

PAR-26

AWS Graviton2 環境で AWS Fargate に CircleCI で CI/CD しよう ～ ARM64 イメージをビルド、デプロイする

舟木 将彦

CircleCI 合同会社

Principal Developer Advocate



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

本セッション資料や記載内容については一切の転用を禁止しております

Agenda

- 1 Fargate on Graviton2 を活用した「できあがり」像
今回実現したいゴールの共有
- 2 コンテナイメージのビルド、レジストリへのプッシュ自動化
ビルド、テスト、リリース（登録）の自動化
- 3 CloudFormation を使った Fargate 構築、自動デプロイ
インフラストラクチャーの構築とデプロイの自動化
- 4 まとめ
Graviton 2 環境 + CircleCI による自動化のメリット

Fargate on Graviton2 を 活用した「できあがり」像

今回実現したいゴールの共有



AWS Fargate とは

サーバーや Amazon EC2 インスタンスのクラスターを管理することなく
コンテナを実行。サーバレスで従量制料金のコンピューティングエンジン。

やらなくていいこと

コンテナ実行のための仮想マシンの
プロビジョニング、設定、スケール

サーバタイプの選択

クラスターをスケールするタイミングの決定

クラスターのパッキングの最適化

やるべきこと

アプリケーションをコンテナにパッケージ化

CPU とメモリ要件を指定

ネットワークと IAM ポリシーを定義

アプリケーションを起動

https://docs.aws.amazon.com/ja_jp/AmazonECS/latest/userguide/what-is-fargate.html を要約

「できあがり」までのプロセス

- ① 子カバ画像をネットから収集・表示する React アプリを開発し、コンテナイメージを作成
- ② 作成したイメージを Amazon Elastic Container Registry (Amazon ECR) にプッシュ
- ③ AWS CloudFormation を使って定義した AWS Fargate を構築
- ④ 作成したイメージを Graviton2 上で動作する Fargate タスクにデプロイ

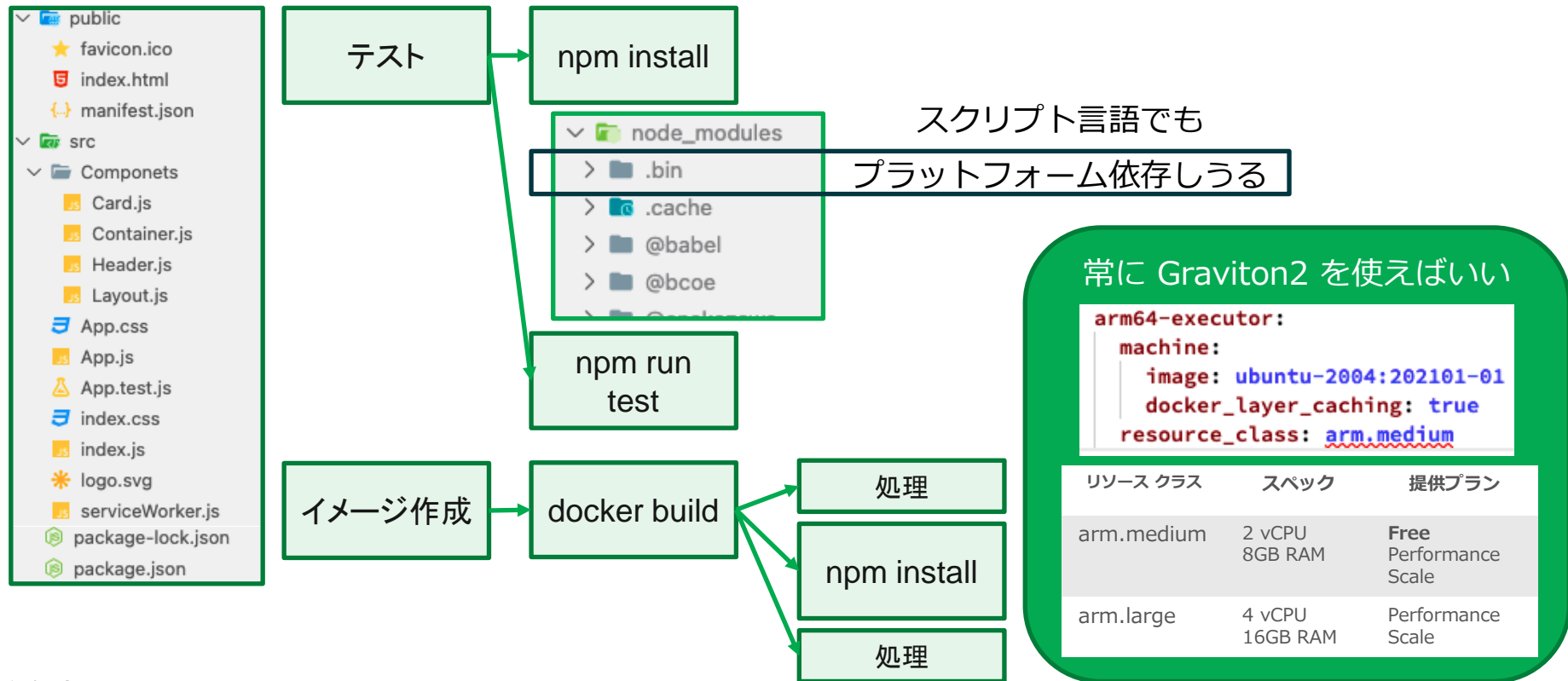
以上を CircleCI を使って AWS Graviton2 搭載環境で自動化します

コンテナイメージのビルド、 レジストリへのプッシュの自動化

ビルド、テスト、リリース（登録）の自動化



React アプリ (Node.js) のコンテナイメージ作成



CloudFormation を使った Fargate 構築、自動デプロイ

インフラストラクチャーの構築とデプロイの自動化



CircleCI から CloudFormation を使用

Orb とは

再利用 および 共有が可能な
YAML 設定ファイルパッケージ

この cloudformation Orb は
frameio さんが開発、公開
(誰でも開発、組織内／外に公開可能)

lint (cfn-lint を使用) や変更セットの作成、
スタックの更新が、必要なパラメタを指定
するだけで可能



The screenshot shows the CircleCI Orb Developer page for the `frameio/cloudformation@0.25.1` orb. The page is titled "circleci デベロッパー" and includes navigation links for "ホーム", "イメージ", and "Orbs". The "Orbs" section is active, showing the orb's details. The left sidebar lists sections: "Orb Quick Start Guide", "Usage Examples" (with sub-items: `changeset-approval-deploy`, `deploy`, `lint`), "Jobs" (with sub-items: `changeset`, `deploy`, `lint`), "Commands" (with sub-items: `deploy`, `lint`), "Executors" (with sub-item: `alpine`), and "Orb Source". The main content area displays the orb's name, version, and a description: "AWS Cloudformation CircleCI Orb." It also shows metadata: "Created: January 18, 2020 | Version Published: February 16, 2022 | Releases: 43 | Org Usage: < 25" and a homepage link: <https://github.com/Frameio/frameio-orbs>. Below this, there is a "See Orb Licensing" link. The "Orb Quick Start Guide" section provides two steps: 1. Use CircleCI version 2.1 at the top of your `.circleci/config.yml` file. 2. Add the orbs stanza below your version, invoking the orb. The guide includes code snippets for the `version` and `orbs` sections of the config file.

circleci デベロッパー ドキュメント circleci.com

ホーム イメージ **Orbs**

Orb Quick Start Guide

Usage Examples

- `changeset-approval-deploy`
- `deploy`
- `lint`

Jobs

- `changeset`
- `deploy`
- `lint`

Commands

- `deploy`
- `lint`

Executors

- `alpine`

Orb Source

Orbs > **frameio/cloudformation@0.25.1**

frameio/cloudformation@0.25.1

AWS Cloudformation CircleCI Orb.

Created: January 18, 2020 | Version Published: February 16, 2022 | Releases: 43 | Org Usage: < 25

Homepage: <https://github.com/Frameio/frameio-orbs>

[See Orb Licensing](#)

Orb Quick Start Guide

- Use CircleCI version 2.1 at the top of your `.circleci/config.yml` file.

```
version: 2.1
```

- Add the orbs stanza below your version, invoking the orb:

```
orbs:
  cloudformation: frameio/cloudformation@0.25.1
```

まとめ

Graviton 2 環境 + CircleCI による自動化のメリット



Graviton 2 環境 + CircleCI による自動化のメリット

Graviton 2 : コストパフォーマンスの高い動作環境

Fargate : コンテナアプリケーションをスケーラブルに構築可能

CloudFormation : AWS リソースを YAML ベースで定義可能（変更可能）

CircleCI :

- AWS リソースの変更・更新
- AWS で実行するアプリケーションのビルド、テスト、デプロイを自動化することで、
- アプリケーションの開発により注力可能 + グループでの開発を効率化し、開発者体験（DX: Developer eXperience）を向上

Thank you!

舟木 将彦

CircleCI 合同会社
Principal Developer Advocate

