

# はじめてのコンテナワークロード ～ コンテナと AWS のコンテナサービスが解決する 課題とは ～

堀内 保大

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 インターネットメディアソリューション部  
ソリューションアーキテクト

# 自己紹介

堀内 保大 (Yasuhiro Horiuchi)  @ka\_shino\_ki

AWS Japan, Solutions Architect

主に Web 系のお客様の支援

前職：Sler で性能関連の技術支援



## 好きなAWSサービス

Amazon EKS 

AWS Fargate 

Amazon Managed Service for Prometheus (AMP) 



# 本セッションの想定聴講者とゴール

## 本セッションの対象となる方

- コンテナに興味がある開発者
- コンテナという言葉を知っているが、仕組みやメリットを知らない
- AWS のコンテナサービスがどういうときに使われるものなのかわからない

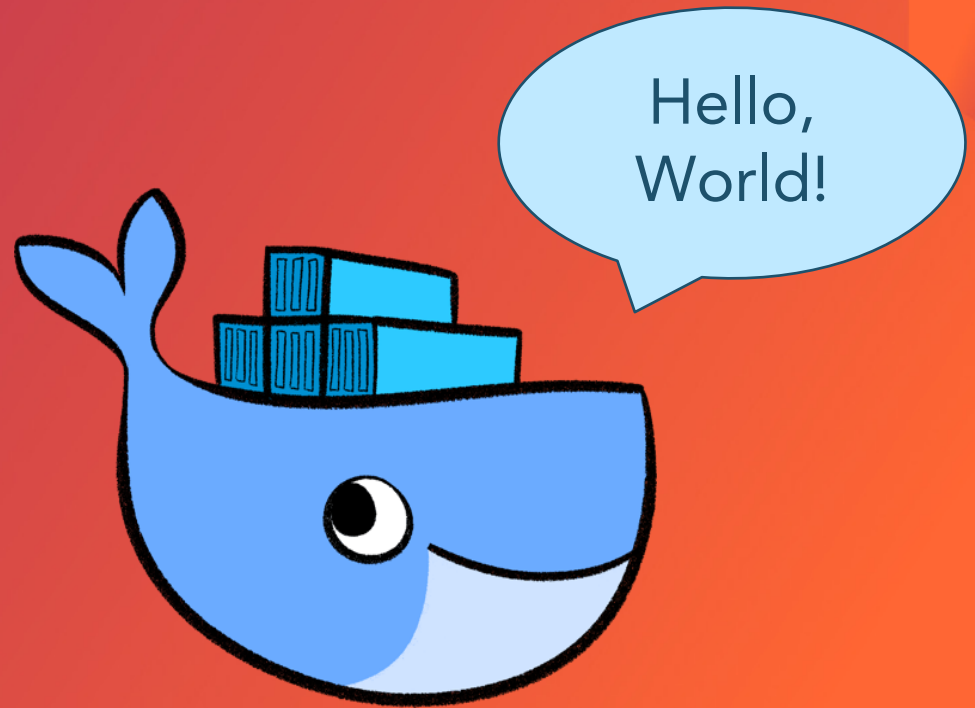
## 本セッションのゴール

- コンテナが解決できる課題を理解する
- AWS のコンテナサービスが解決する課題を理解する
- 自身のワークロードをコンテナで動かす際に必要な AWS サービスを選択できるようになる

# アジェンダ

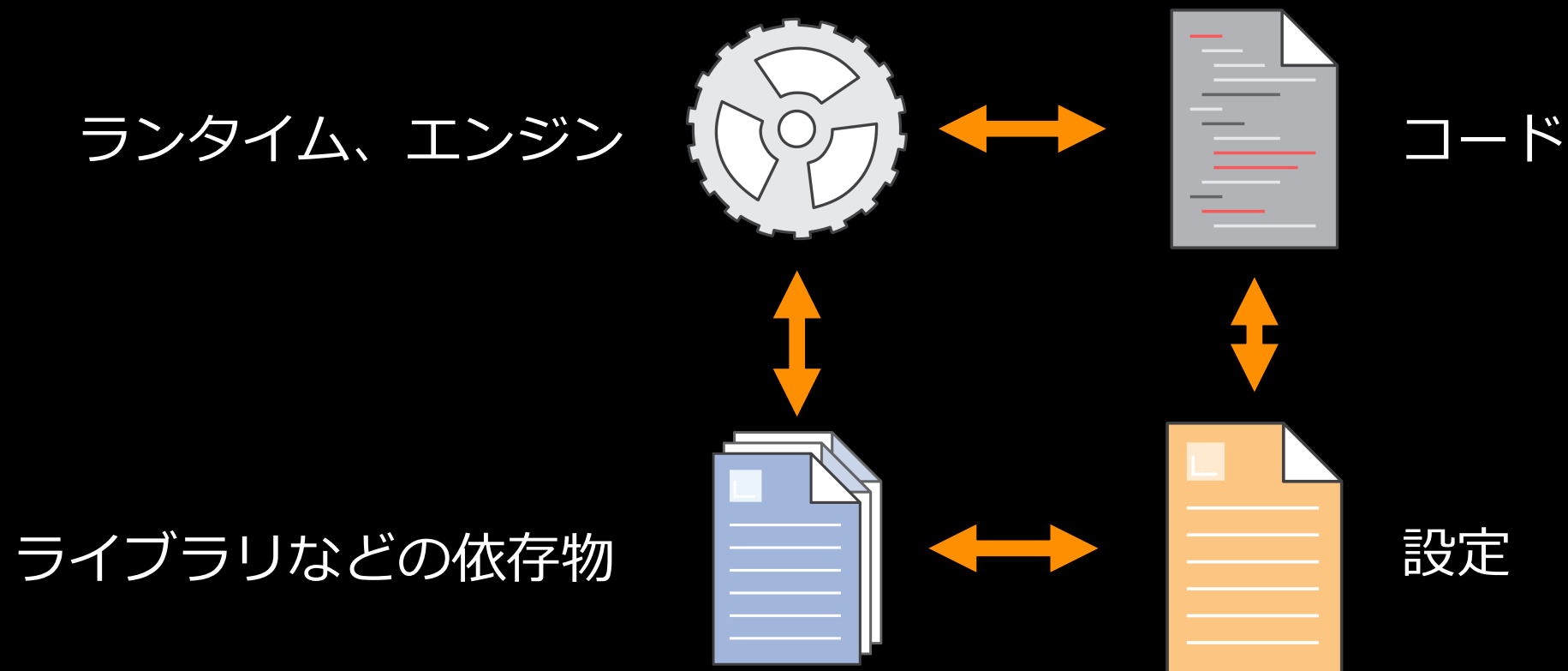
- なぜコンテナか？ ~コンテナの技術的特性~
- コンテナオーケストレーションとは？
- AWS におけるコンテナサービス
- まとめ・次の一歩

# なぜコンテナか？ ~コンテナの技術的特性~

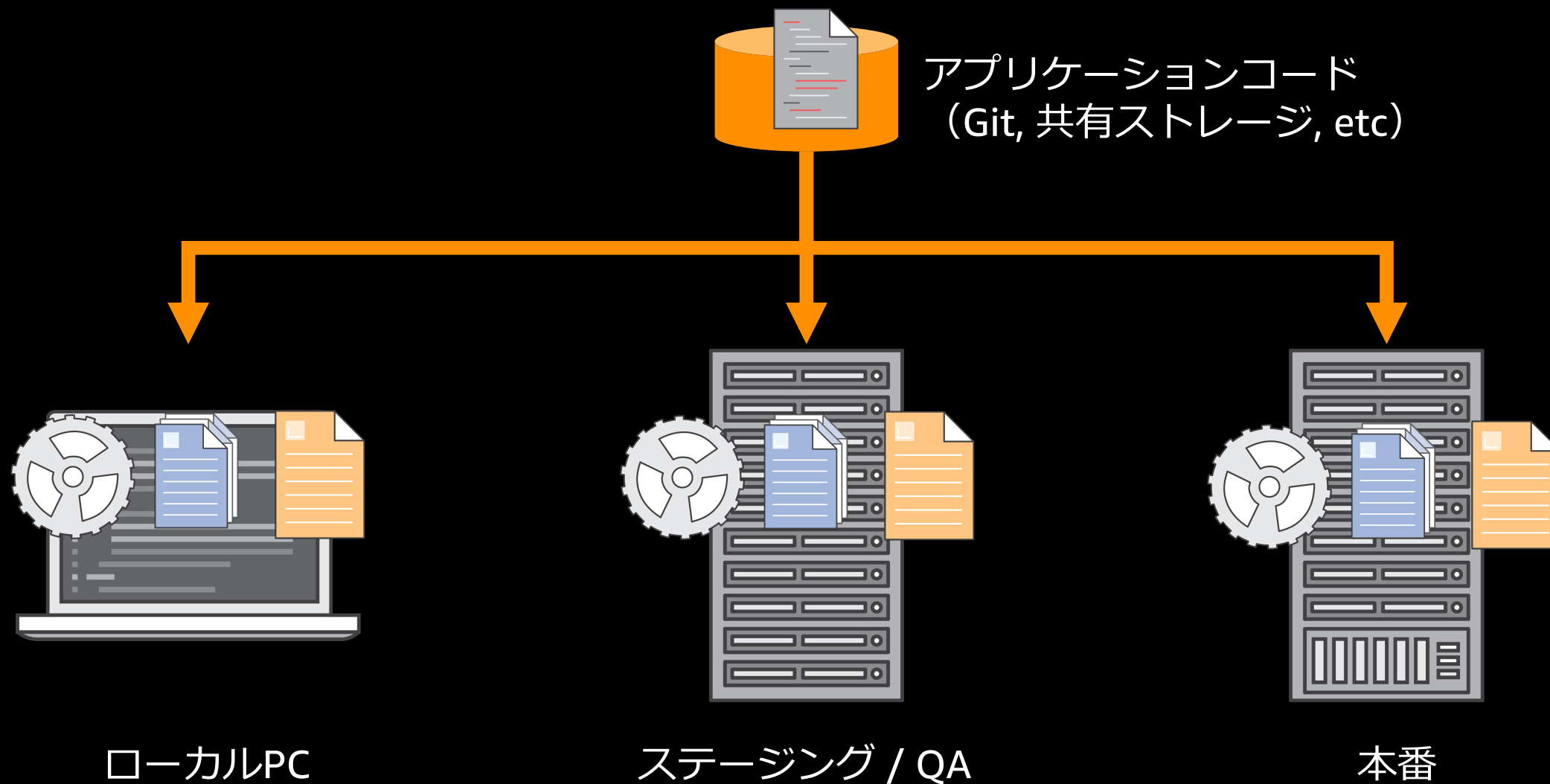


# アプリケーションを構成するコンポーネント

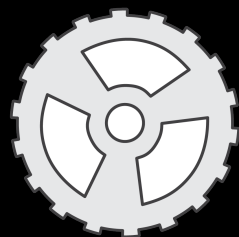
#AWSBuilders



# 異なる複数の環境



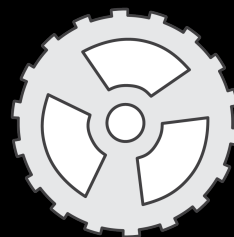
# ローカルでは動いたけど、本番で動かない？



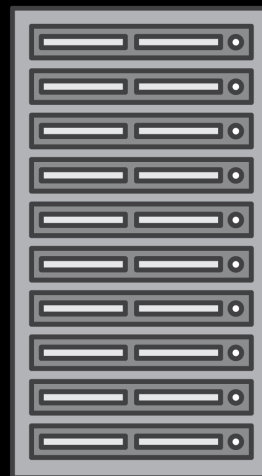
v6.0.0



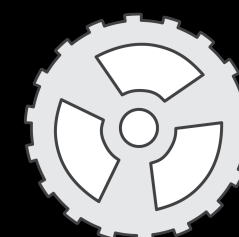
ローカルPC



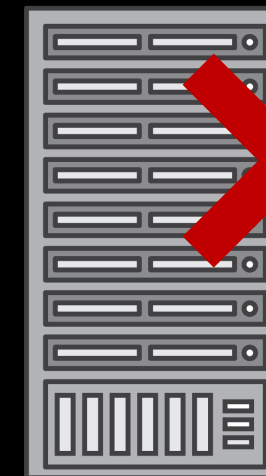
v7.0.0



ステージング / QA



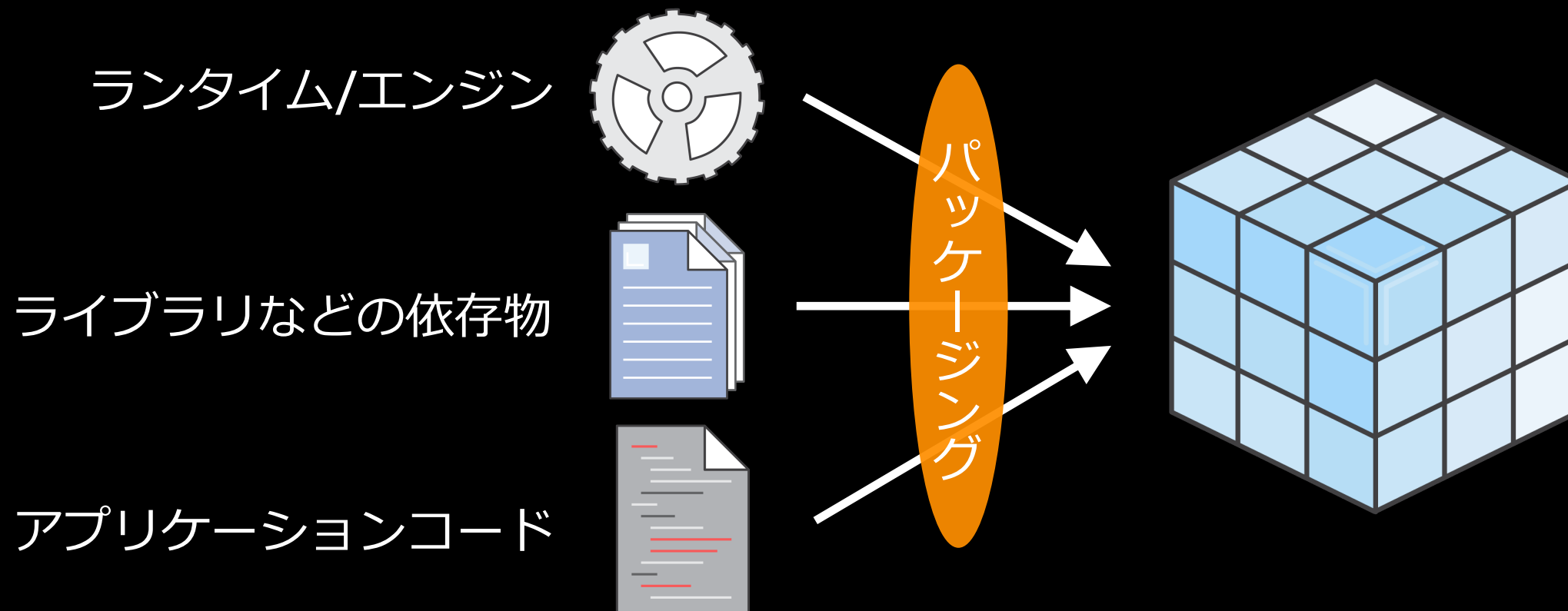
v4.0.0



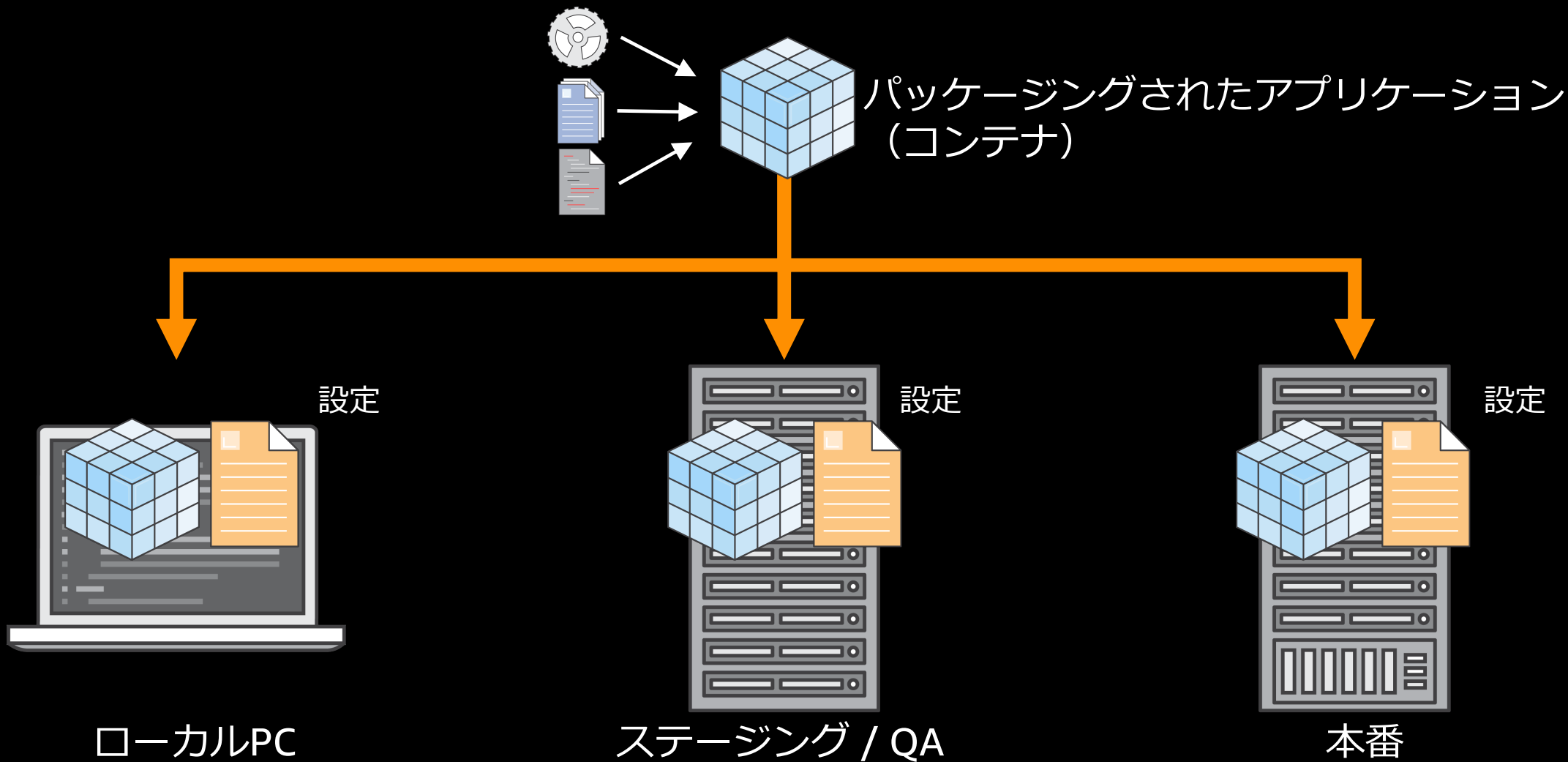
本番



# コンテナが実現すること



# コンテナが実現すること



# Docker とは？

## ◆ コンテナのライフサイクル管理・デプロイツール

- ✓ Docker 社が開発、2013年3月にオープンソース化
- ✓ Apache 2.0 ライセンス

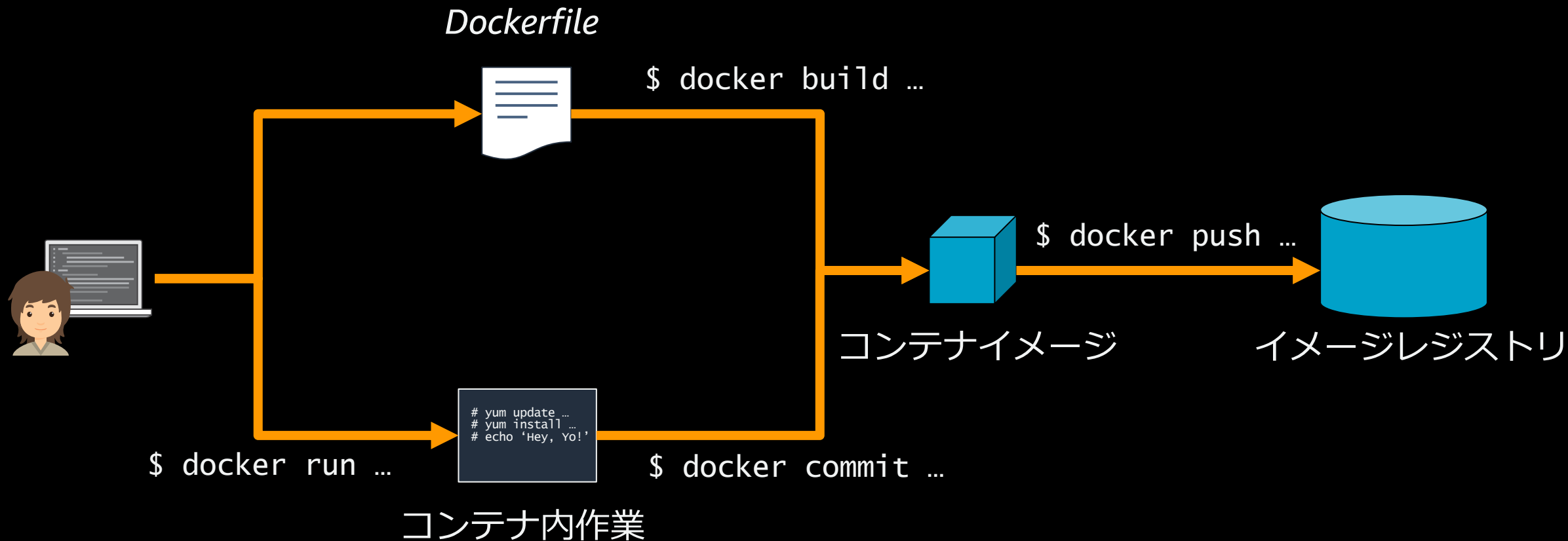


## ◆ Docker の革新

- ✓ アプリケーションのパッケージング（コンテナ化）
- ✓ コンテナ化、コンテナ実行を実現する統一的なコマンド群、エコシステム

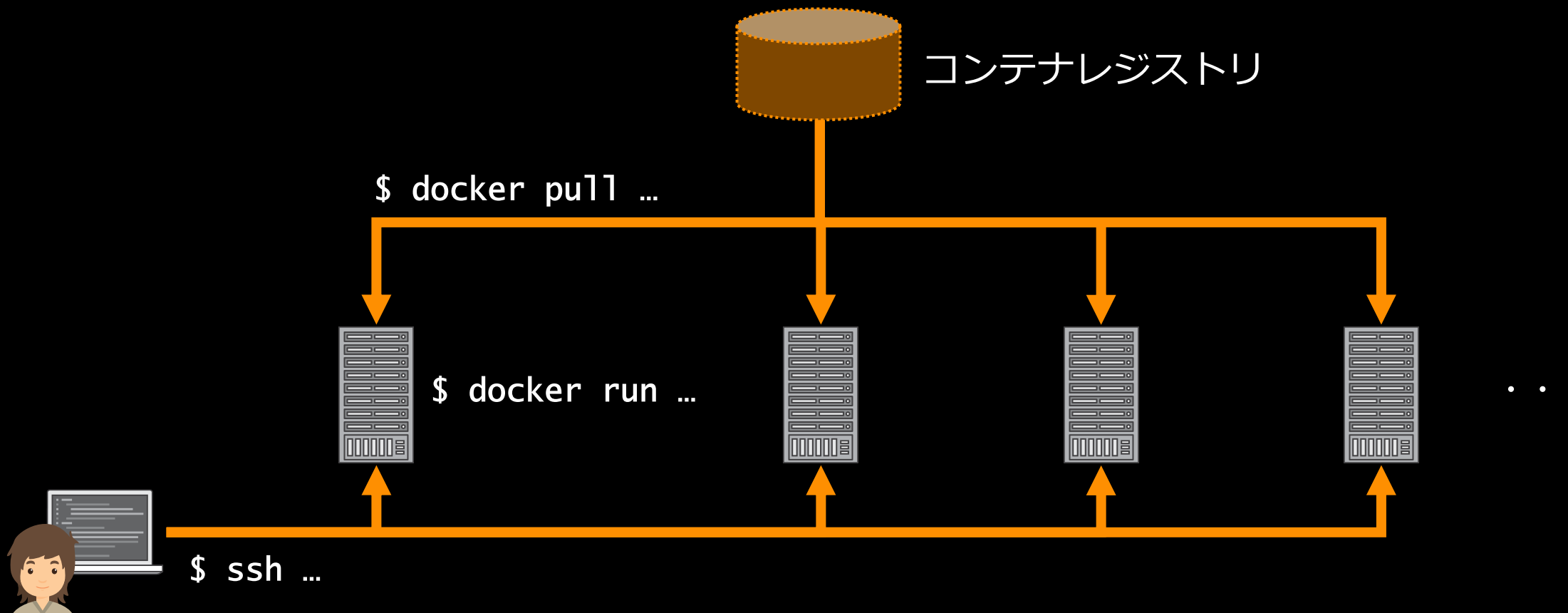
# Docker を利用した基本ワークフロー

- コンテナイメージ作成 -



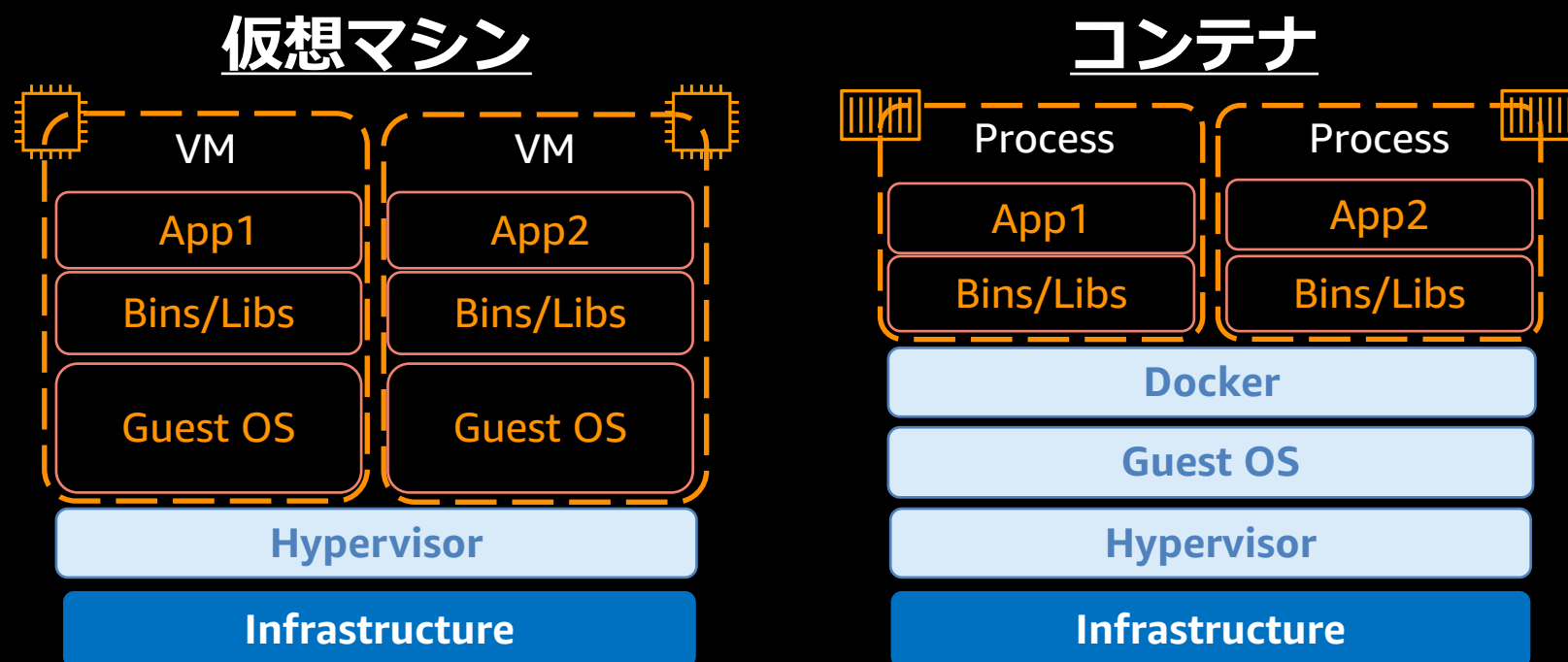
# Docker を利用した基本ワークフロー

- コンテナ実行 -



# 仮想マシンとコンテナの違いって？

リソースが隔離されたOS上のプロセス  
(仮想マシンと同様に「起動・停止・削除」などのライフサイクルを持つ)



# なぜコンテナか？

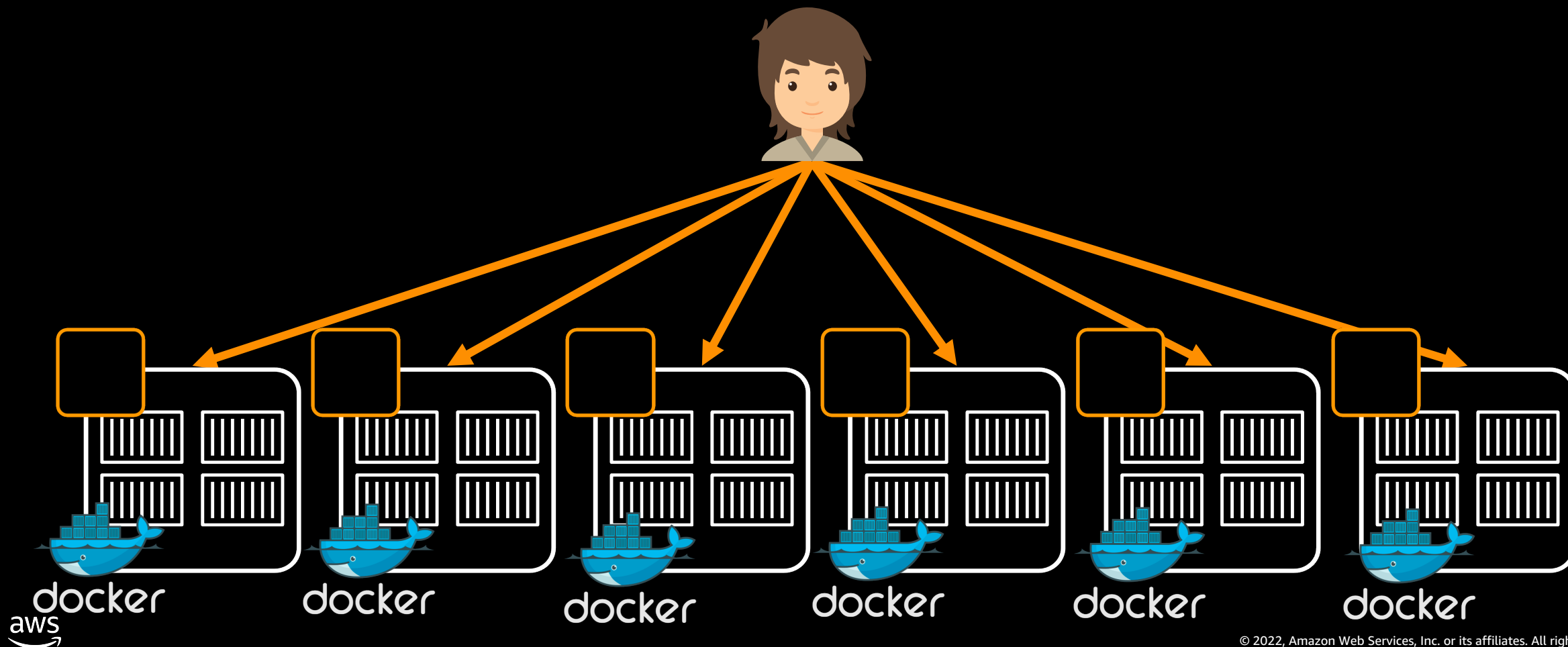
- コンテナの**技術的特性**
  - ✓ アプリケーションの依存物全てを一つにパッケージング可能
  - ✓ パッケージの統一的なデリバリ方法を実現するコマンド群
  - ✓ コンテナを実行するための統一的なコマンド群
- **技術的特性を活かすことで次のような効果が期待できる**
  - ✓ 複数環境にわたる一貫した実行可能性
  - ✓ アプリケーションの可搬性
  - ✓ 高速な開発とリリースサイクルの実現

# コンテナオーケストレーションとは？

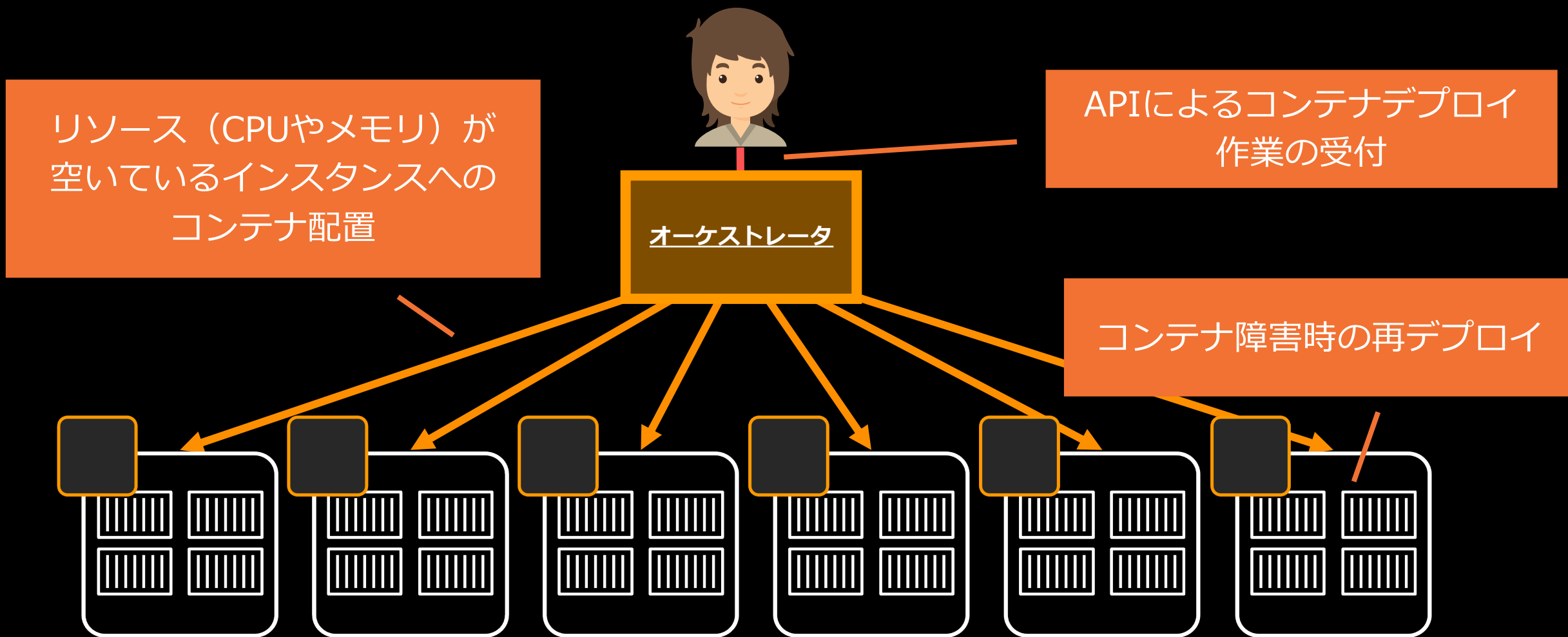


# Docker の基本ワークフローにおける課題

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



# コンテナオーケストレーションによる解決



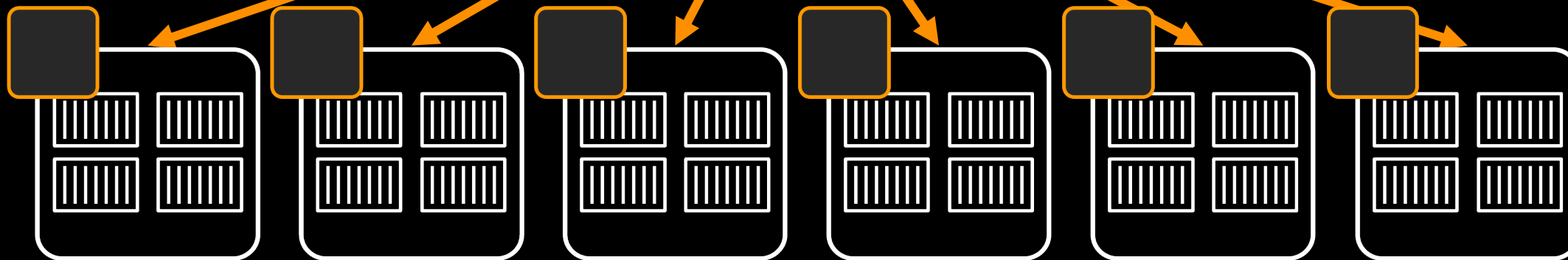
# コンテナオーケストレーションによる解決

「この EC2 インスタンスの  
クラスターでコンテナを  
実行したいです」



オーケストレータ

「このコンテナを3つのAZに  
分散させて10個デプロイして、  
このロードバランサーに  
つないでください」



# AWS におけるコンテナサービス

# AWS におけ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 Elastic  
Container Service  
(Amazon ECS)



Amazon Elastic  
Kubernetes Service  
(Amazon EKS)

## イメージレジストリ

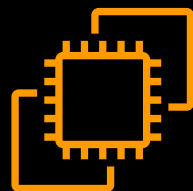
コンテナイメージの格納



Amazon Elastic  
Container Registry  
(Amazon ECR)

## ホスティング

コンテナ実行環境

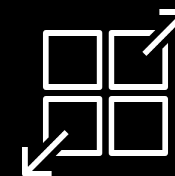


Amazon Elastic  
Compute Cloud  
(Amazon EC2)



AWS Fargate

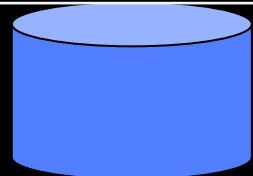
## 手軽にコンテナワーク ロードを実行



AWS App Runner

# AWS のコンテナサービスによる解決

イメージレジストリ



Amazon ECR



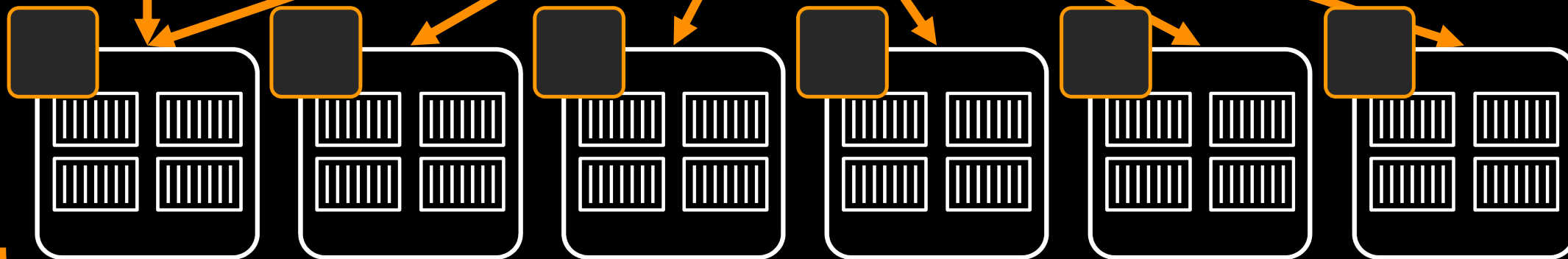
オーケストレータ



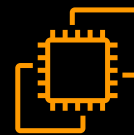
Amazon ECS



Amazon EKS



ホスティング



Amazon EC2



AWS Fargate

# Amazon ECS パワフル&シンプル



**Amazon ECS**

他の AWS サービスとのネイティブな統合

フルマネージドなコントロールプレーン

様々なデプロイツールをエクスペリエンスに合わせて選択

# Amazon EKS: オープン&フレキシブル Kubernetes をオーケストレータとして使いたい方へ



**Amazon EKS**

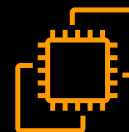
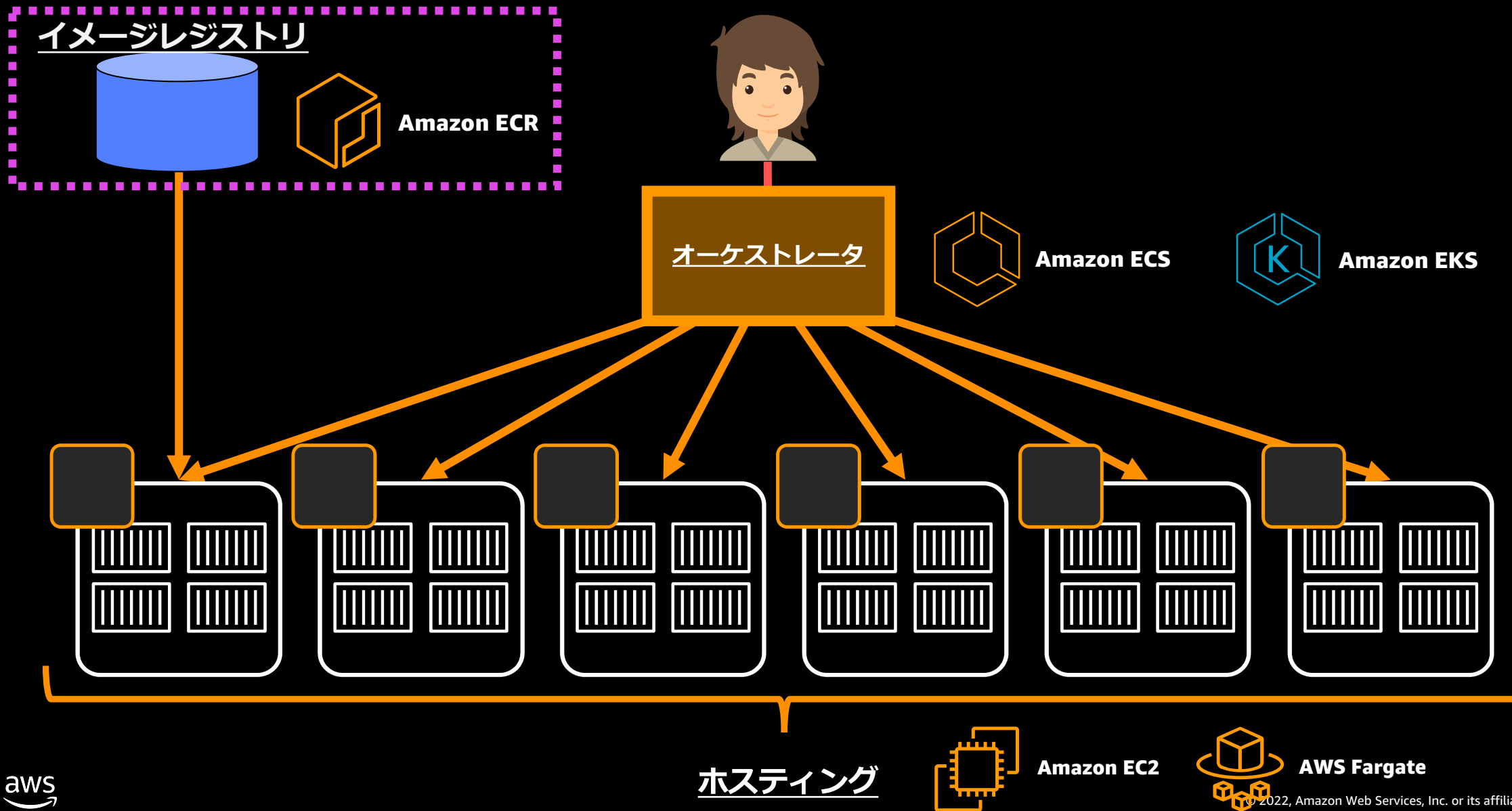
Kubernetes エコシステムとコミュニティ  
による多様な選択肢

セキュアで高可用性を持つ Kubernetes 環  
境を提供

一貫性のあるデプロイのエクスペリエンス



# AWS のコンテナサービスによる解決



# Amazon ECR



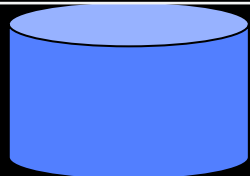
**Amazon ECR**

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

# AWS のコンテナサービスによる解決

イメージレジストリ



Amazon ECR



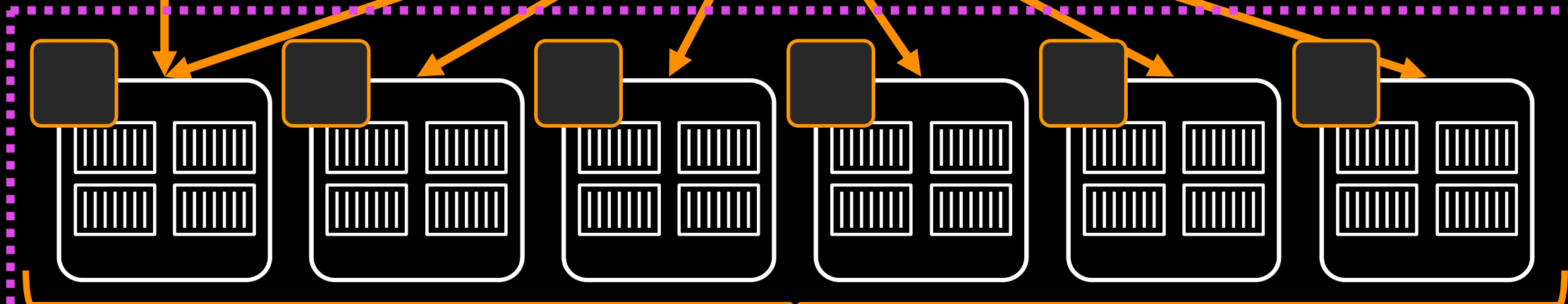
オーケストレータ



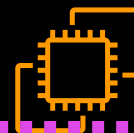
Amazon ECS



Amazon EKS



ホスティング

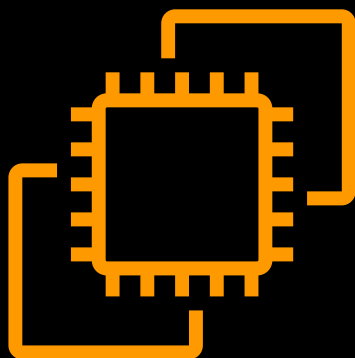


Amazon EC2



AWS Fargate

# Amazon EC2



## Amazon EC2

安全でサイズ変更可能な仮想マシン

多様なインスタンスタイプ

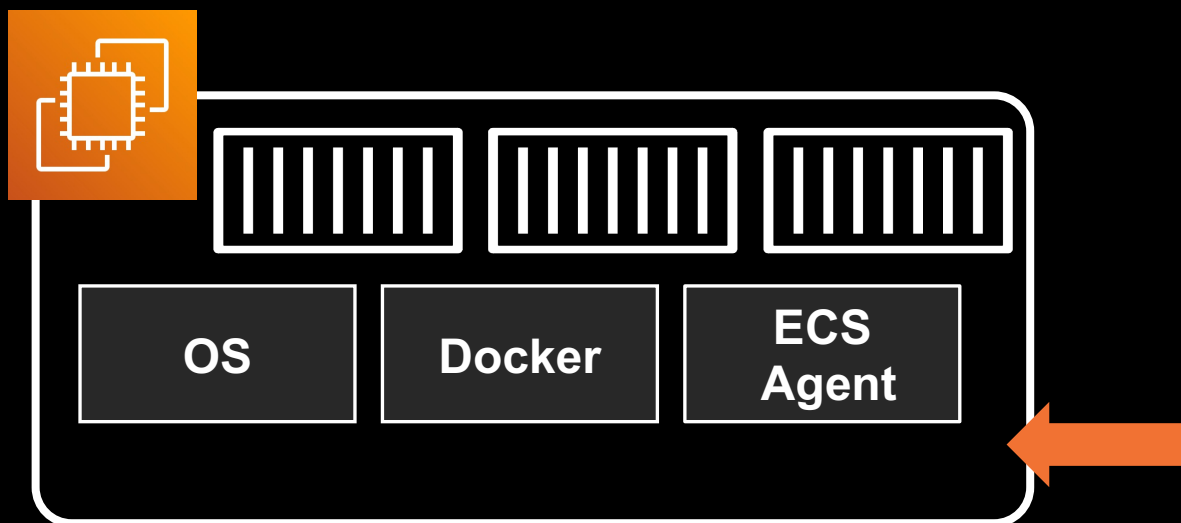
- GPU等の特殊なハードウェアが必要なワークロード
- CPUやメモリが潤沢に必要なワークロード

ホストのカスタマイズ性  
(カーネルチューニング等)

# コンテナ実行環境における課題

## ECS on EC2

(コンテナを仮想サーバー上で動作)



### 課題

- OS やエージェント類へのパッチ当てや更新
- インスタンスタイプの選択
- EC2 インスタンス数のスケーリング (コンテナ数や使用状況に応じた)

# AWS Fargate



**AWS Fargate**

## AWS マネージド

EC2 インスタンスのプロビジョン、スケール、管理不要  
大半のワークロード(4vCPU, 30GBメモリ以下)をカバー

## コンテナネイティブ

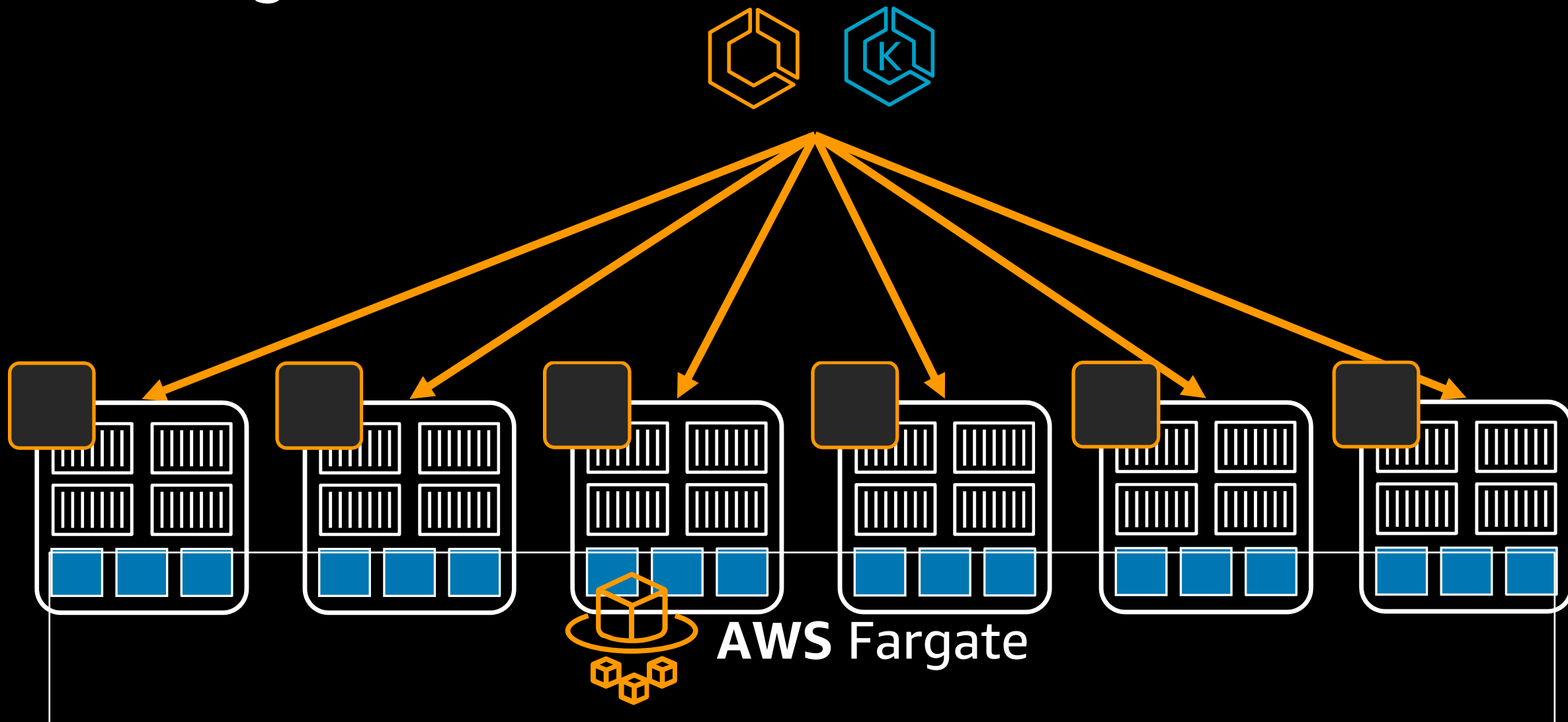
仮想マシンを意識しないシームレスなスケールリング  
コンテナの起動時間・使用リソースに応じた料金設定

## AWS サービスとの連携

VPC ネットワーキング、Elastic Load Balancing、  
IAM、CloudWatch、etc.

# AWS Fargate によるホスティングイメージ

#AWSBuilders



# AWS におけ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 Elastic Container Service (Amazon ECS)



Amazon Elastic Kubernetes Service (Amazon EKS)

## イメージレジストリ

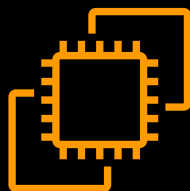
コンテナイメージの格納



Amazon Elastic Container Registry (Amazon ECR)

## ホスティング

コンテナ実行環境

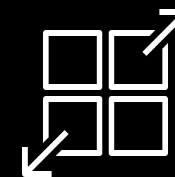


Amazon Elastic Compute Cloud (Amazon EC2)



AWS Fargate

## 手軽にコンテナワークロードを実行



AWS App Runner



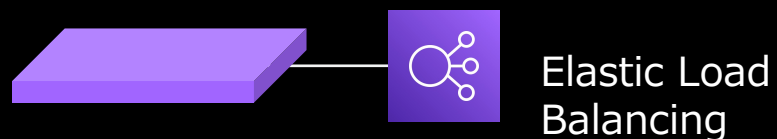
# クラウドインフラストラクチャには 多様なサービスが存在



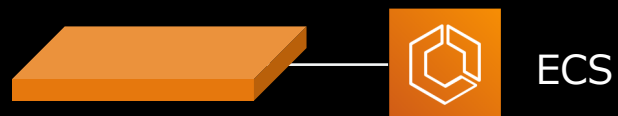
AWS CodeBuild



AWS Auto Scaling



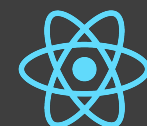
Elastic Load  
Balancing



ECS



AWS Fargate

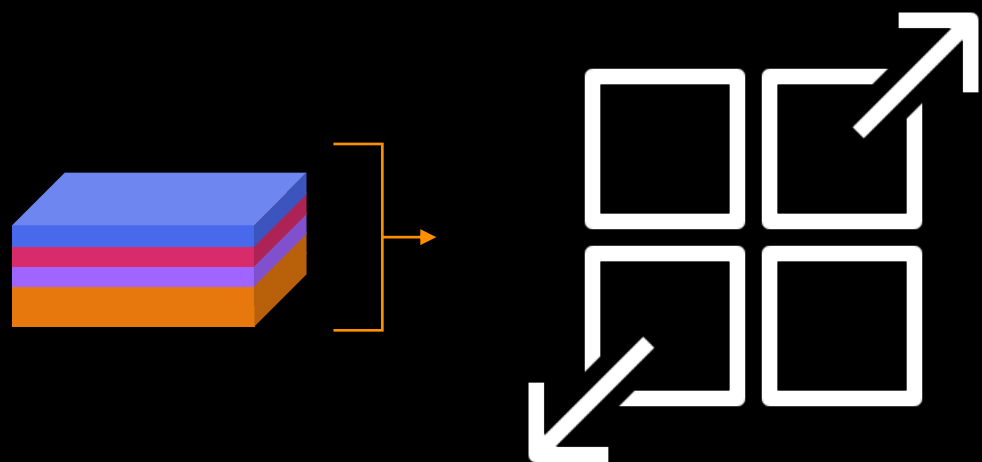


フロントエンドアプリケーション

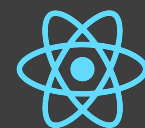


バックエンドアプリケーション

# クラウドインフラストラクチャには 多様なサービスが存在



**AWS App Runner**



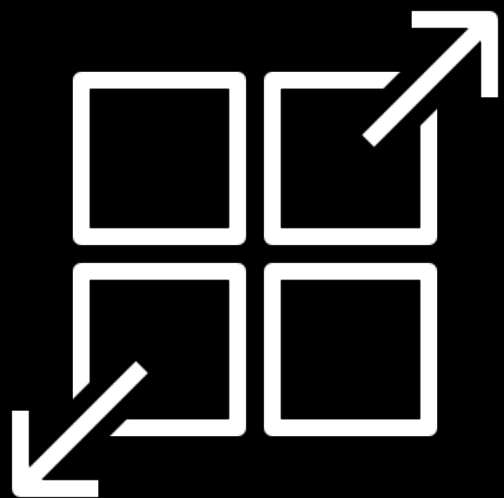
フロントエンドアプリケーション



バックエンドアプリケーション

# AWS App Runner - 手軽にコンテナワークロードを実行する

小規模から大規模まであらゆるスケールの Web アプリケーションを素早く展開



**AWS App Runner**

ネットワーク、サーバ、デプロイパイプライン、  
コンテナオーケストレーションツールの管理が不要

コンテナレジストリまたは Git リポジトリから  
直接デプロイ

アプリケーションの実行時間に対して秒単位での  
課金体系

# まとめ・次の一歩

# AWS におけ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 Elastic Container Service (Amazon ECS)



Amazon Elastic Kubernetes Service (Amazon EKS)

## イメージレジストリ

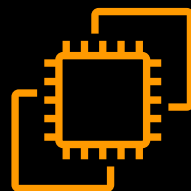
コンテナイメージの格納



Amazon Elastic Container Registry (Amazon ECR)

## ホスティング

コンテナ実行環境

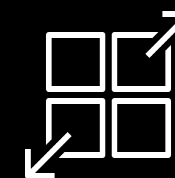


Amazon Elastic Compute Cloud (Amazon EC2)



AWS Fargate

## 手軽にコンテナワークロードを実行



AWS App Runner

# コンテナを利用してサービスを開発しましょう！



## アプリケーション

---

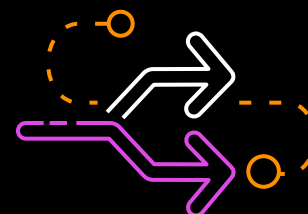
Web アプリケーション  
モバイルアプリケーション  
IoT  
データ処理



## 共通サービス プラットフォーム

---

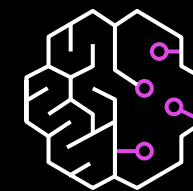
CI/CD  
IaaS  
マネジメント、セキュリティ  
、ガバナンス  
ロギング、モニタリング



## エンタープライズ app のマイグレーション

---

.NET Classic Windows app  
Linux app  
3rd party app



## 機械学習 (ML)

---

自動運転車  
レコメンデーションエンジン  
不正検出  
Chatbots

# 次の一歩 – ワークショップにチャレンジ –

- ECS Workshop : <https://ecsworkshop.com/>
- EKS Workshop : <https://www.eksworkshop.com/>
- App Runner Workshop : <https://www.apprunnerworkshop.com/>



**aws**

Search...

- Introduction
- Start the workshop
- Deploying
- Microservices to ECS
- Monitoring ECS
- Clusters and Containers
- Capacity Providers
- Blue/Green Deployments

## Amazon ECS Workshop



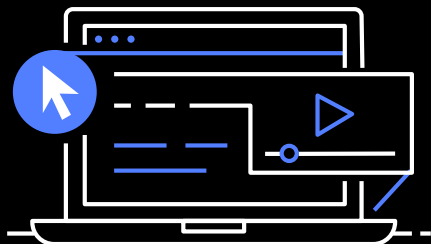
In this workshop, we will launch a frontend and multiple backend services on Amazon Elastic Container Service, and explore how you might adopt this workflow into your environment.

# 次の一步 - 参考となる資料リンク -

- BlackBelt ([資料一覧](#) | [Youtube Playlist](#))  
※以下は一部抜粋。今後増えていきます！
  - なぜ今コンテナなのか ([資料](#) | [Youtube](#))
  - AWSコンテナ全体概要 ([資料](#) | [Youtube](#))
  - Docker 入門 ([資料](#) | [Youtube](#))
  - ECS 入門 ([資料](#) | [Youtube](#))
  - ECS Fargate 入門 ([資料](#) | [Youtube](#))
  - EKS 入門 ([資料](#) | [Youtube](#))
  - コンテナとサーバレスの使い分け ([資料](#) | [Youtube](#))
- 2020 AWS Dev Day Online Japan
  - そのコンテナでサービスインできますか? ([資料](#))
  - 第2回 AWS Fargate かんたんデプロイ選手権 ([資料](#))
- 2020 AWS re:Invent
  - 202現実世界で活用するAWS サーバーレス・コンテナサービス ([資料](#))
- 2021 AWS Summit Online
  - コンテナ・サーバーレスを使えばモダンアプリケーションになりますか? ([資料](#) | [Youtube](#))
  - AWS でのコンテナワークロードにおける多様なビルディングブロックの選択肢 ([資料](#) | [Youtube](#))
- 2021 AWS Dev Day Online Japan
  - アプリケーション開発者は Amazon ECS あるいは Kubernetes のどこまでを知っていればいいのか ([資料](#) | [Youtube](#))
- AWS コンテナサービスの公開ロードマップ
  - <https://github.com/aws/containers-roadmap>



# AWS デジタルトレーニング



学習方法を柔軟に選ぶことができます。

550以上の無料デジタルトレーニングでクラウドのスキルを身につけることができ、クラスルームトレーニングでさらに深く学ぶこともできます。

## おすすめのコース

- **[AWS Cloud Practitioner Essentials \(Japanese\)](#)**

このコースは、特定の技術領域ではなく、アマゾン ウェブ サービス (AWS) クラウドを全体的に理解したい方を対象としています。受講者は、AWS クラウドの概念、AWS のサービス、セキュリティ、アーキテクチャ、料金、サポートについて学習し、AWS クラウドについての知識を深めます。このコースは、AWS Certified Cloud Practitioner 試験の準備にも役立ちます。

- **[Getting Started with AWS Security, Identity, and Compliance \(Japanese\)](#)**

このコースでは、AWS のセキュリティテクノロジー、ユースケース、メリット、サービスの概要について説明します。AWS のセキュリティ、アイデンティティ、コンプライアンスのカテゴリにおけるさまざまなサービスを紹介します。このコースを修了すると、クラウド内のセキュリティの重要性を理解し、データの保護に使用する AWS のサービスを選択できるようになります。

- **[Moving to Managed Databases on AWS \(Japanese\)](#)**

リフトアンドシフト戦略は、クラウドを利用しようとする組織が一般的に採用する方法です。このコースでは、AWS ヒーローである Alex DeBrie が、アマゾン ウェブ サービス (AWS) においてセルフマネージドデータベースからフルマネージドデータベースソリューションに移行することのメリットについて説明します。また、AWS クラウドに移行する際に考慮すべき重要な要素についても学習します。最後に、この Moving to Managed Databases on AWS の 8 つのコースのカリキュラムの一環として、一般的なデータベースの移行に関するユースケースのコースをご紹介します。

# 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](mailto:aws-apj-marketing@amazon.com)



[twitter.com/awscloud\\_jp](https://twitter.com/awscloud_jp)



[facebook.com/600986860012140](https://www.facebook.com/600986860012140)



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



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



[twitch.tv/aws](https://www.twitch.tv/aws)

# Thank you!

堀内 保大 (Yasuhiro Horiuchi)

horiuyas@amazon.co.jp

 @ka\_shino\_ki

