



コンテナとサーバレスの使い分け

AWS Black Belt Online Seminar

AWS Container Specialist
Principal Solution Architect

荒木靖宏

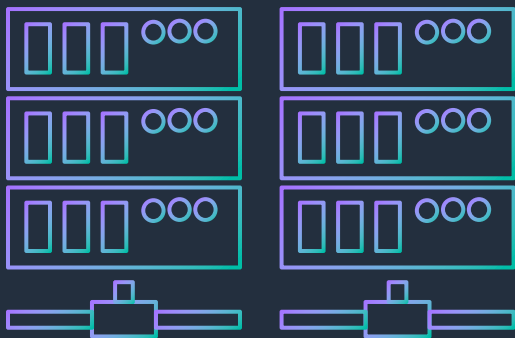
2021-July



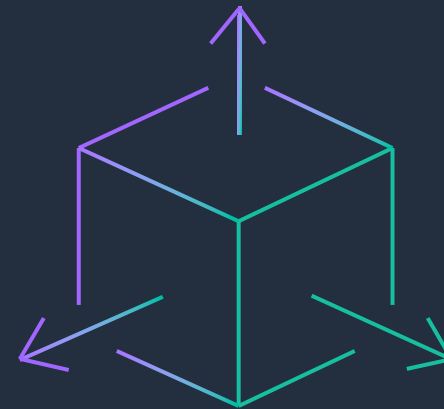
このセッションで扱うこと

- コンテナとサーバレスの特徴およびシステム構成
- 一般的なWebアプリにおける作り方比較
- コンテナとサーバレスどちらを使うかのディシジョンツリー

サーバーレスの特徴



サーバ管理が不要



柔軟なスケーリング



アイドル時のリソース確保が不要

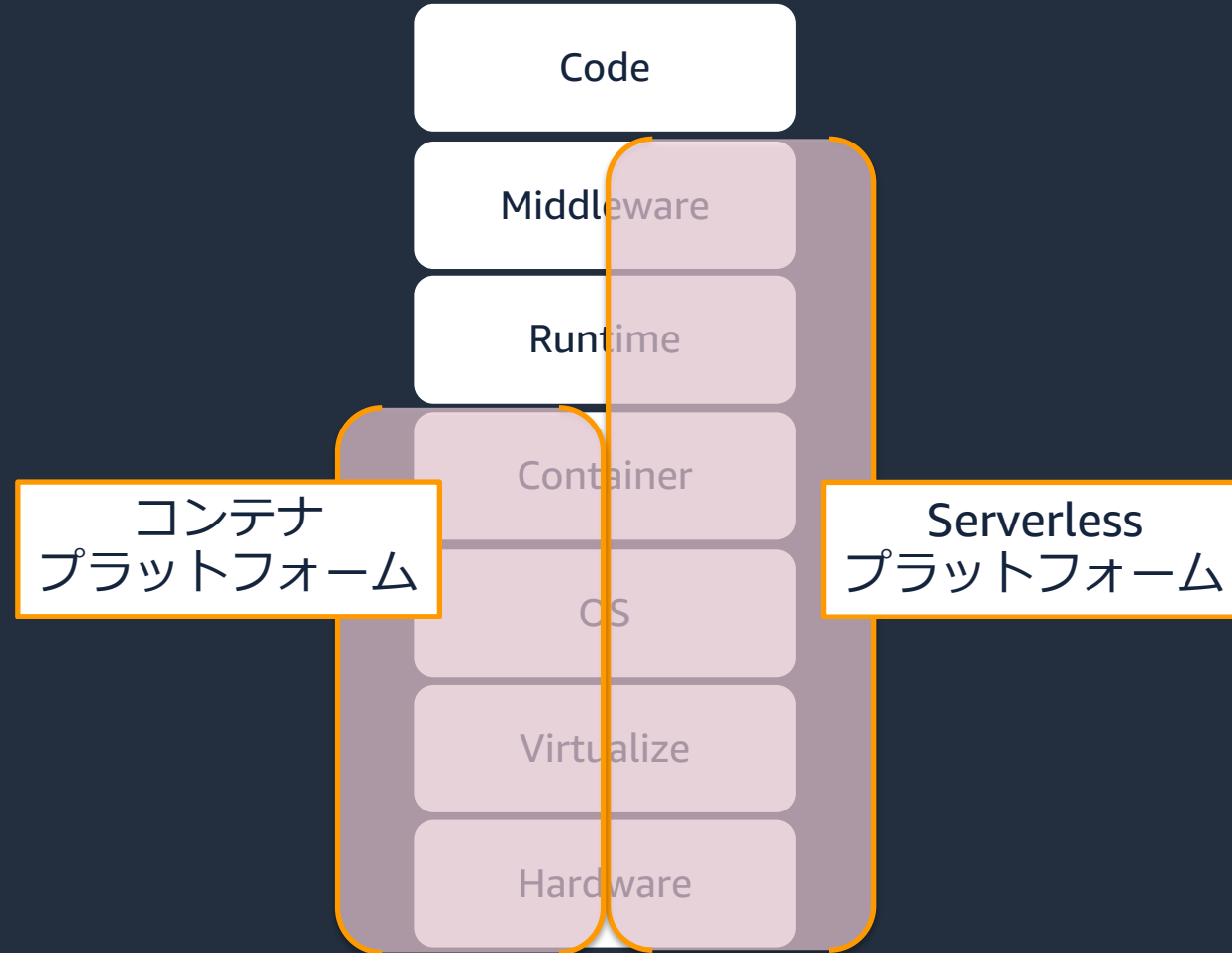


組み込まれた高可用性


















Source: [Cloud Native Application Security Report | Snyk](#)

サーバ・バックエンド



サービス構成のためのビルディングブロック（一部）

Network  Elastic Load Balancing	Storage  Amazon S3  Amazon EFS	Database  Amazon DynamoDB  Amazon RDS  Amazon Aurora
API Proxy  Amazon API Gateway  AWS AppSync	Streaming  Amazon Kinesis	
Authentication, WAF  Amazon Cognito  AWS WAF	Monitoring and Debugging  Amazon CloudWatch  AWS X-Ray	Messaging and Queues  Amazon SQS  Amazon SNS

一般的な3-tier WEB アプリケーション

プレゼンテーション層

アプリケーション層

永続層



ウェブ
サーバー

アプリケーション
サーバー

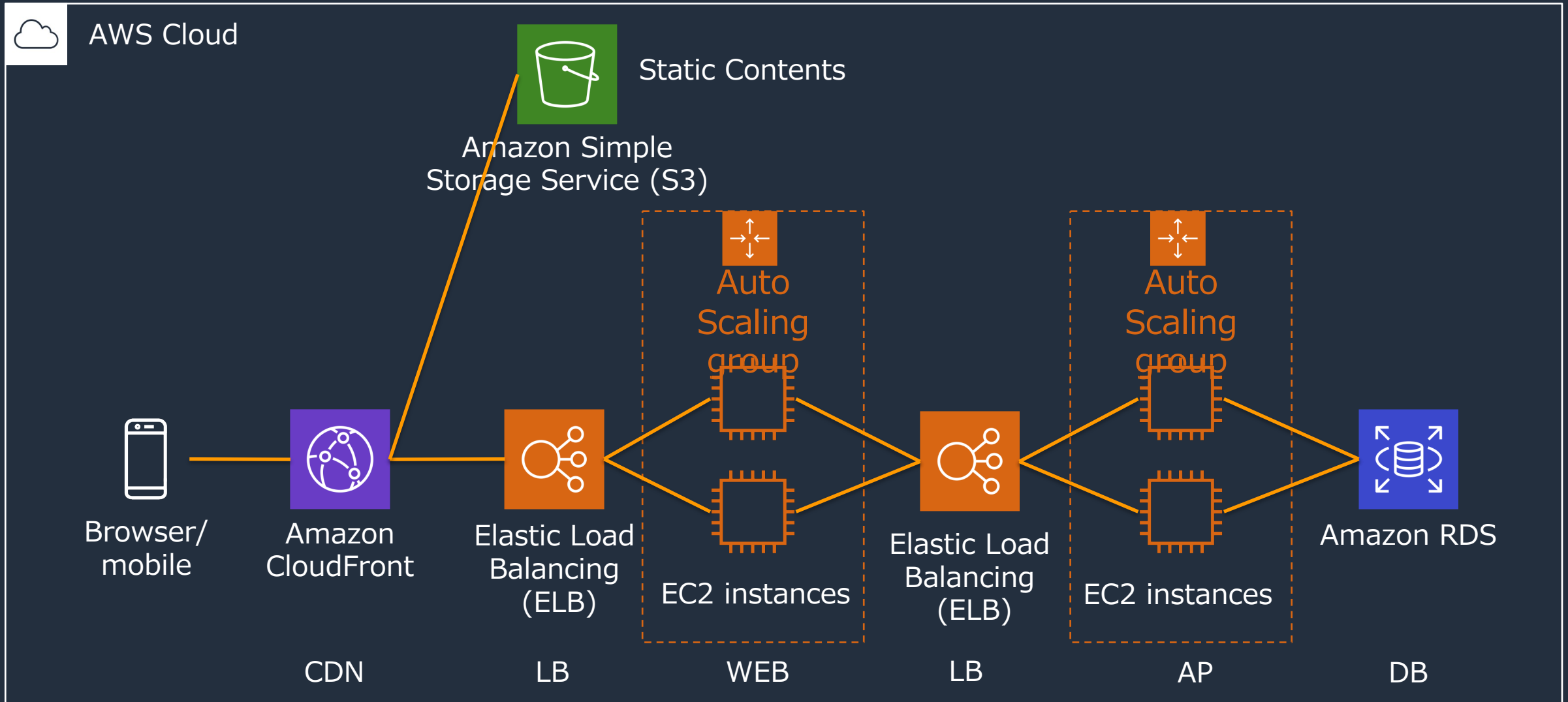
データベース
サーバー

Apache、
NGINX等

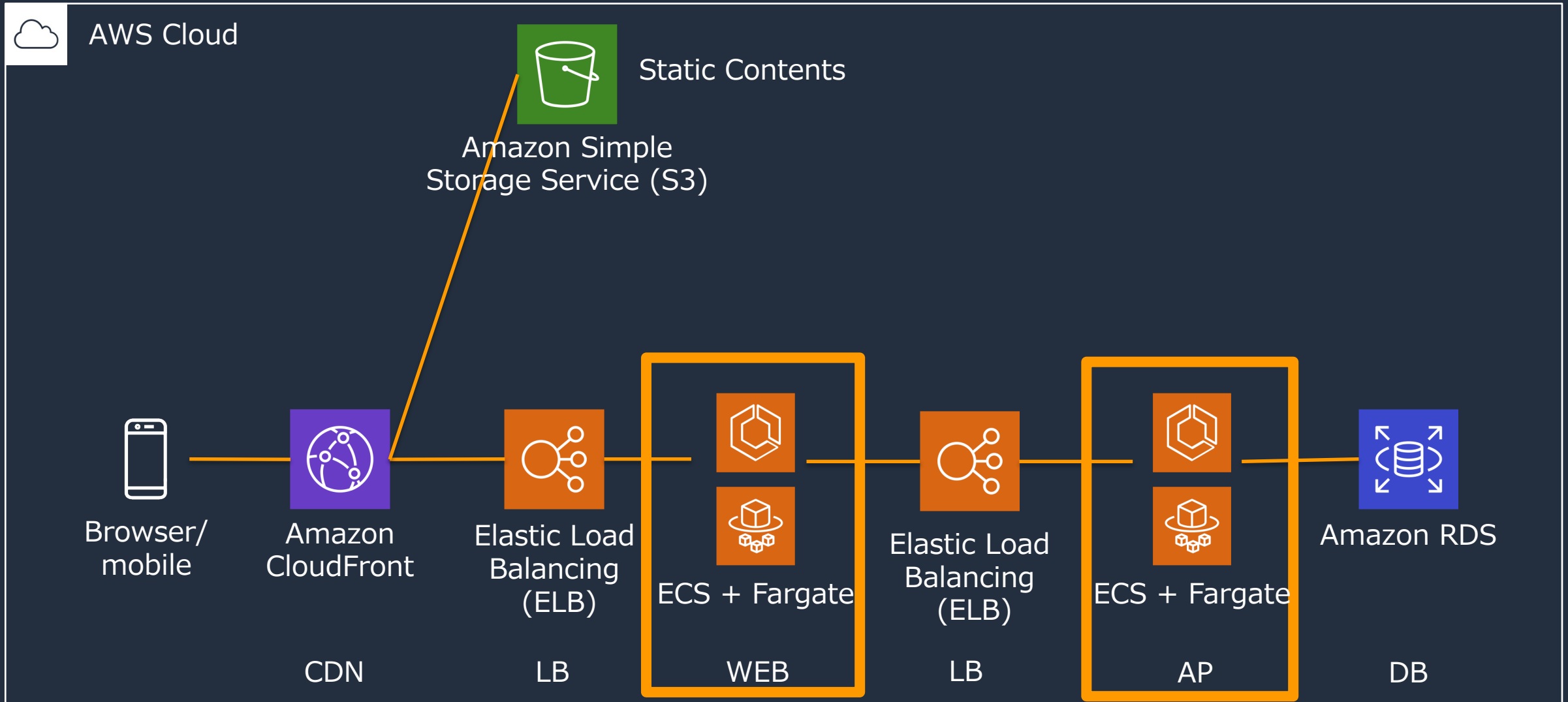
Tomcat、
Weblogic等

Oracle、
MySQL等

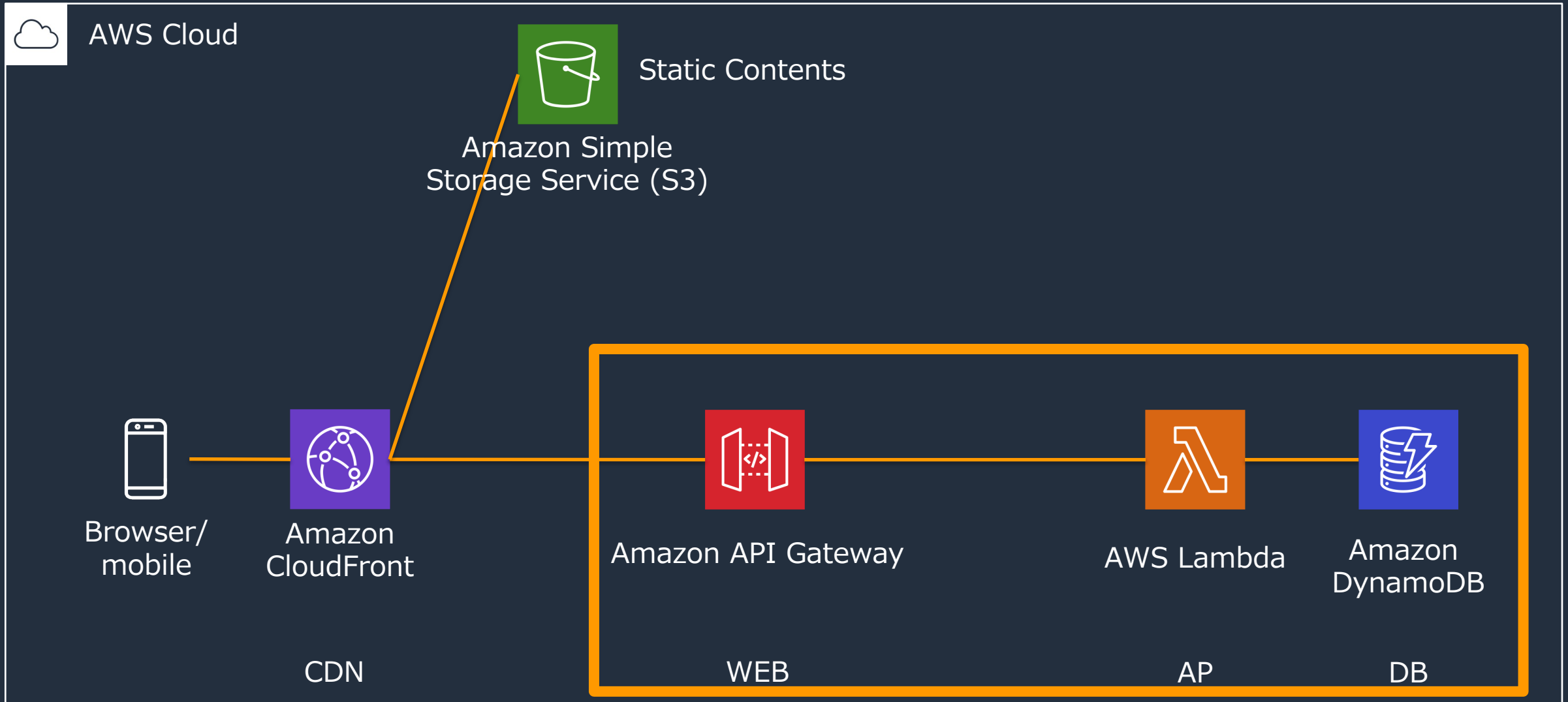
EC2(仮想サーバー)での構成例



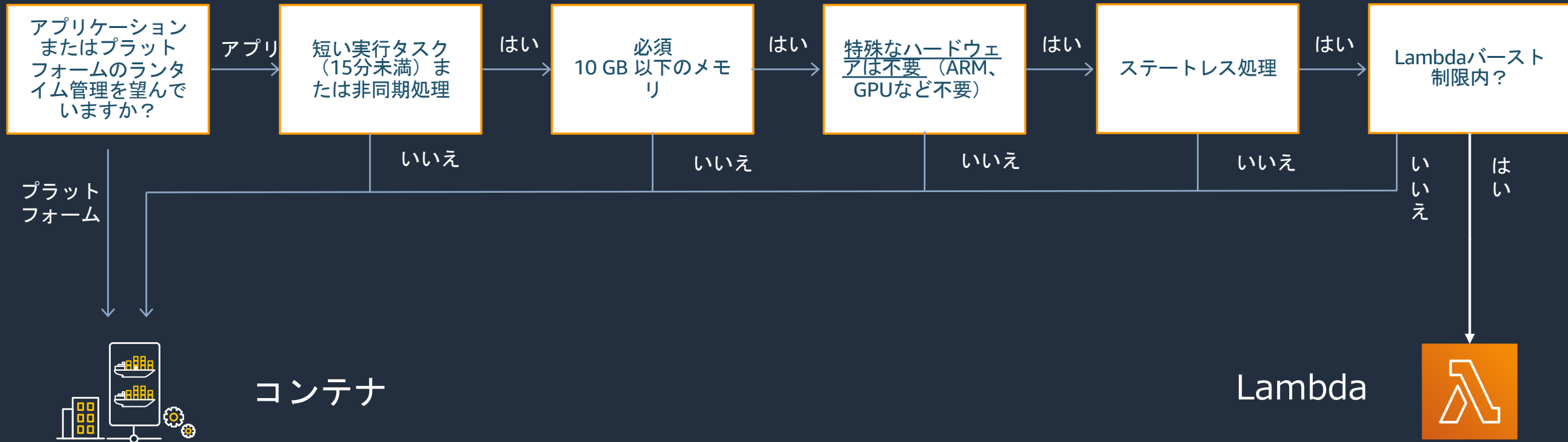
コンテナでの構成例



サーバーレスでの構成例



デシジョンツリー



このセッションで扱ったこと

- コンテナとサーバレスの特徴およびシステム構成
→AWSが管理運用する範囲が違う
- 一般的なWebアプリにおける作り方比較
→同じような構成もできるが制約もある
- コンテナとサーバレスどちらを使うかのディシジョンツリー
→適用条件を確認しましょう