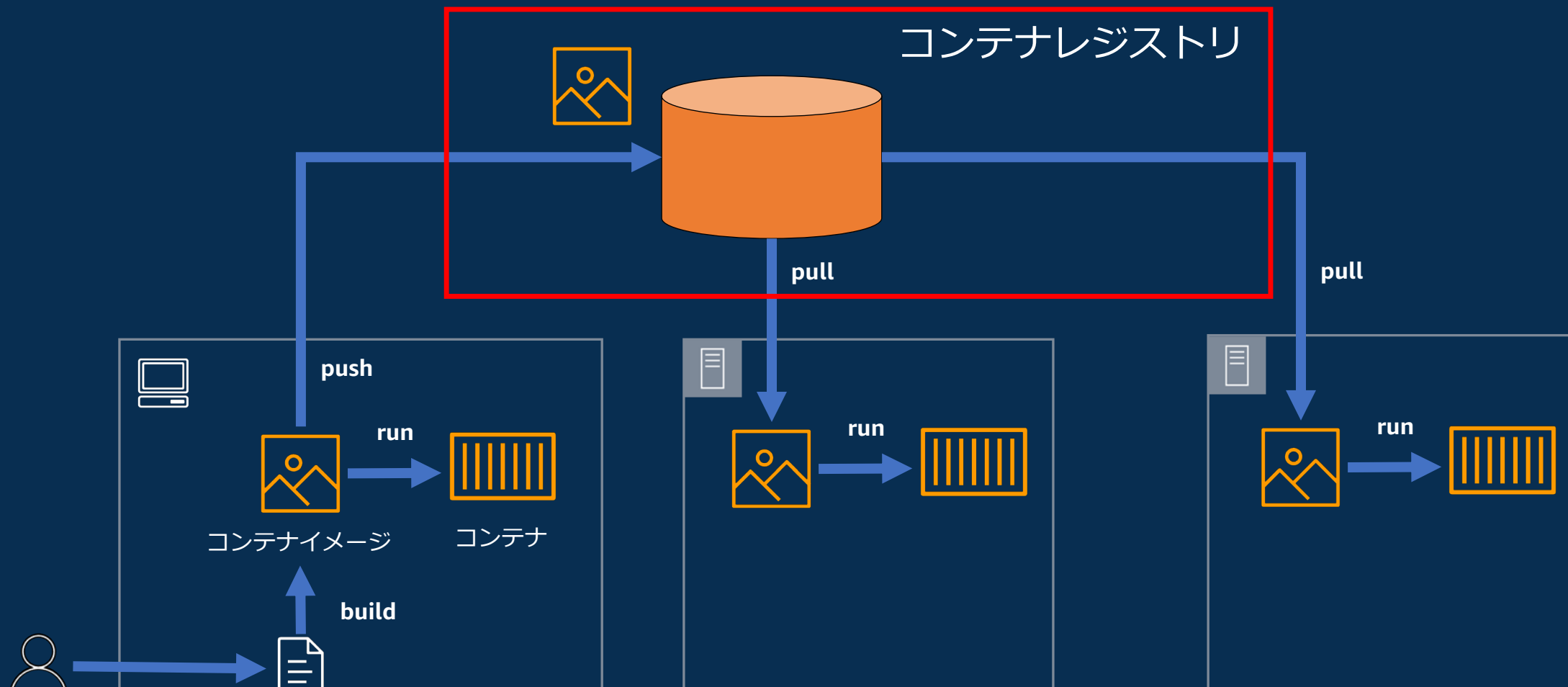
AWS Fargate

フルマネージド



AWS App Runner

コンテナ実行の基本的なフロー



セキュリティ、スケーラビリティ、信頼性が必要

Amazon ECR

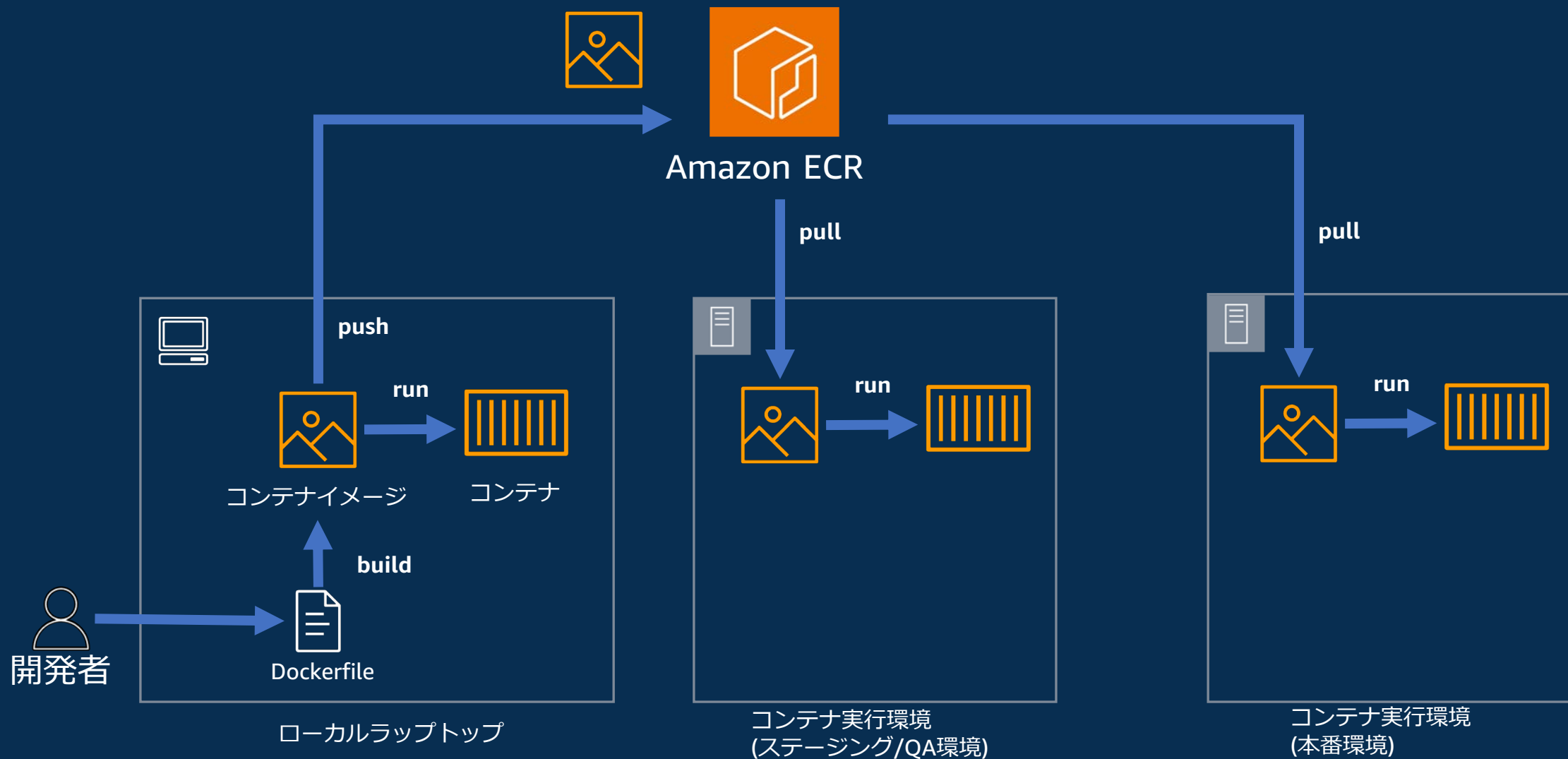


Amazon Elastic Container Registry
(Amazon ECR)

フルマネージドなコンテナイメージレジストリ

セキュア – 保管イメージの自動的な暗号化、
IAM 連携、脆弱性スキャン

Amazon ECRの利用



紹介するコンテナサービス

✓ オーケストレーション

コンテナのデプロイ、スケジューリング、
スケーリング



Amazon ECS



Amazon EKS



レジストリ

コンテナイメージの格納



Amazon ECR

✓ コンピューティング

サーバーレスなコンテナ実行環境



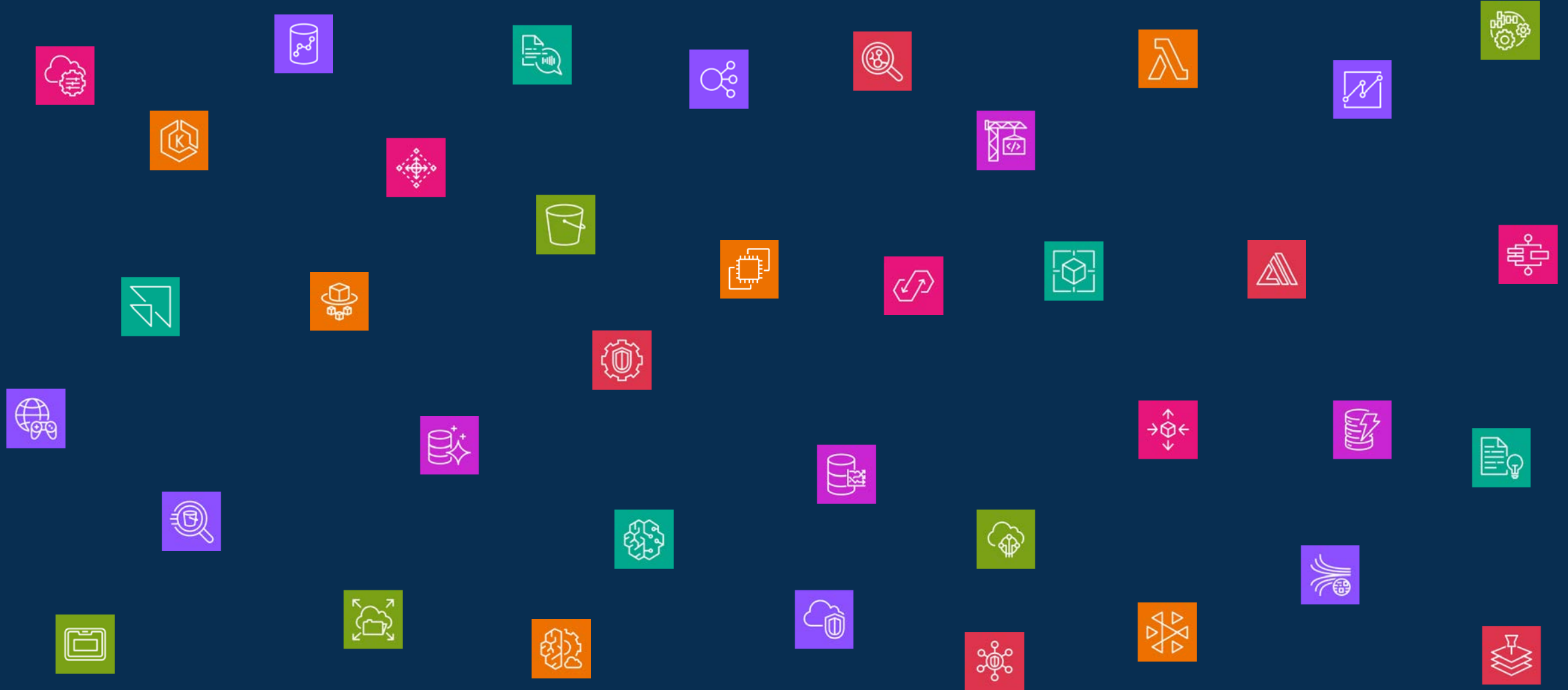
AWS Fargate

フルマネージド



AWS App Runner

AWS は幅広い選択肢を提供



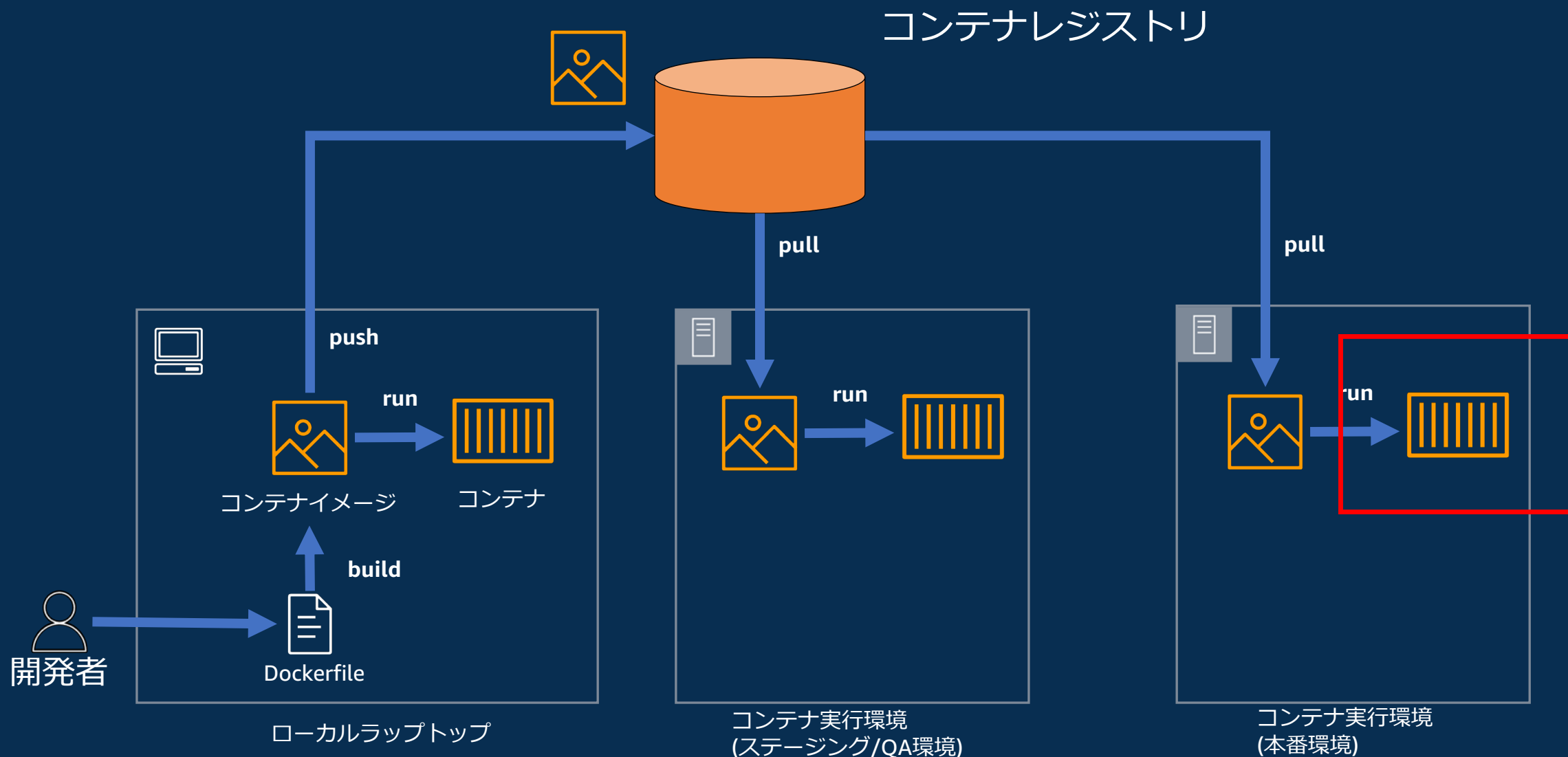
AWS は幅広い選択肢を提供

200 以上のサービス

A close-up photograph of a child's hand, wearing a white sleeve, interacting with several colorful LEGO bricks. The bricks are in shades of red, blue, yellow, green, and white. The child's fingers are positioned to connect or adjust the bricks. The background is a soft, out-of-focus light color.

ブロックのようにサービスを組み合わせ
アプリケーション構築が可能

コンテナ実行の基本的なフロー



AWS サービスを組み合わせることでインフラを構築



AWS CodeBuild
[CI/CD]



AWS Auto Scaling
[オートスケーリング]



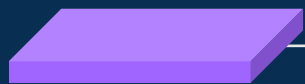
Application Load Balancer (ALB)
[ロードバランサー]



Amazon Elastic Container Service (Amazon ECS)
[オーケストレーション]



AWS Fargate
[コンピューティング]



Amazon Virtual Private Cloud (Amazon VPC)
[ネットワーク]

柔軟性が高い

手間がかかる

構築済みのインフラにすぐにデプロイ



App Runner がもたらすメリット



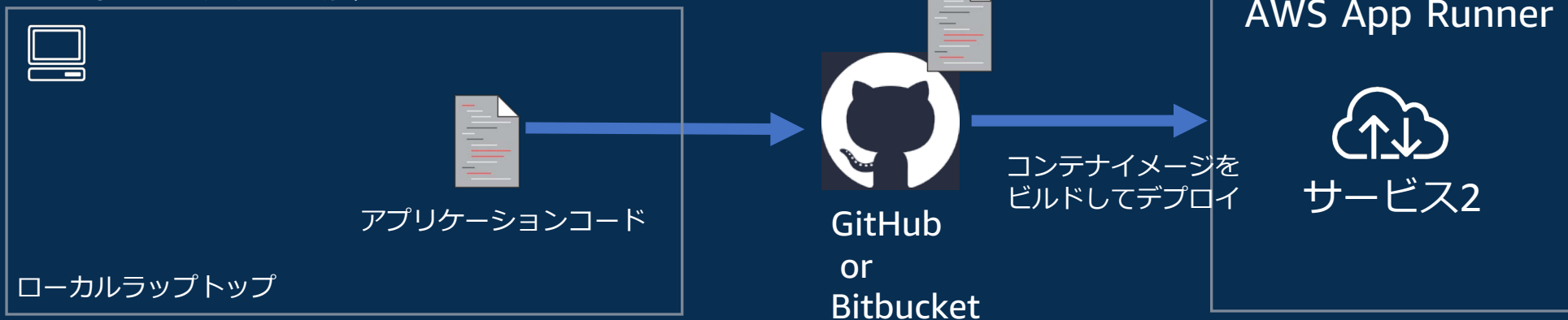
- 市場投入までの時間を短縮
 - 簡単かつ迅速に Web アプリを公開
- アプリケーション開発に集中
- アプリケーションの
実行時間に対して課金

AWS App Runner の利用方法

コンテナイメージベースの方法



コードベースの方法



← https ユーザー

紹介するコンテナサービス

✓ オーケストレーション

コンテナのデプロイ、スケジューリング、
スケーリング



Amazon ECS



Amazon EKS



レジストリ

コンテナイメージの格納



Amazon ECR

✓ コンピューティング

サーバーレスなコンテナ実行環境



AWS Fargate



フルマネージド



AWS App Runner

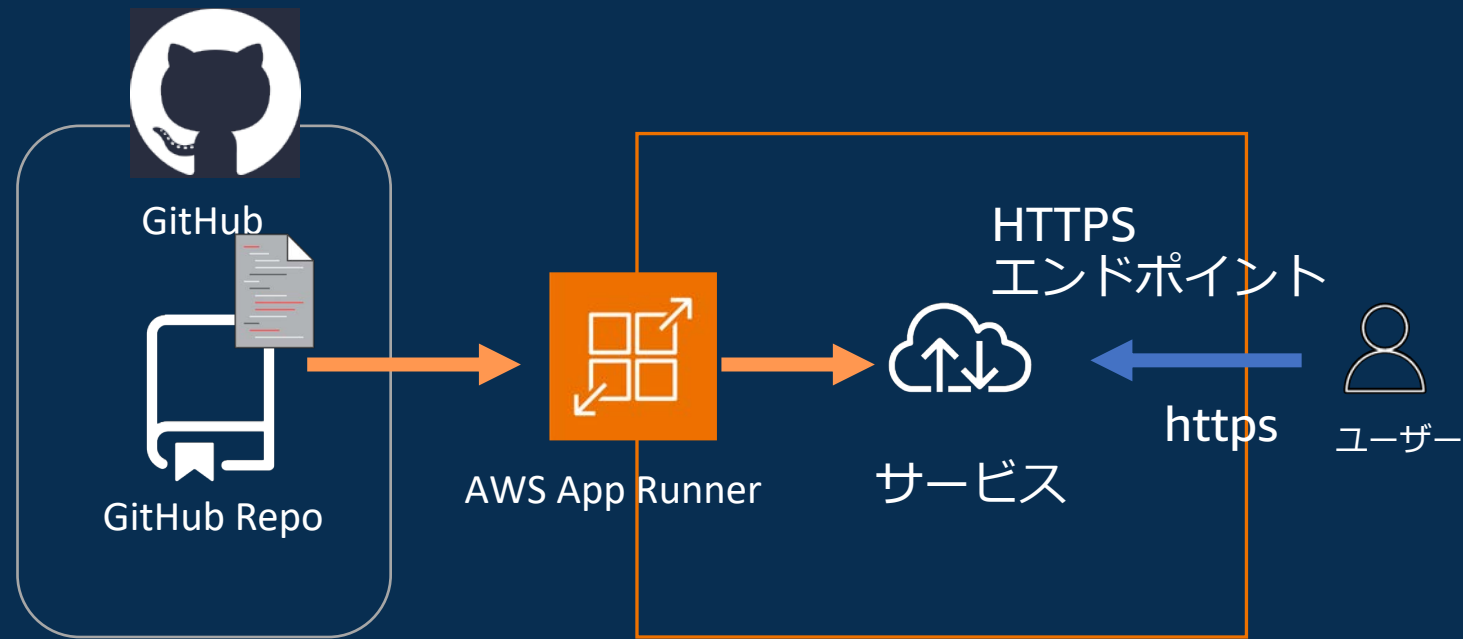
コンテナサービスの始め方



デモ1 – App Runner で WEB アプリ(コンテナ)を公開

デモ1の流れ

1. ソースコード確認(Github)
2. App Runner 上にサービス作成
 1. App Runner がソースコード取得
 2. App Runner がコンテナをビルド & デプロイ
3. WEB アプリ動作確認

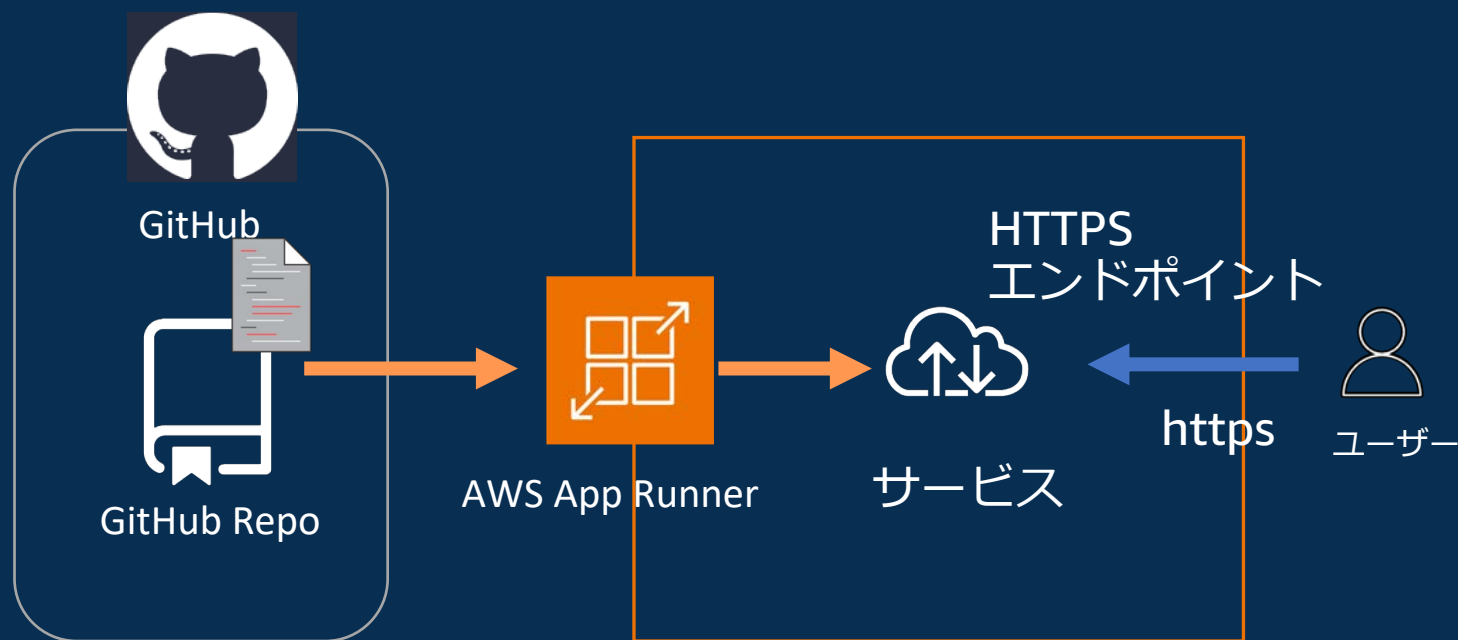


デモ 1 実施

デモ2 – ソースコードの変更をトリガーに自動デプロイ

デモ2の流れ

1. App Runner トリガー設定変更 (手動->自動)
2. ソースコード変更(Github)
3. App Runner がサービスを自動デプロイ
4. WEB アプリ動作確認(変更確認)



デモ 2 実施

まとめ



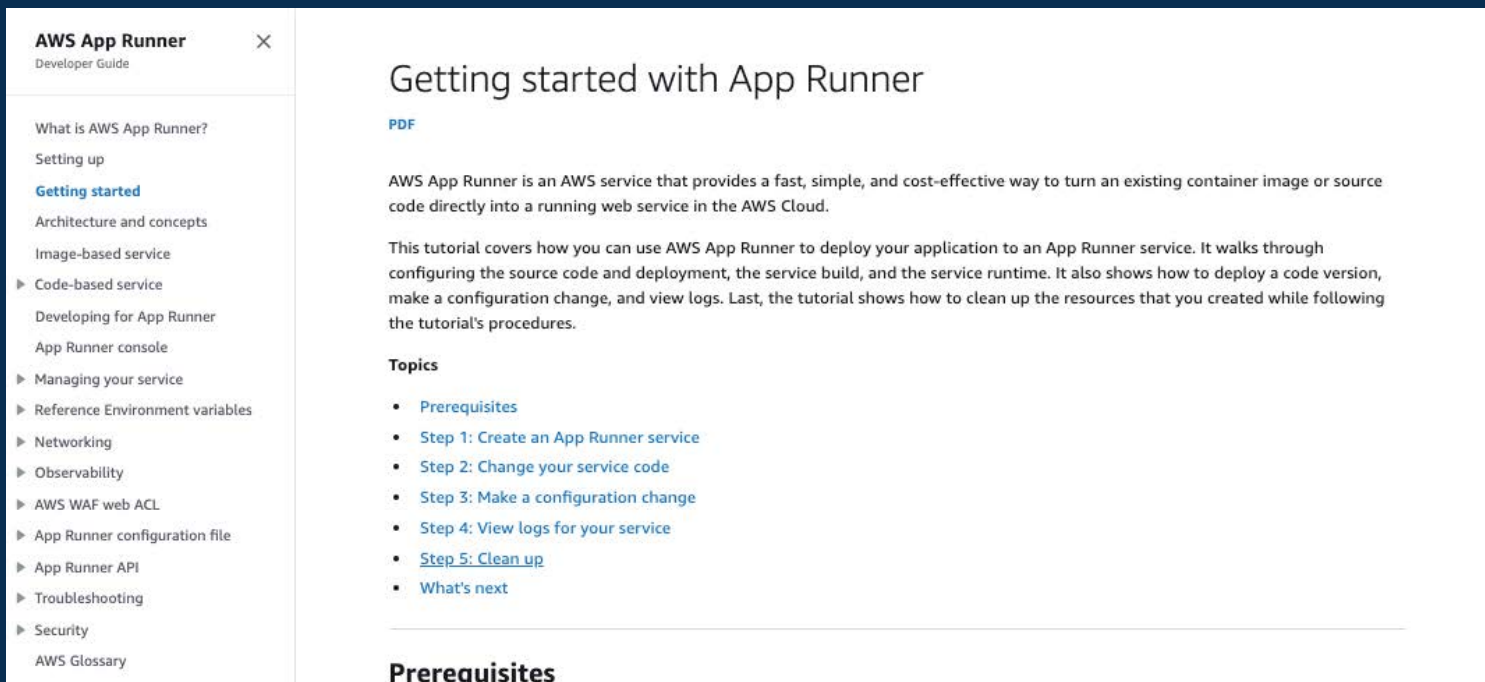
まとめ

- なぜモダナイゼーションが必要か
 - 変化に迅速に対応し、イノベーションを促進するため
- なぜコンテナを利用するのか
 - アプリケーションに必要なものをパッケージングしているため、可搬性がある
 - 可搬性を活かし、デリバリ速度を高める
- コンテナサービス紹介
- コンテナサービスの始め方
 - フルマネージドの App Runner の使い方を紹介



今後に向けて

1st ステップ : AWS App Runner のチュートリアルをやってみよう



Getting started with App Runner

<https://docs.aws.amazon.com/apprunner/latest/dg/getting-started.html>

(英語版)

今後に向けて

実践に向けて：ハンズオン - AWS App Runner ワークショップ

AWS App Runner ワークショップ

はじめに

- ワークショップのセットアップ
 - AWS 主催のイベントに参加している場合
 - 自身の AWS アカウントを使用する場合
 - Cloud9 環境への接続
- インフラストラクチャの概要
 - 事前作成済みのリソース
 - ラボで作成するリソース
- AWS App Runner サービスの作成
 - イメージベースのデプロイ
 - ソースコードベースのデプロイ (GitHub)

AWS App Runner ワークショップ

AWS App Runner ワークショップへようこそ！

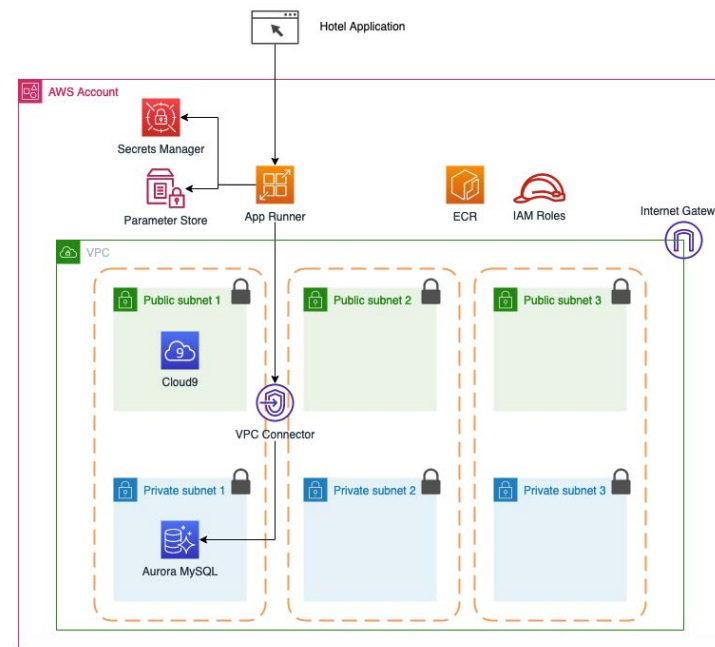
AWS App Runner は、インフラストラクチャを管理することなく、AWS 上にアプリケーションをデプロイするための高速、シンプルかつセキュアな方法を提供します。利用者がソースコードまたはコンテナイメージを提供すると、App Runner はコンテナ化されたウェブアプリケーションを自動的にビルド・デプロイし、トラフィックの暗号化や負荷分散、トラフィックの需要に合わせたスケーリングを実現します。プライベートなサービスの作成や、プライベートな VPC 内のサービスへの接続も簡単に設定できます。また、**AWS X-Ray** や **AWS WAF** などの AWS サービスとも統合されています。

このワークショップでは、App Runner の機能や他の AWS サービスとの統合を体験していきます。それでは、早速ワークショップを始めましょう！

[Previous](#) [Next](#)

データベースへの接続

データベース接続のラボでは、Aurora データベースへのネットワーク接続を実現するために VPC コネクタを作成します。また、Secrets Manager に保存されたデータベースの認証情報と Systems Manager パラメータストアに保存されたパラメータを、アプリケーション起動時に環境変数として取得します。





Thank you!



AWS TRAINING & CERTIFICATION

600+ ある AWS Skill Builder の無料デジタルコースで学ぼう

30 以上の AWS ソリューションの中から、自分にもっとも関係のあるクラウドスキルとサービスにフォーカスし、自習用のデジタル学習プランとランプアップガイドで学ぶことができます。

自分に合ったスキルアップ方法で学ぼう

EXPLORE.SKILLBUILDER.AWS »



あなたのクラウドスキルを AWS 認定で証明しよう

業界で認められた資格を取得して、スキルアップの一步を踏み出しましょう。AWS Certified の取得方法と、準備に役立つ AWS のリソースをご覧ください。

受験準備のためのリソースにアクセスしよう »



AWS Builders Online Series にご参加いただきありがとうございます

楽しんでいただけましたか? ぜひアンケートにご協力ください。
本日のイベントに関するご意見/ご感想や今後のイベントについてのご希望や改善のご提案などがございましたら、ぜひお聞かせください。



aws-apj-marketing@amazon.com



twitter.com/awscloud_jp



facebook.com/600986860012140



<https://www.youtube.com/user/AmazonWebServicesJP>



<https://www.linkedin.com/showcase/aws-careers/>



twitch.tv/aws