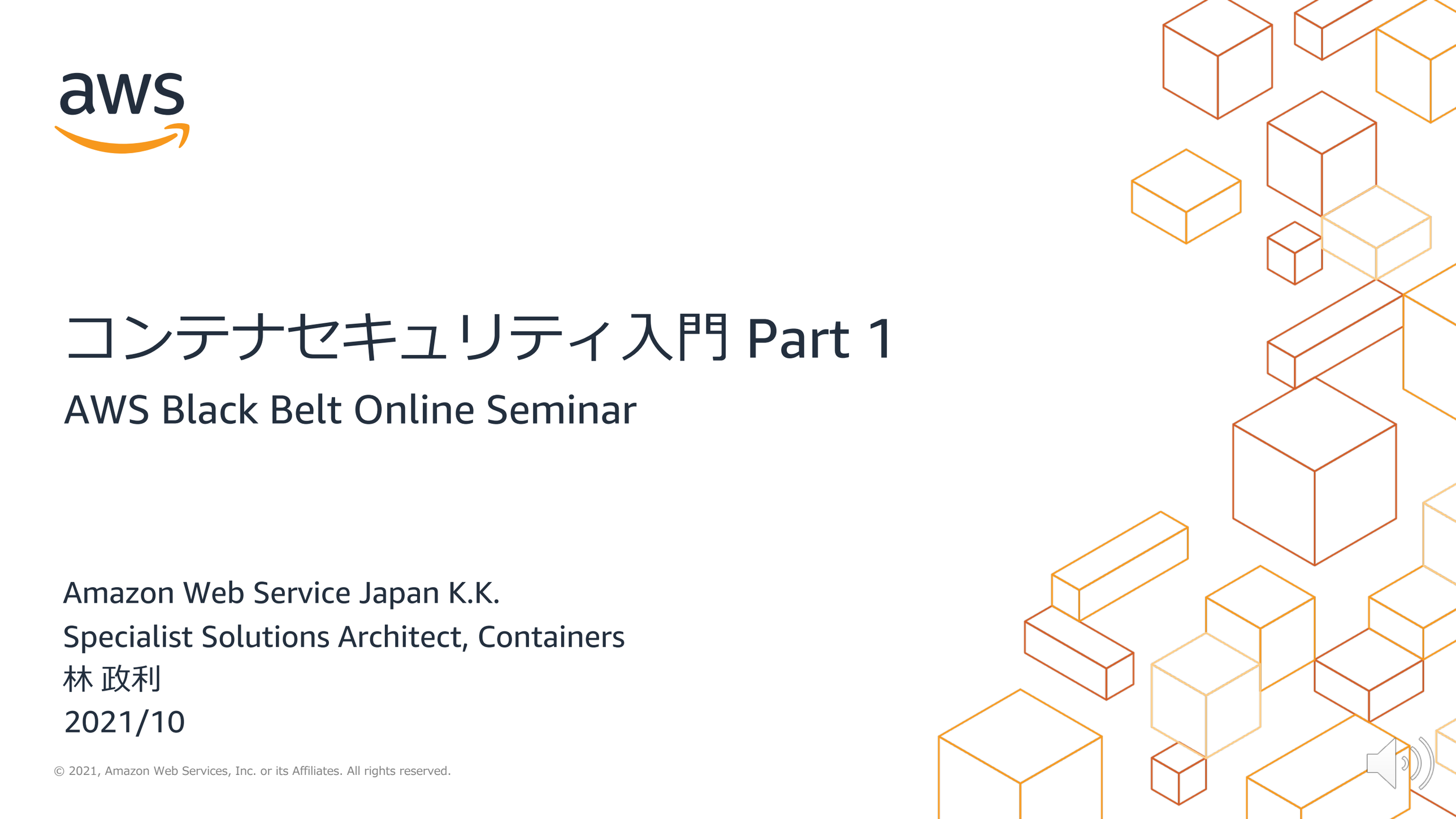


コンテナセキュリティ入門 Part 1

AWS Black Belt Online Seminar

Amazon Web Service Japan K.K.
Specialist Solutions Architect, Containers
林 政利
2021/10

このセッションで扱うこと・学べること

コンテナについての一般的なセキュリティの考え方

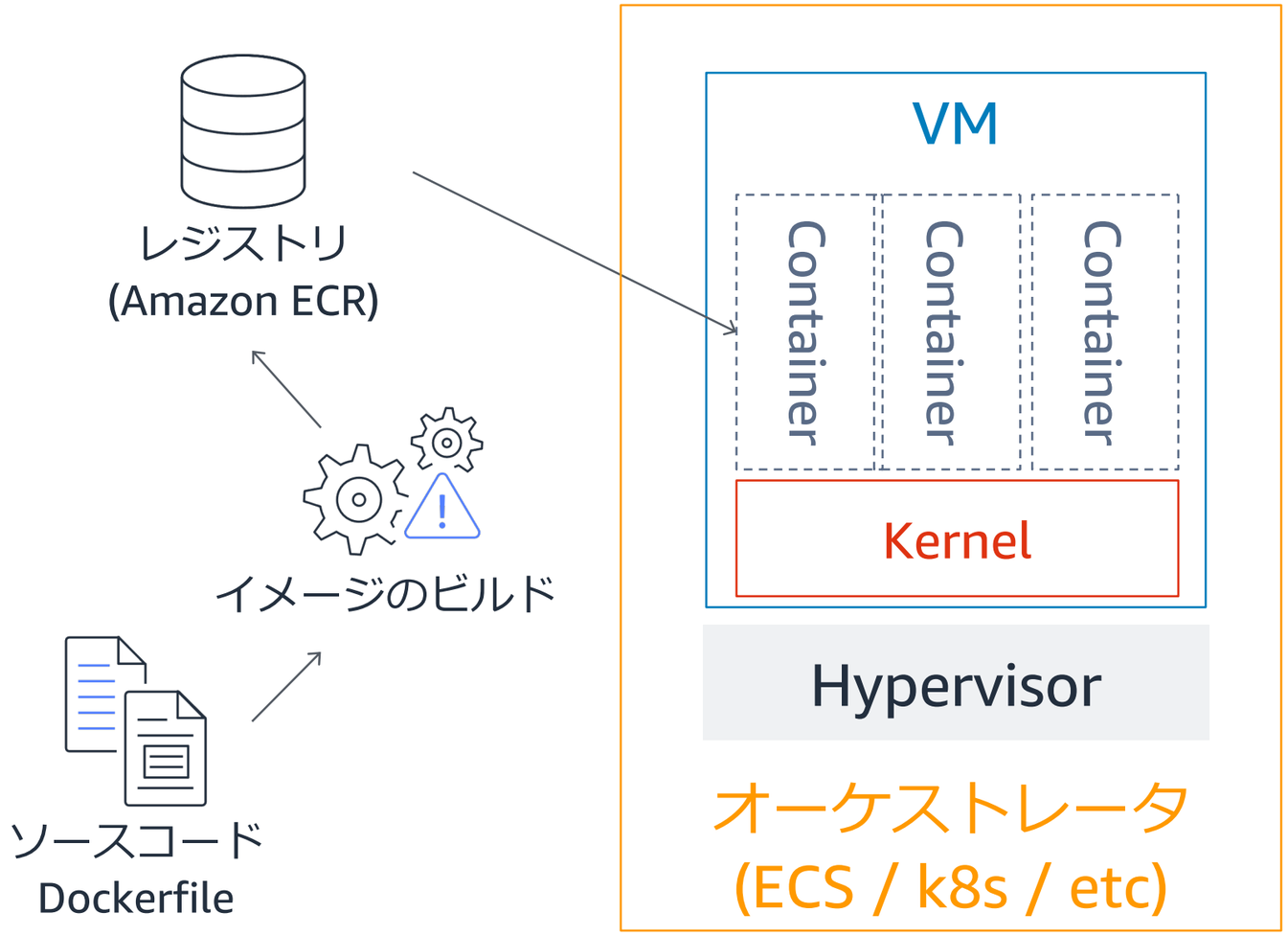
実運用でのガイダンス、指針の整理

セッションに関連するAWSのコンテナ関連サービス

- Amazon Elastic Container Service (Amazon ECS)
  - 完全マネージド型のコンテナオーケストレーションサービス
- Amazon Elastic Kubernetes Service (Amazon EKS)
  - Kubernetes を簡単に実行できるマネージド Kubernetes (k8s) サービス
- AWS Fargate
  - コンテナ向けサーバーレスコンピューティングエンジン
- Amazon Elastic Container Registry (Amazon ECR)
  - 完全マネージド型のコンテナイメージレジストリ

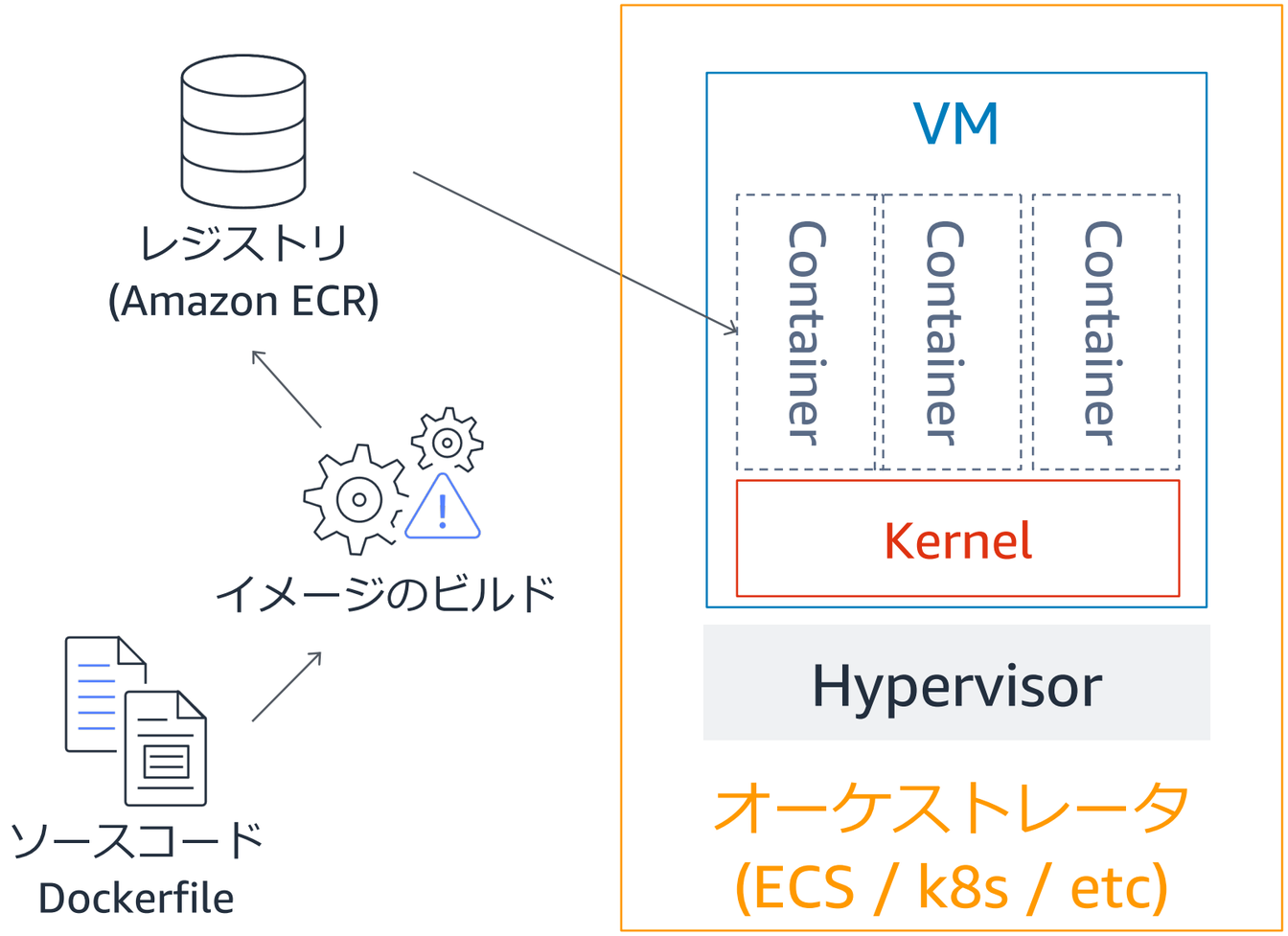
アジェンダ

コンテナのライフサイクルで考えるリスクの場所と対策



- イメージの開発とビルド
- サプライチェーン
- オーケストレータの保護
- ホストのセキュリティ
- 実行時のセキュリティ

アジェンダ



- イメージの開発とビルド
- サプライチェーン
- オーケストレータの保護
- ホストのセキュリティ
- 実行時のセキュリティ

セキュアなイメージの開発

セキュリティに関する Dockerfile ベストプラクティス

- コンテナへの攻撃面
  - 信頼できるレジストリの信頼できるイメージのみを使う
  - 実行時にパッケージをインストールしない
  - イメージを小さく保つ
    - 小さなベースイメージを使う
    - Multi Stage Build の活用する
- コンテナの実行権限
  - USER を指定する
    - 未指定の場合、コンテナが root で実行される
- シークレットの取り扱い
  - コンテナイメージに埋め込まない
- Lint ツールの利用
  - Haskell Dockerfile Linter (hadolint)
  - dockerfile-lint

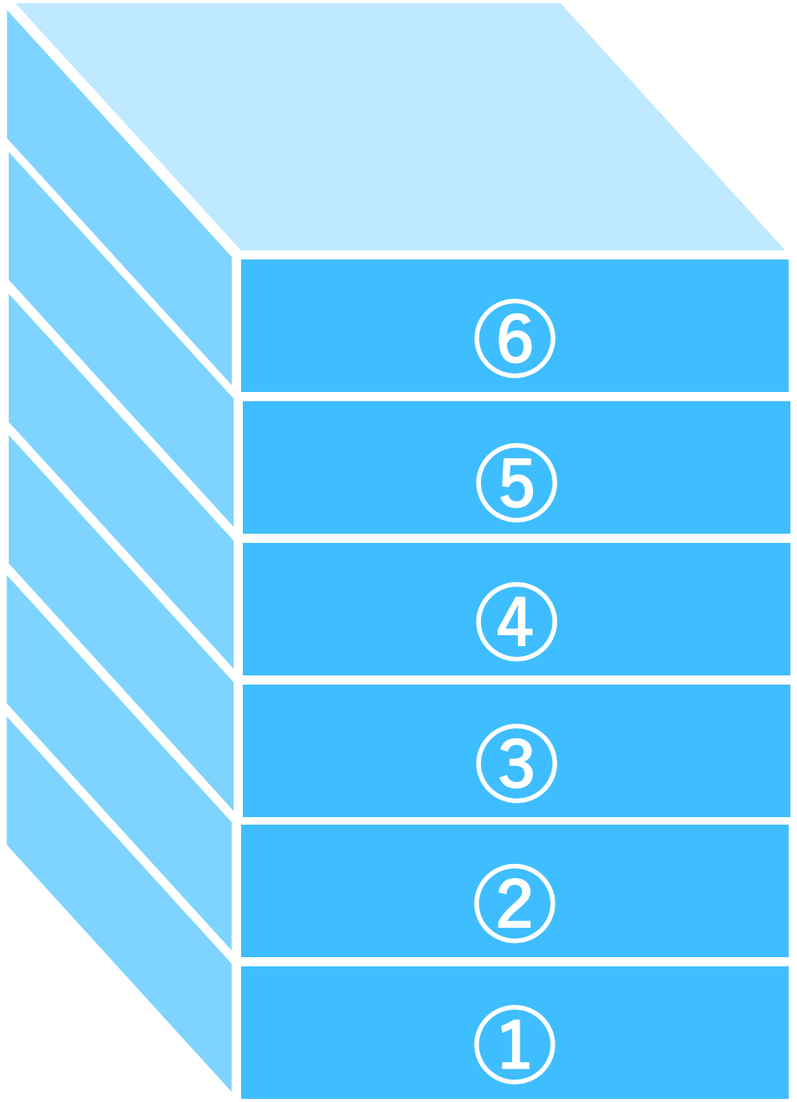
コンテナイメージの構造とシークレット

レイヤにシークレットを埋め込まない

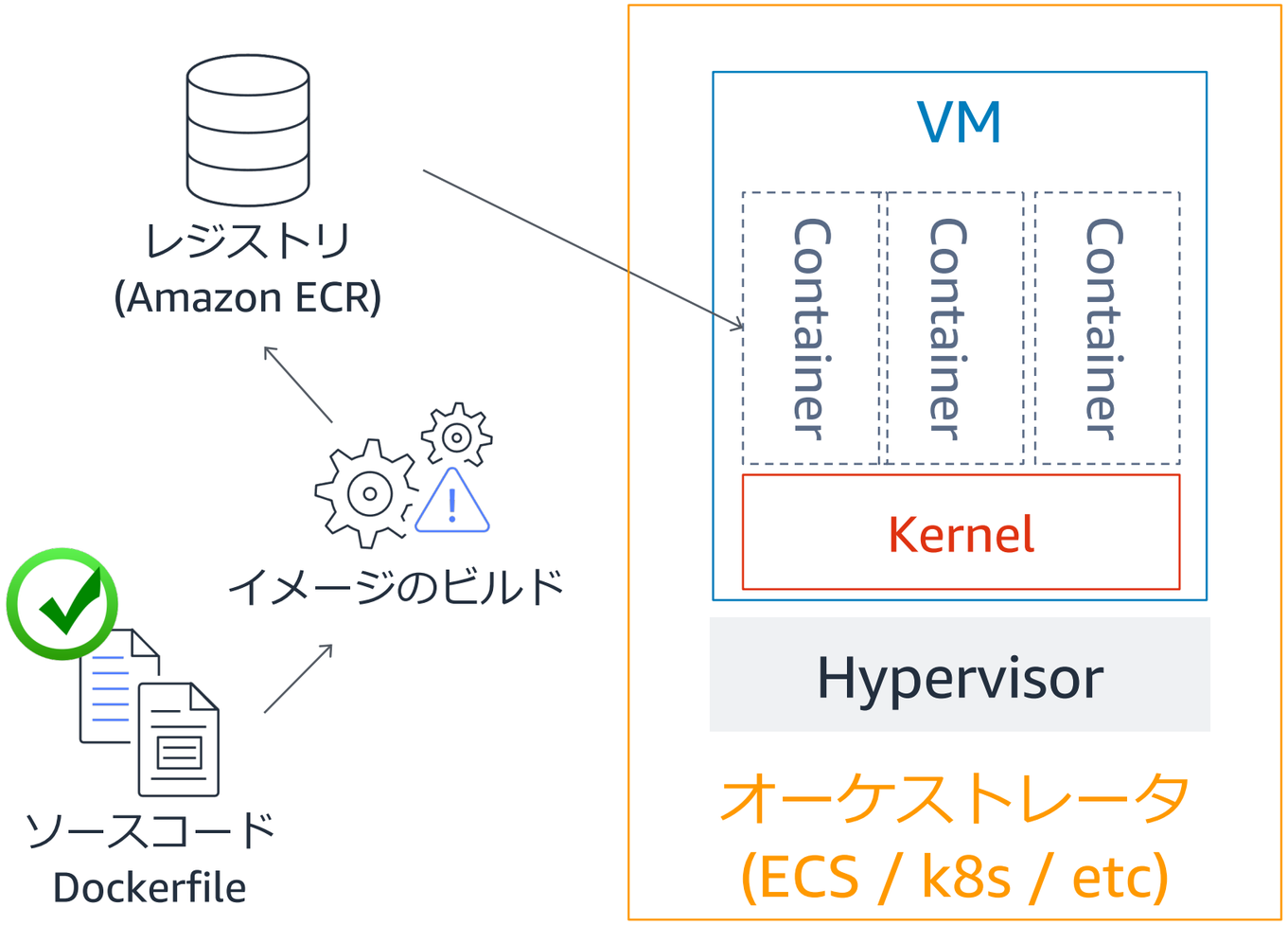
```
FROM ubuntu:20.04

# レイヤに保存する
RUN echo "top-secret" > /pswd.txt

# 上で保存されたレイヤからは削除されない
RUN rm /pass.txt
```



アジェンダ

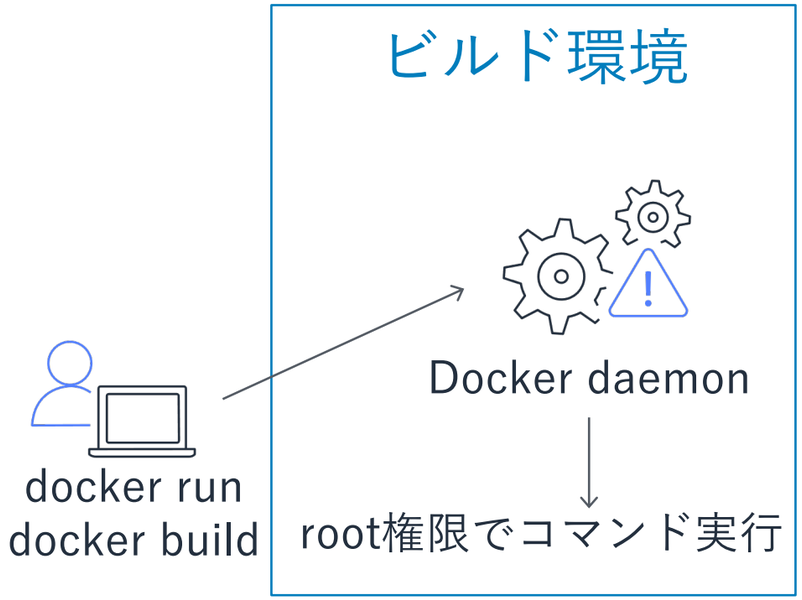


- イメージの開発とビルド
- サプライチェーン
- オーケストレータの保護
- ホストのセキュリティ
- 実行時のセキュリティ

ビルド時のセキュリティ

Docker ビルドする環境を防御

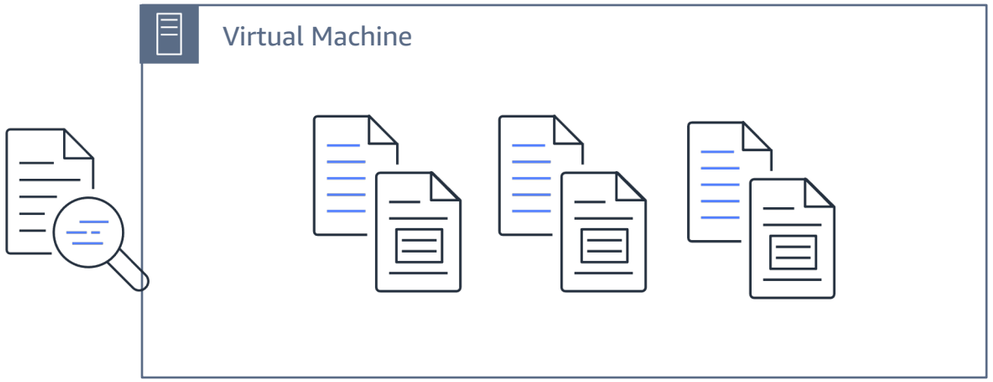
- Docker daemon は長期間稼働する root のプロセス
  - Rootless モードも GA したが、実環境での実績作りはこれから
- docker run できるユーザーは実質的にrootでホストマシンを操作できる
- ビルドマシンを攻撃され、ビルドプロセスに介入されるリスク
- root 権限を使用せずにイメージをビルドする選択肢
  - BuildKit
  - Buildah, kaniko
- そもそもビルド環境の管理をしない
  - ビルド用の SaaS
  - AWS CodeBuild



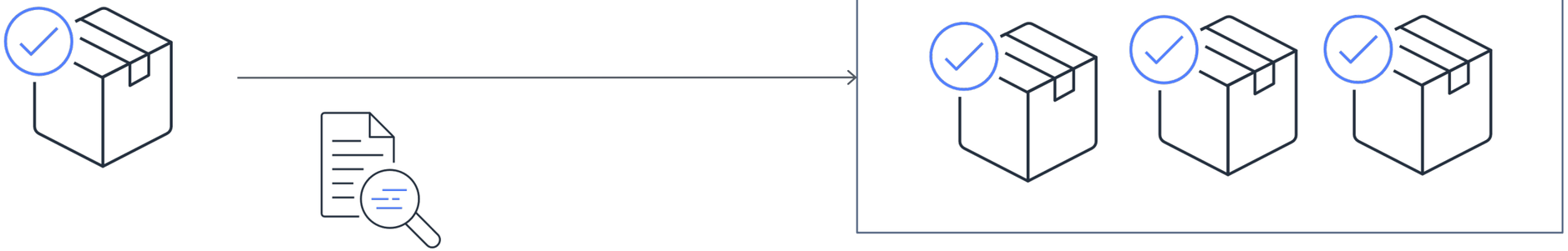
脆弱性対策、コンテナ環境と Shift Left

コンテナでは仮想マシンよりもパイプラインの「前で」脆弱性対策が可能

アプリケーションの実行環境で
パッチを適用

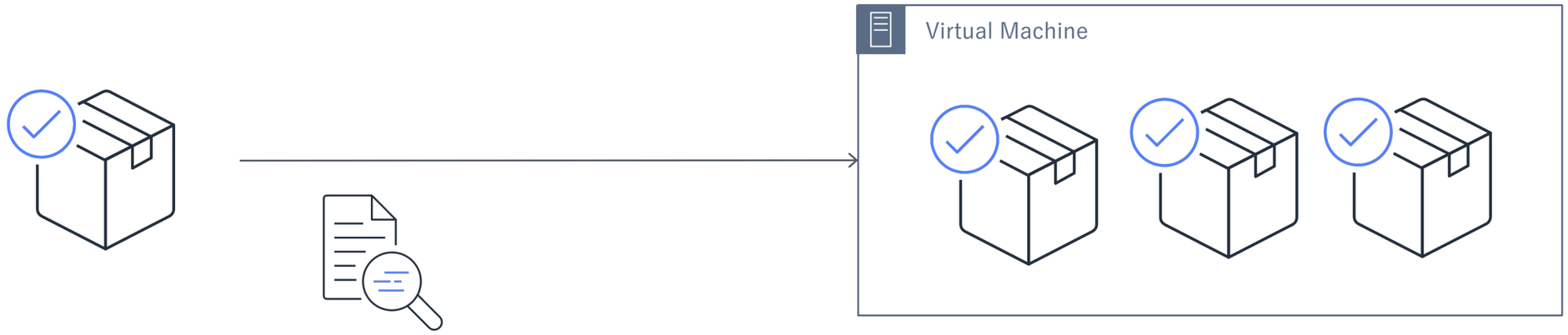


イメージをスキャンして配布



コンテナ環境の脆弱性対応ルール

コンテナがイミュータブルで、かつ定期的にスキャン、デプロイすることが重要



イミュータブル

スキャンしたイメージと同一のものが実行されていること

定期的にスキャンしてデプロイ

セキュリティパッチが適用されたイメージをデプロイ

脆弱性スキャンの注意点

脆弱性がスキャンできない、または誤って検出されるケース

未対応ディストリビューション

脆弱性スキャンの情報ソースとしてディストリビューションの
セキュリティアドバイザリが利用される

OS パッケージのみスキャンされる

Node.js や Java などのライブラリは Trivy など一部のツール
のみが対応している

未知の脆弱性、誤検出、ソースの情報不足

未知の脆弱性、偽陽性、偽陰性

スキャンを絶対視しない、小さいシンプルなイメージを作ること

多層防御も考える

このセッションで扱うこと・学べること

コンテナについての一般的なセキュリティの考え方

セキュアなイメージの構築方法

本セッションの担当: 林　政利

Specialist Solutions Architect, Containers

好きなサービス

Amazon Elastic Kubernetes Service (Amazon EKS)
AWS Certificate Manager

SIer

フリーランス
受託開発

Web系企業
サービス開発・運用

外資ベンダー
コンテナ担当SA・
サポートエンジニア

AWS Japan
コンテナ担当SA