



ECS 入門

AWS Black Belt Online Seminar

Solutions Architect, Industry Solutions

Amazon Web Services Japan K.K.

杉本 晋吾

2021-Aug



このセッションで扱うこと

- Amazon Elastic Container Service (ECS) に関する基本的な情報
 - オーケストレーターの目的
 - Amazon Elastic Compute Cloud (EC2) で ECS を利用する
 - Amazon ECS での基礎的な用語

このセッションの主な対象者

- Amazon ECS や AWS におけるコンテナ実行環境に興味がある
- Docker などのコンテナの概念は多少知っている
- AWS 上で EC2 などのサービスに触れたことがある

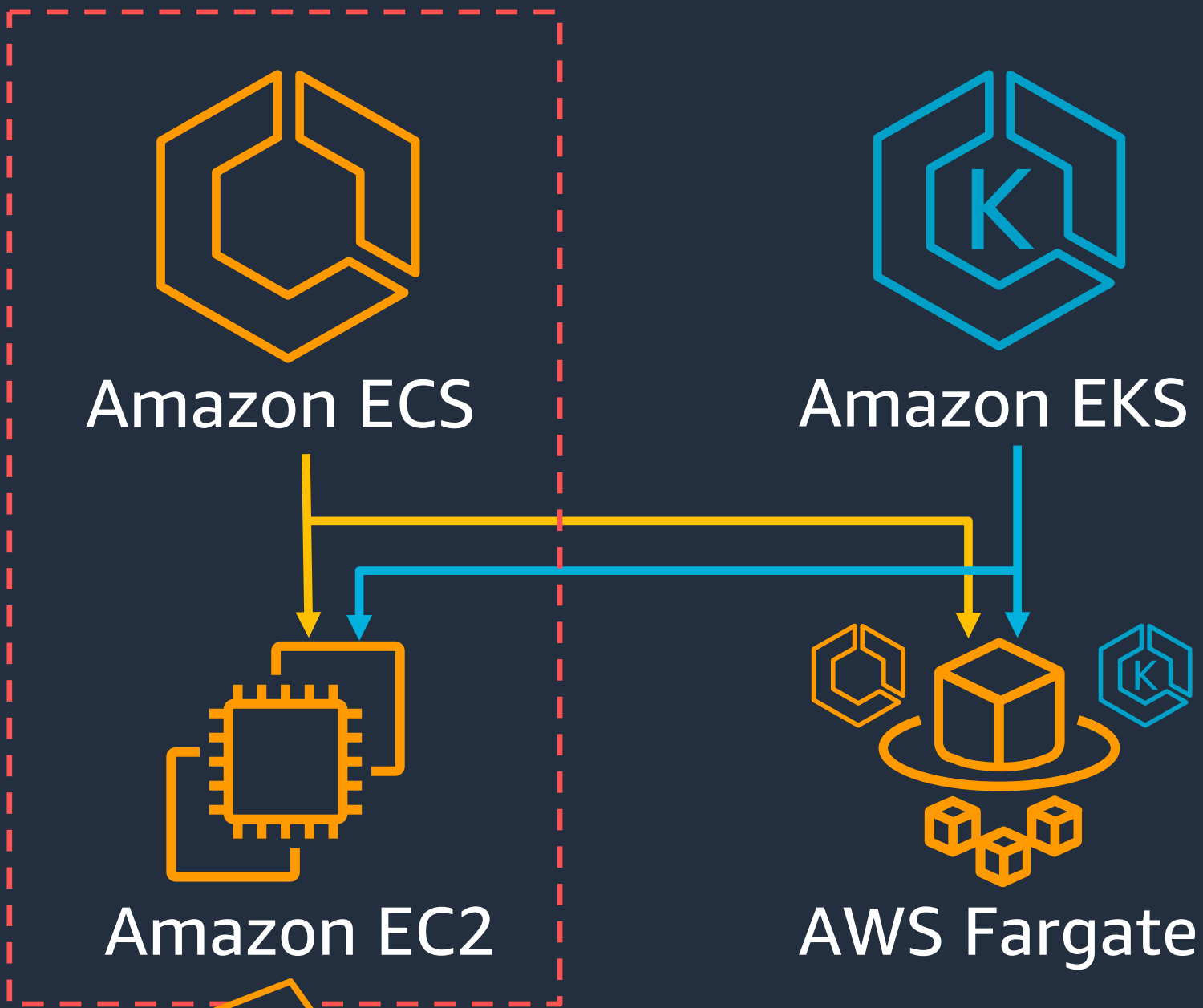
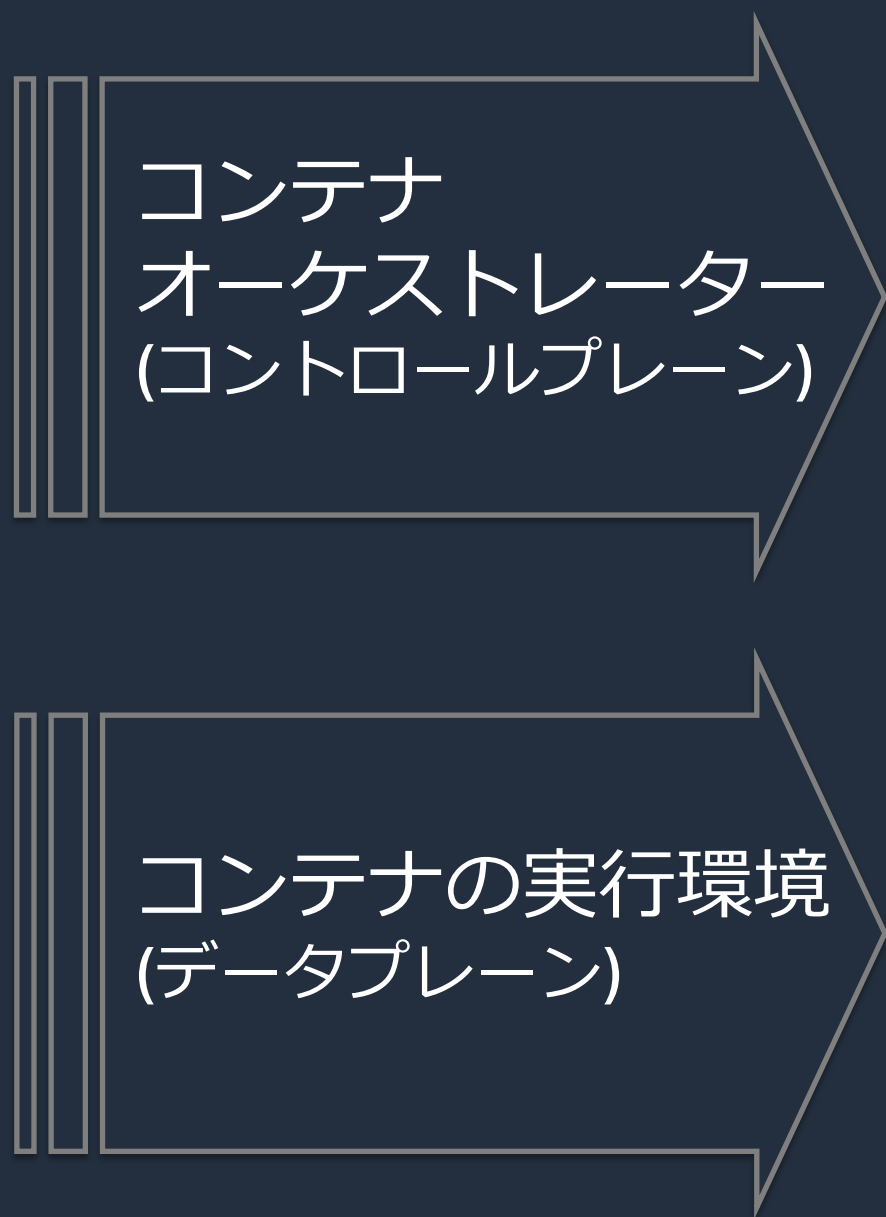
Amazon Elastic Container Service (Amazon ECS)



パワフル & シンプル

- クラウドでコンテナを本番環境利用するための **オーケストレーター**
- 他の AWS サービスと高度に連携しコンテナを大規模に実行
- **フルマネージドなコントロールプレーン**
- 多様なワークロードをサポートする「タスク」「サービス」という **シンプルなリソース表現**

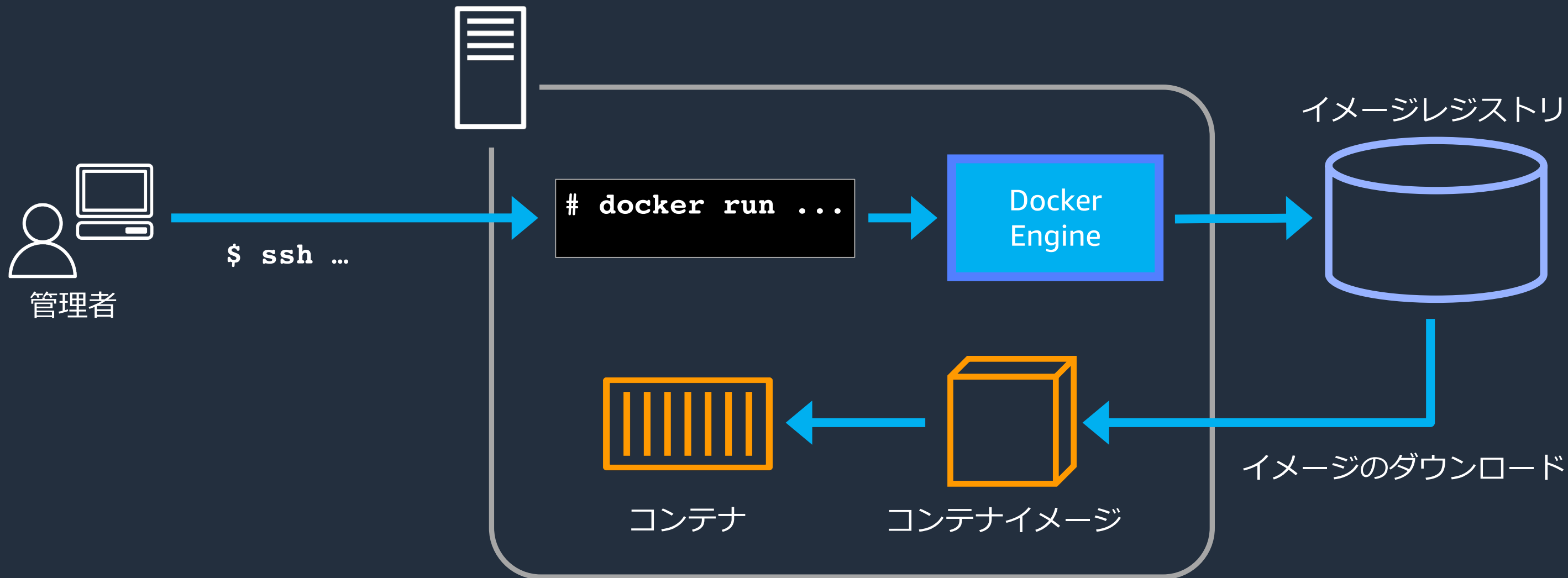
AWS 上でのコンテナ実行における選択肢



このセッションでの話題の範囲

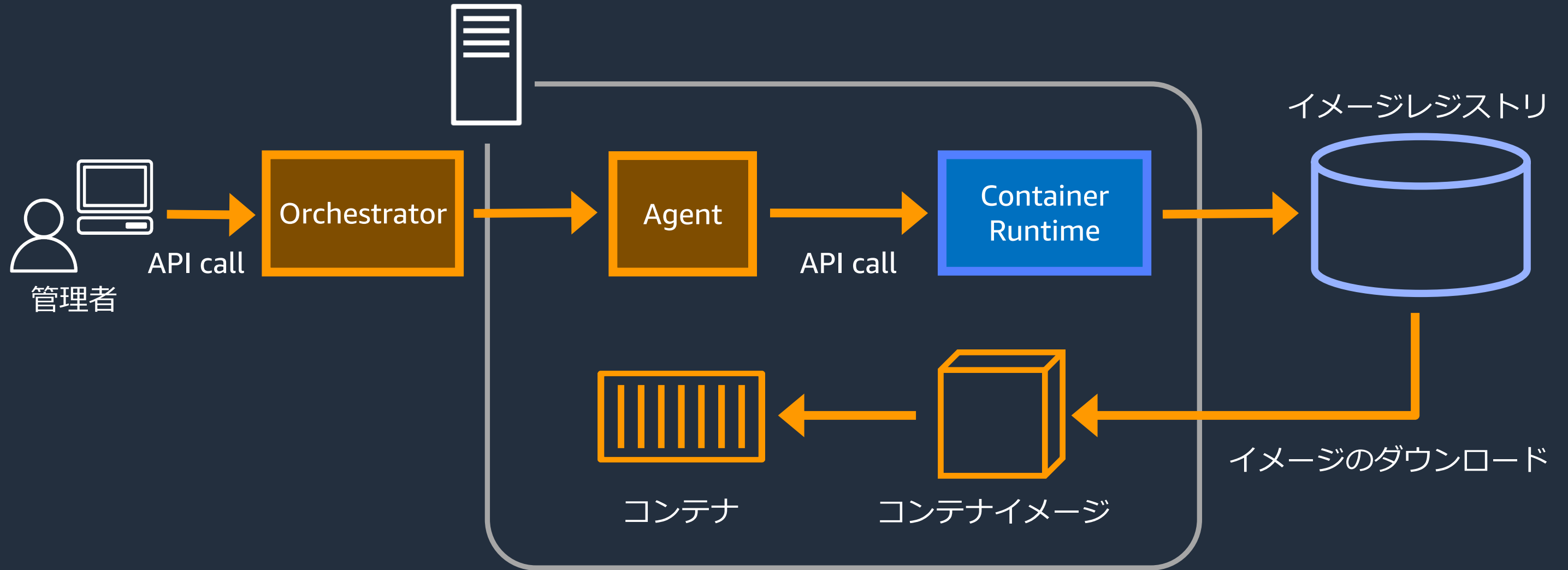
コンテナオーケストレーターの目的

サーバー上でのコンテナ実行 (dockerの場合)



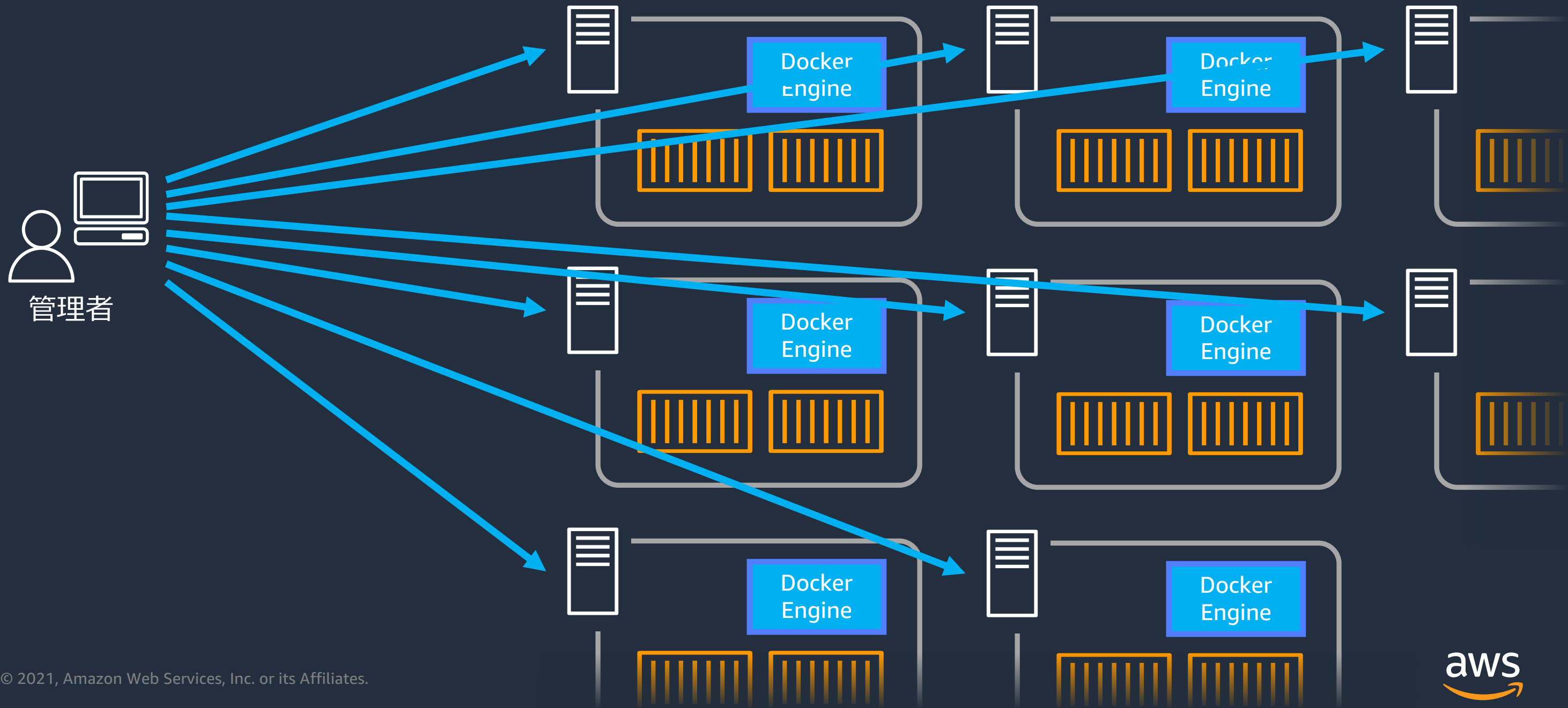
コンテナオーケストレーターの目的

コンテナオーケストレーションの仕組み



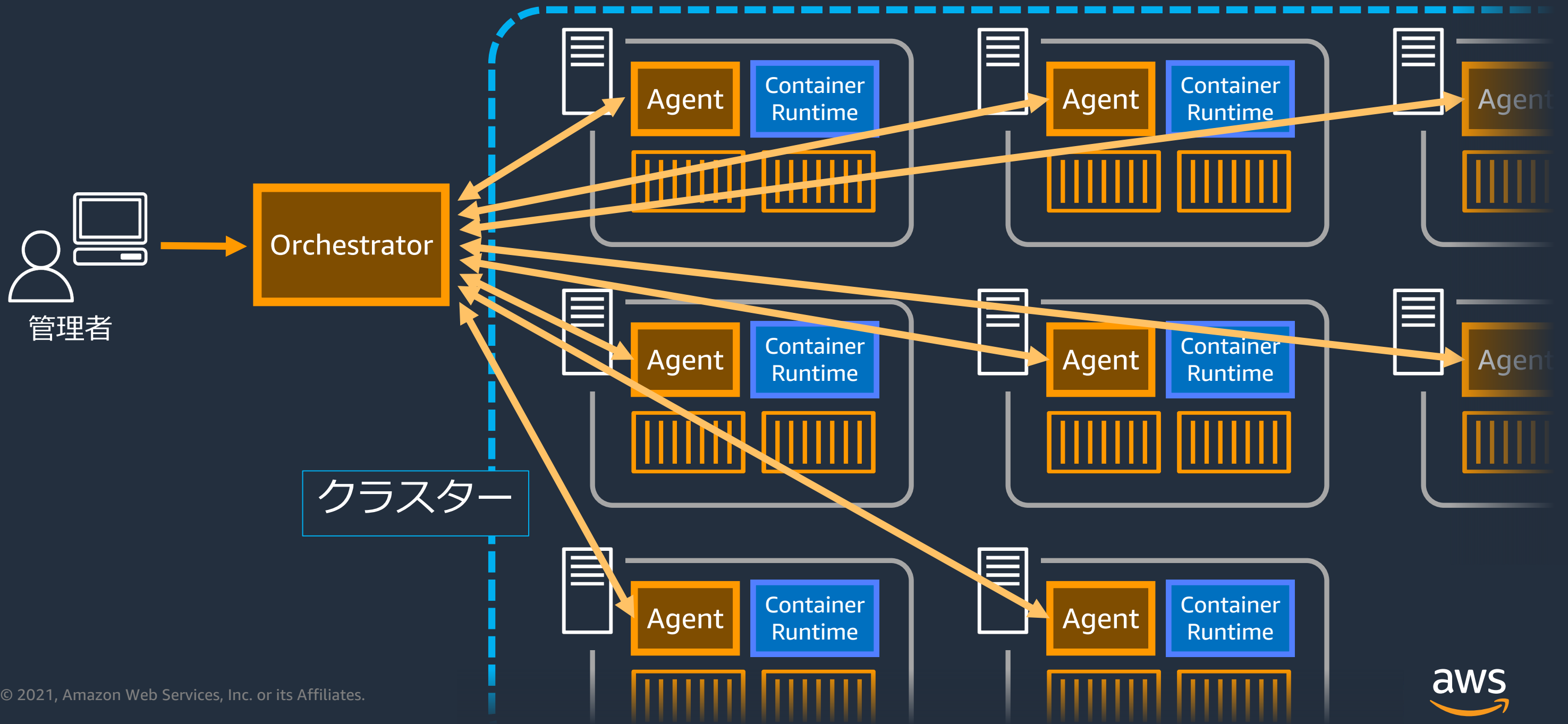
コンテナオーケストレーターの目的

コンテナホストが多数になると、非効率でミスオペレーションにつながる



コンテナオーケストレーターの目的

コンテナオーケストレーションによるクラスター管理



Amazon ECS の動作イメージ (on EC2)



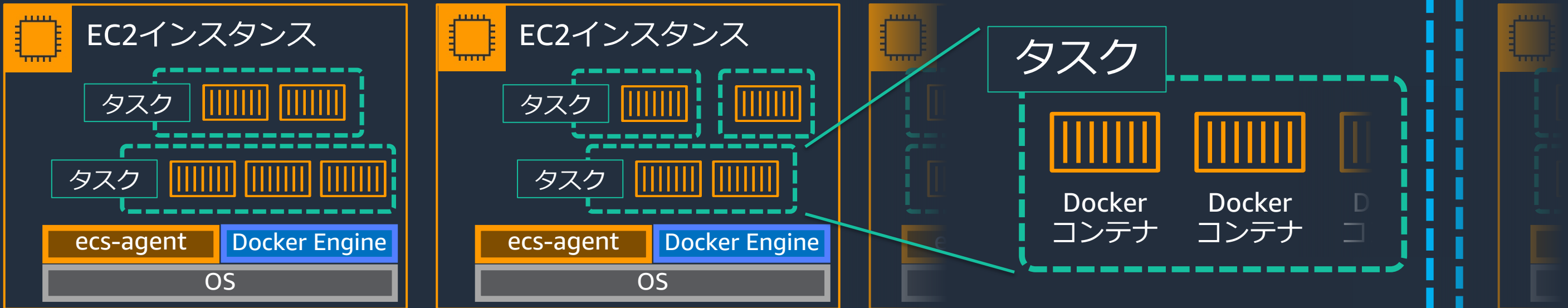
Amazon ECS

スケジューリングとオーケストレーション

クラスター管理

配置エンジン

クラスター

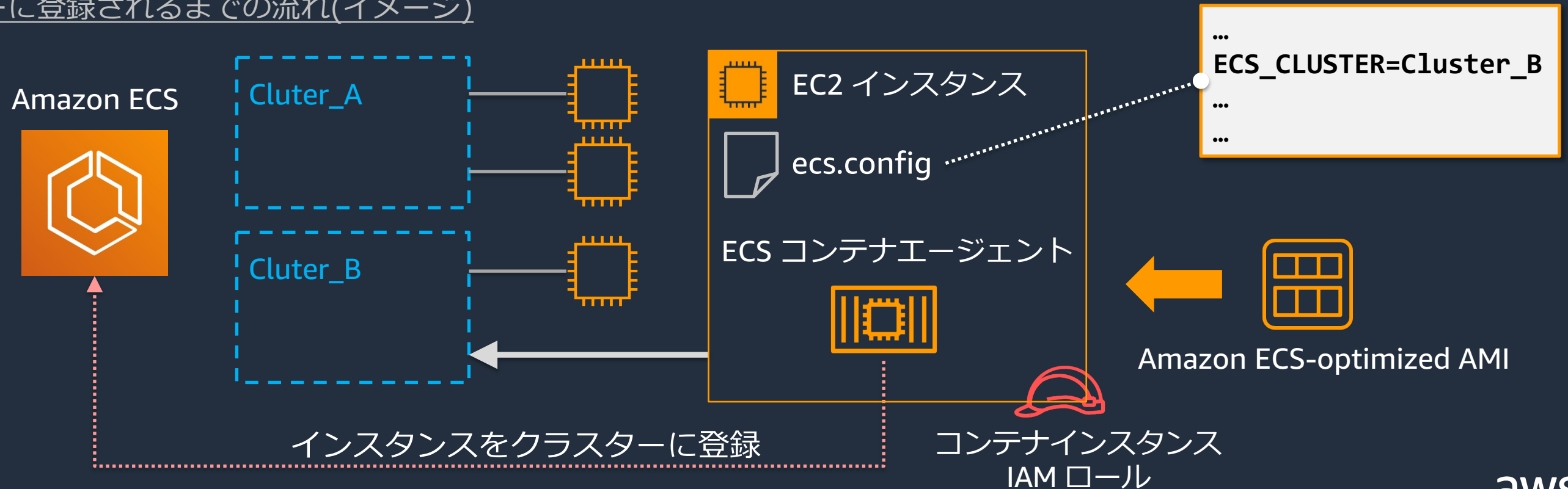


ECS コンテナインスタンス

コンテナの実行環境として ECS クラスタに登録されている EC2 インスタンス

- ECS コンテナエージェントや Docker デーモンを実行
- EC2 起動タイプのタスクはアクティブなコンテナインスタンスに配置される

クラスタに登録されるまでの流れ(イメージ)



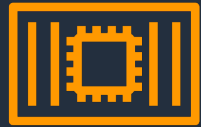
Amazon ECS-optimized AMI



コンテナインスタンスの要件/推奨事項にしたがって
事前構成されたAMI

- インストール済み
 - Docker デーモン
 - ECS コンテナエージェントetc ...
- Amazon Linux AMI / Amazon Linux 2 / Windows / GPU など
を用意

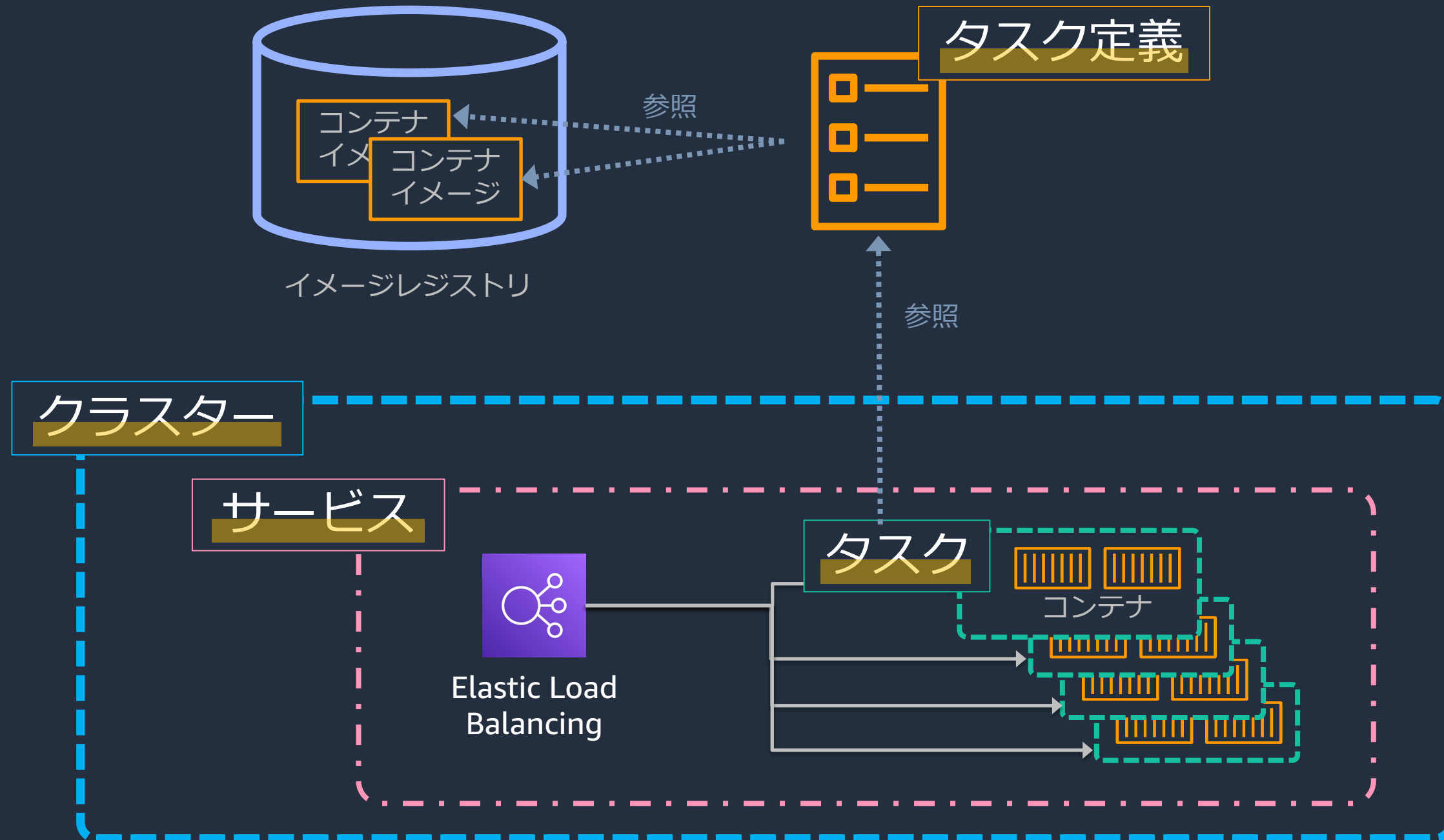
ECS コンテナエージェント



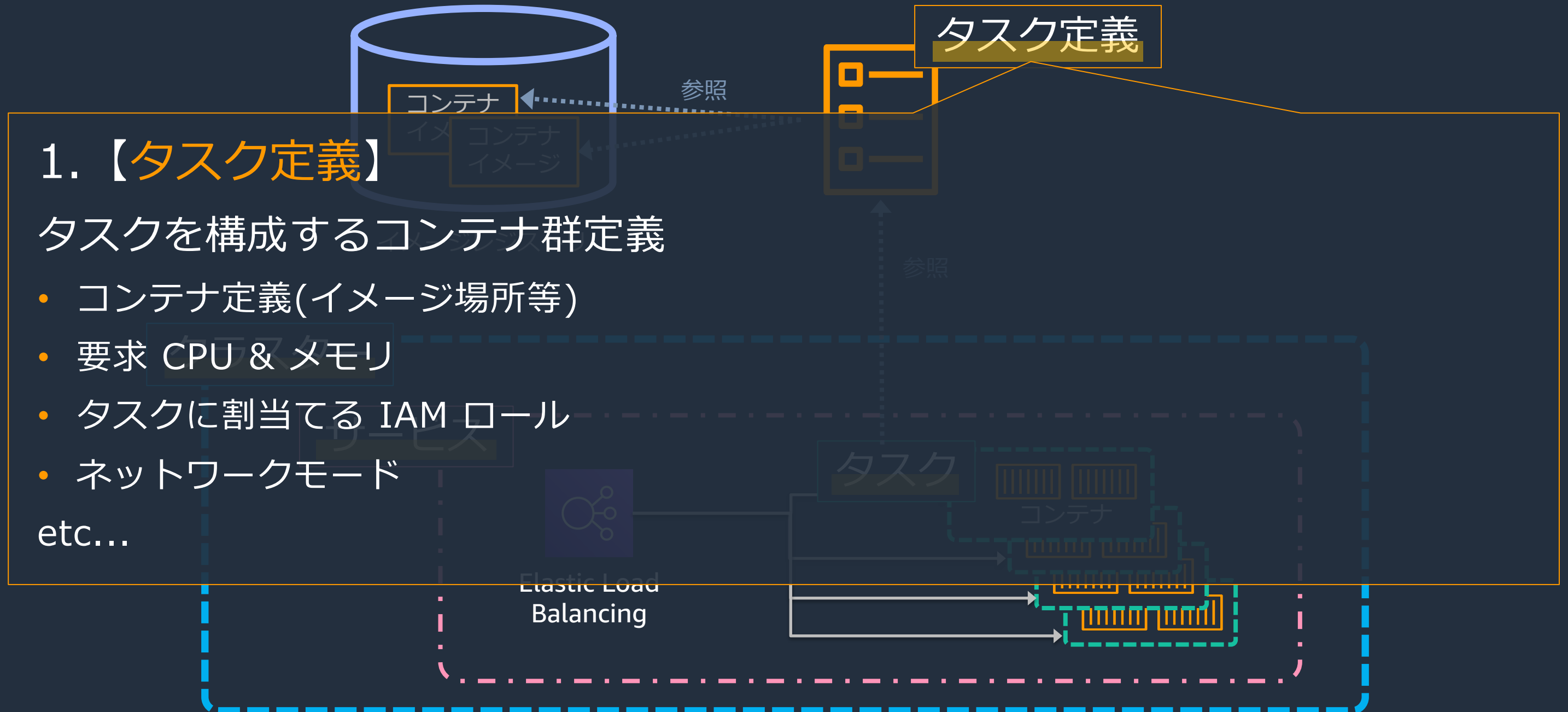
ECS コントロールプレーンと通信して様々な処理を行う

- コンテナインスタンスの管理やタスクの実行/停止など
- Amazon ECS-optimized AMI を元に起動した場合、事前にインストール済み
- このエージェント自体もコンテナで実行される

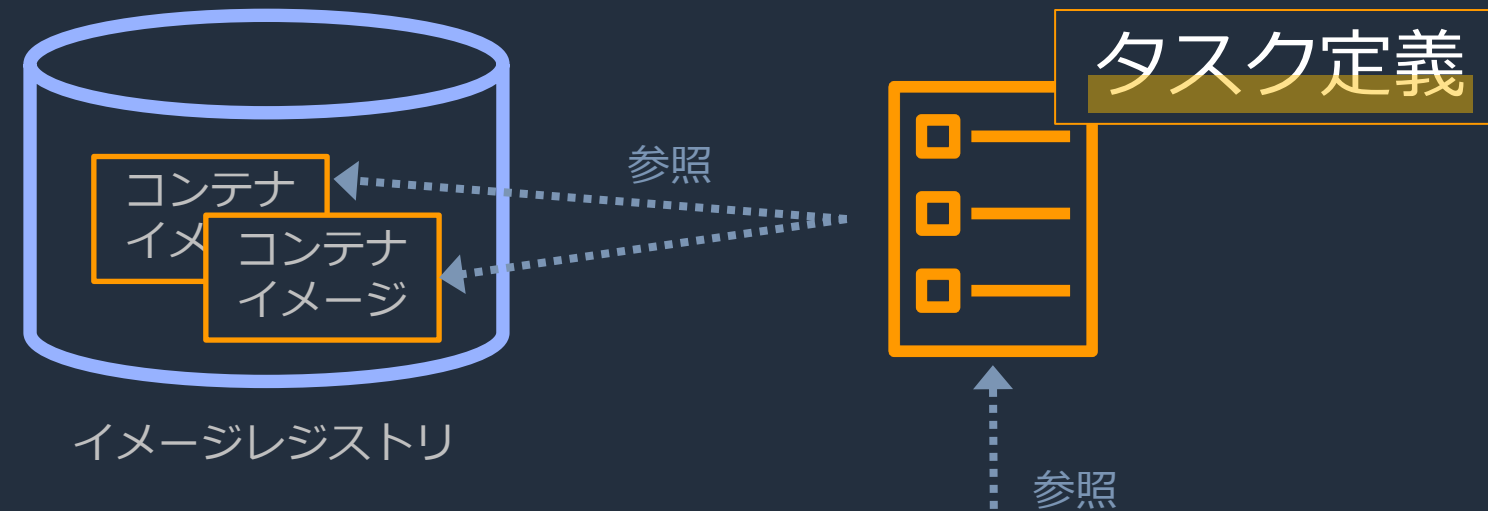
Amazon ECS の主要な 4 つの構成要素



Amazon ECS の主要な 4 つの構成要素



Amazon ECS の主要な 4 つの構成要素

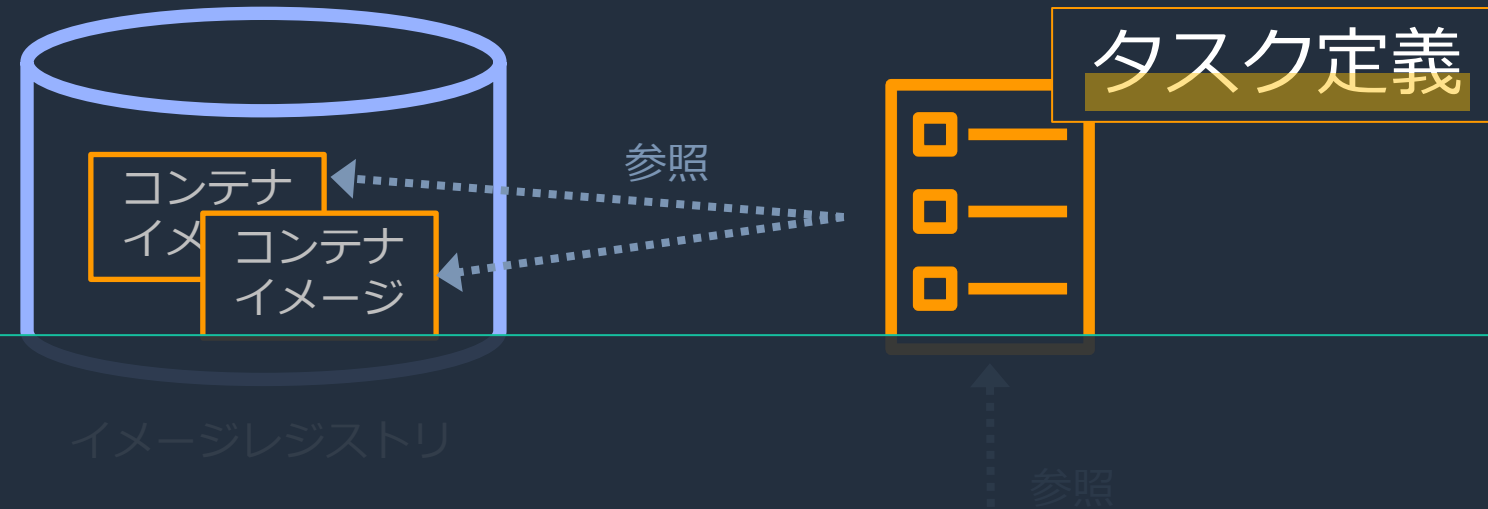


クラスター

2. 【クラスター】

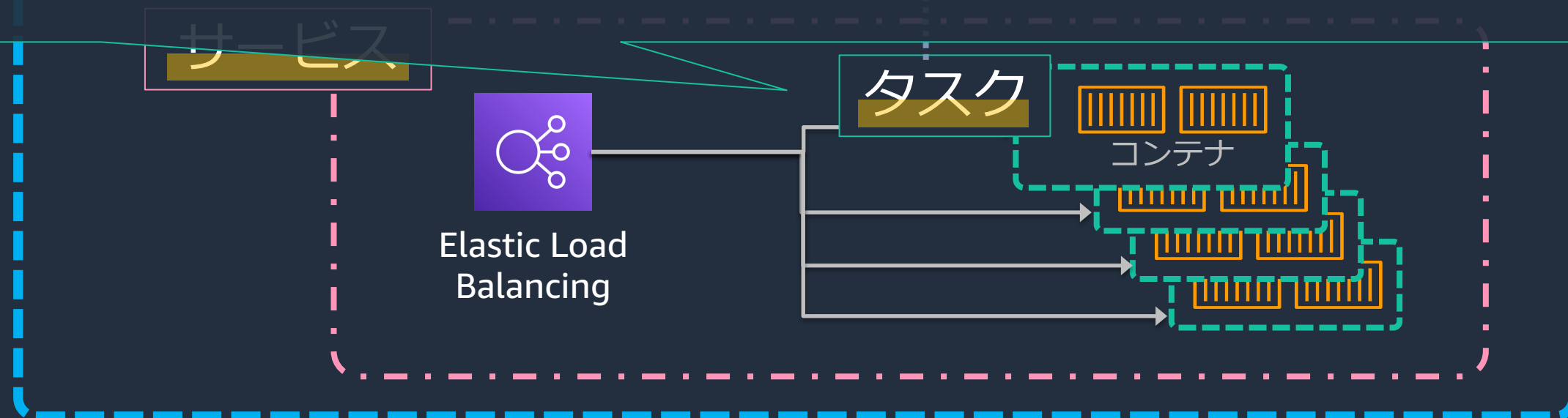
- 実行環境の境界
- IAM 権限の境界 (クラスターに対する操作)
- スケジュールされたタスクの実行を設定可能

Amazon ECS の主要な 4 つの構成要素



3. 【タスク】

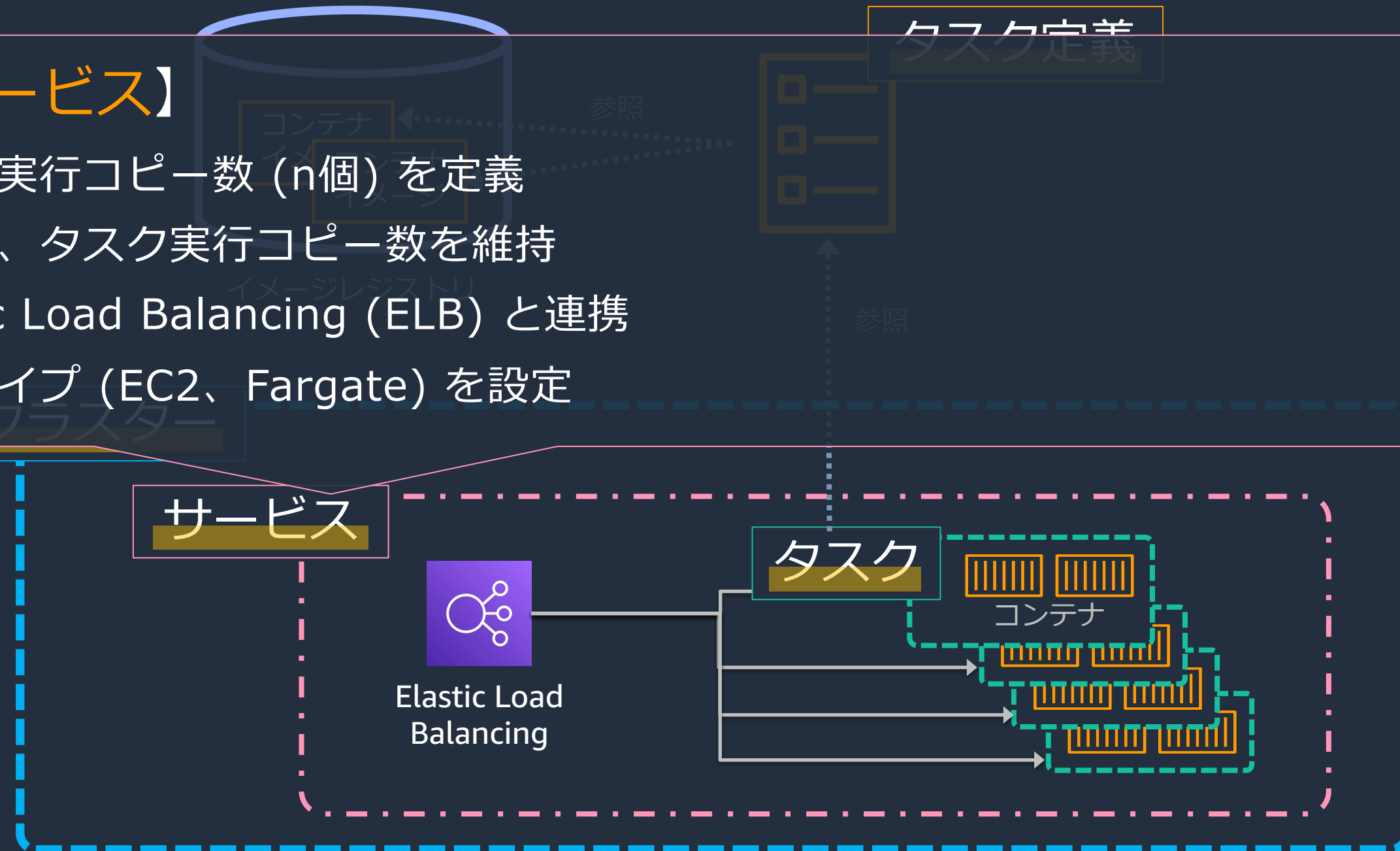
- タスク定義に基づき起動されるコンテナ群
- タスク内コンテナは同一ホスト上で実行される



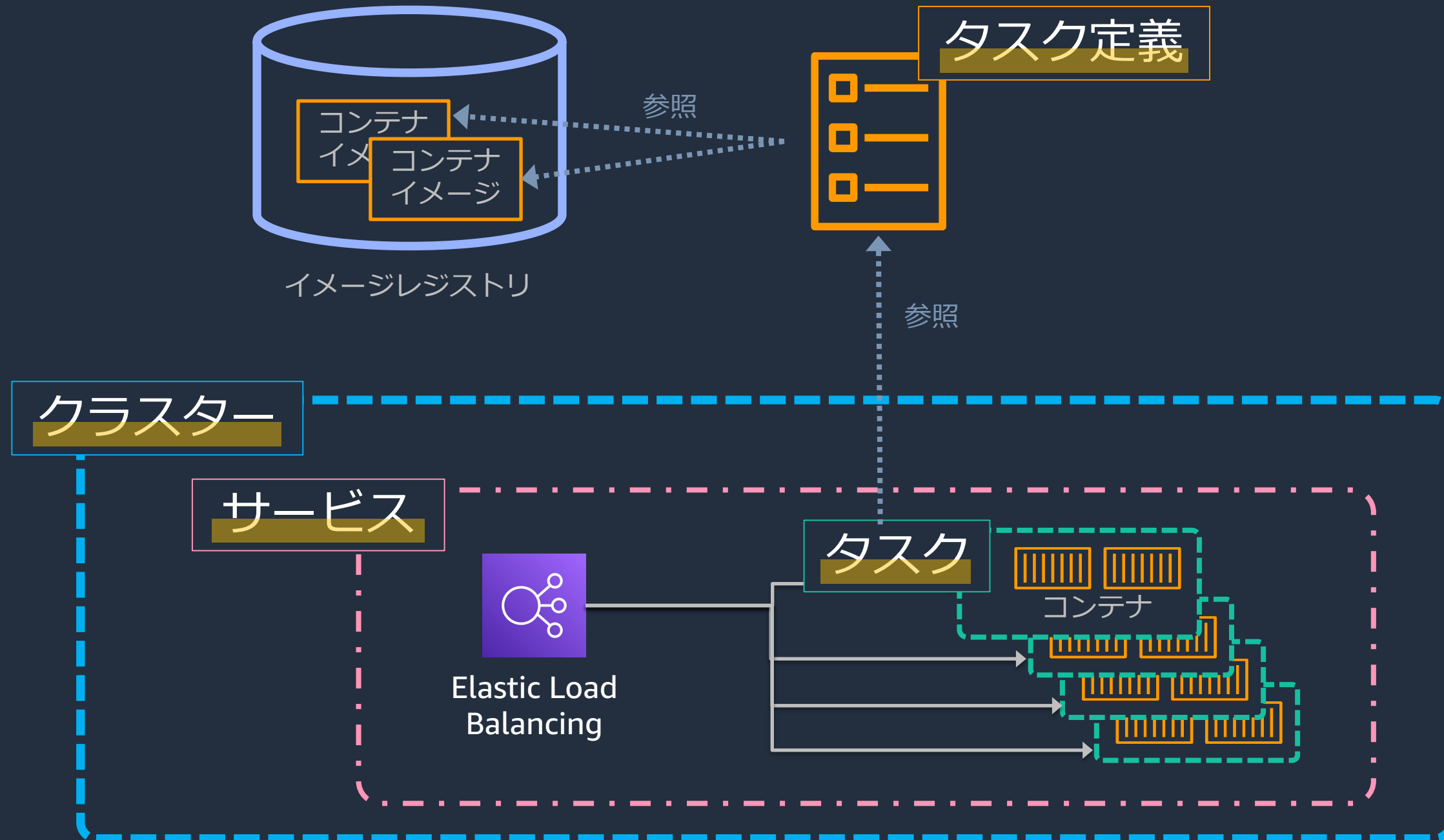
Amazon ECS の主要な 4 つの構成要素

4. 【サービス】

- タスク実行コピー数 (n個) を定義
- 起動後、タスク実行コピー数を維持
- Elastic Load Balancing (ELB) と連携
- 起動タイプ (EC2、Fargate) を設定



Amazon ECS の主要な 4 つの構成要素



Amazon ECS (on EC2) の特徴と課題

特徴

- コンテナホストを自由に選択、設定できる
- CPU、メモリ、ディスク、OS、バージョンなど
- ネットワークモードの選択 (外部接続しない、ホストのネットワークを利用、ENIへのアタッチなど)
- 柔軟なデータボリューム利用

課題

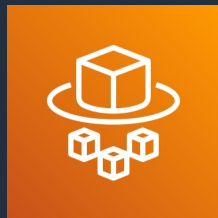
- **コンテナホストの管理が必要**
- OSやエージェント類へのパッチ当てや更新
- EC2インスタンス数のスケールリング

→ これら課題は
Amazon ECS on Fargate で解決

コンテナ実行場所への対応

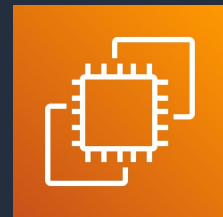
 AWS クラウド

Serverless



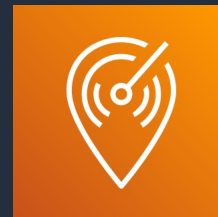
AWS Fargate

EC2 options



Amazon EC2

Edge and 5G



AWS Local
Zones

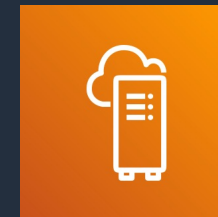


AWS
Wavelength



お客様データセンター

On-premises



AWS
Outposts



ECS Anywhere

Amazon Elastic Container Service (Amazon ECS)

再掲



パワフル & シンプル

- クラウドでコンテナを本番環境利用するための **オーケストレーター**
- 他の AWS サービスと高度に連携しコンテナを大規模に実行
- **フルマネージドなコントロールプレーン**
- 多様なワークロードをサポートする「タスク」「サービス」という **シンプルなリソース表現**

AWS コンテナサービスの公開ロードマップも是非ご覧ください

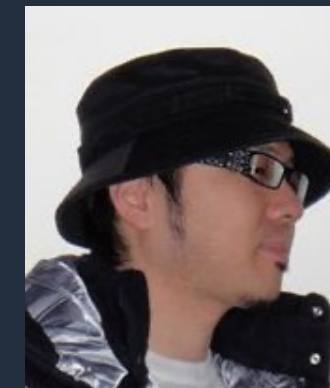
<https://github.com/aws/containers-roadmap>

The screenshot shows the GitHub repository for the AWS Containers Roadmap. The page is organized into columns representing different stages of development: Researching, We're Working On It, Coming Soon, Developer Preview, and Just Shipped. Each column contains a list of issues with their titles, issue numbers, and the status of the request. For example, in the 'Researching' column, there is an issue titled '[Fargate] [request]: Enhance the reliability of FireLens on Fargate' with issue number #700. The 'Just Shipped' column includes issues like '[ECS] [Deployment]: should update CloudFormation events and template when trigger deployment circuit breaker and rollback'.

本セッションの担当

杉本 晋吾 Shingo Sugimoto

技術統括本部 インダストリーソリューション部
ソリューションアーキテクト (SA)



略歴

現在 ← 海外開発製品の技術営業 ← ITコンサル会社のCTO
← 小さいソフトハウス経営 ← アパレル事業会社のCIO
← ECパッケージベンダーのCTO ← ネットプロバイダー管理者

好きなAWSサービス



AWS App Runner



AWS Fargate



Amazon DocumentDB