

Dockerイメージサイズの最小化

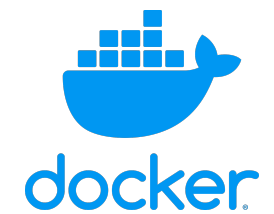
研究発表

作島 大智

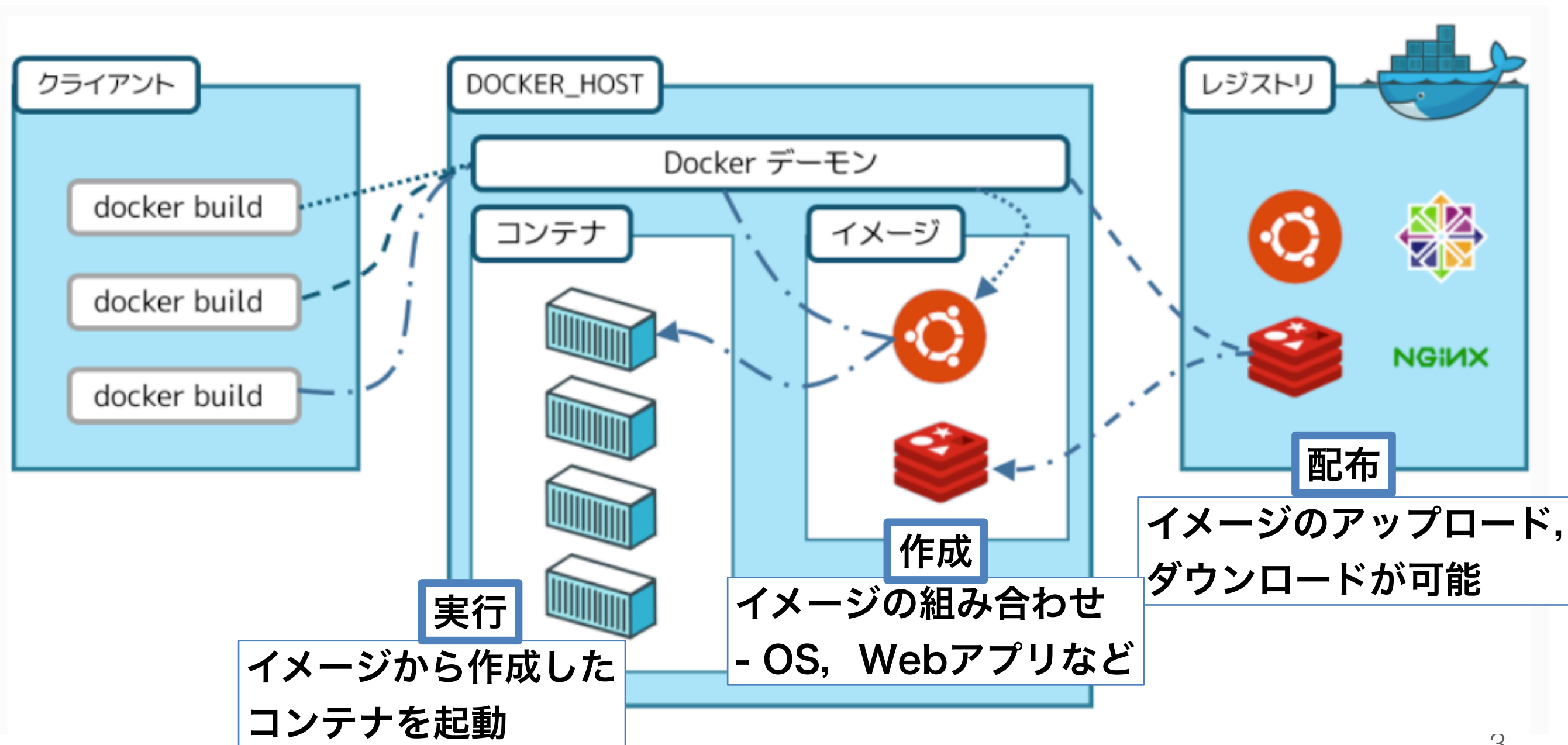
背景

- 分散型アプリケーションやインフラは仮想マシン(VM)中心からコンテナ中心(Docker)に移行している
 - 管理が容易である
 - VMと比較して起動・再起動・停止が高速
- Dockerのパフォーマンスやランタイム適応, セキュリティなどに関して研究が多く行われている

Dockerとは



軽量なコンテナ型の仮想環境を作成・配布・実行
するためのプラットフォーム



Dockerの利点と課題

利点

- どこでも誰でも
同じ環境が作れる
- 作成した環境を
配布しやすい
- Dockerイメージの
作成・破棄が容易

課題

- 増大するビルド時間
- **大きいイメージサイズ**
- 保守性
- 安全性
- 一貫性・反復性

研究目的

目的

dockerイメージサイズを最小限に抑える

イメージサイズが小さいことの利点

- ホストのディスク容量を圧迫しない
- イメージの配布が気軽に行える
- 生涯発生リスクが軽減される

イメージサイズが大きくなる要因

イメージの選択

- ・ 選択しているベースイメージサイズが大きい
- ・ 不要な依存関係(推奨, 不要なパッケージ)がある

dockerfileの記述

- ・ RUNをまとめていない
- ・ マルチステージビルドを行っていない
- ・ 不要なパッケージマネージャのキャッシュがある

仮説

仮説

必要以上にサイズの大きいベースイメージを利用しているのではないか？

Alpine	4.8 MB
CentOS	196.8 MB
Ubuntu	122 MB
Debian	125.1 MB

アプローチ

サイズの小さいベースイメージに変更

- Linux Alpineで置換

必要であればパッケージ等を追加・置換

- 必要になるパッケージの判定に関しては未定

Dockerfileの記述を最適化

- マルチステージビルドを使用
- RUNをまとめる

評価

- イメージサイズを小さくする性能
- 提案手法が適用できるプロジェクトの割合

現在していること

- 手作業で大小のプロジェクトのベースイメージを Alpine に置換する
- ビルド・テストが成功するのを確認する
- 失敗する場合は、パッケージ等を追加する程度で成功するようになるのを確認する

予備資料

ハイパーバイザー型とコンテナ型

