

ストリーミングETL 処理基盤における Graviton2 への移行事例

2022/10/13

株式会社NTTドコモ

田中・矢吹

》本日のトピックス

- ドコモにおけるビッグデータ分析/活用
- ストリーミングETL処理基盤と課題
- Gravitonへの移行事例

》自己紹介

たなか まさひろ
田中 正浩

株式会社NTTドコモ
サービスイノベーション部
ビッグデータ担当

業務: ストリーミングETL処理基盤の開発・運用

<宣伝> NTTドコモ R&D組織 ブログ
技術ブログ

- [ドコモ開発者ブログ](#)

》自己紹介

やぶき けんじ
矢吹 健二

株式会社NTTドコモ
総務人事部
(NTTコミュニケーションズ出向中)

業務: DevOps基盤 Qmonus Value Streamの開発・運用

<宣伝> Qmonus Value Streamの取り組みについて

対外発表

- ["クラウドを最大限活用するInfrastructure as Codeを考えよう", Cloud Native Days Tokyo 2021](#)
- ["社内DevOps基盤、Tekton、Cuelang", Fukabori.fm episode 50](#)
- ["Deliver Your Cloud Native Application with Design Pattern as Code", KubeCon Europe 2020](#)

技術ブログ

- [連載：DevOpsプラットフォームの取り組み](#)

ドコモにおける ビッグデータ分析/活用

ドコモの事業領域

通信事業

通信サービスの提供

携帯電話等の
販売・修理

5G通信

スマートライフ事業

コンテンツ・コマース

金融・決済

ライフスタイル

法人ソリューション

→ 大量・多様なデータが生成

》》 ドコモのデータ分析・活用基盤

ドコモの膨大なデータを、専門家から一般社員まで、
効率的/横断的に分析・活用できる基盤



- データサイズ 数PB
数十TB/day
- 分析者 数百名
開発運用者 2pizza x 2チーム
- オンプレ+クラウド
Internet経由でのアクセス 🙋

ストリーミングETL処理基盤 と課題

》》 ストリーミングETL処理基盤

- ドコモ社内からデータを収集する基盤
- (ニア)リアルタイムにデータを処理し、後段のシステムへとデータ連携する



》課題

- とにかくサーバーがいっぱいなためコストが高い

» コスト削減の取り組み

下記3つの取り組みを実施

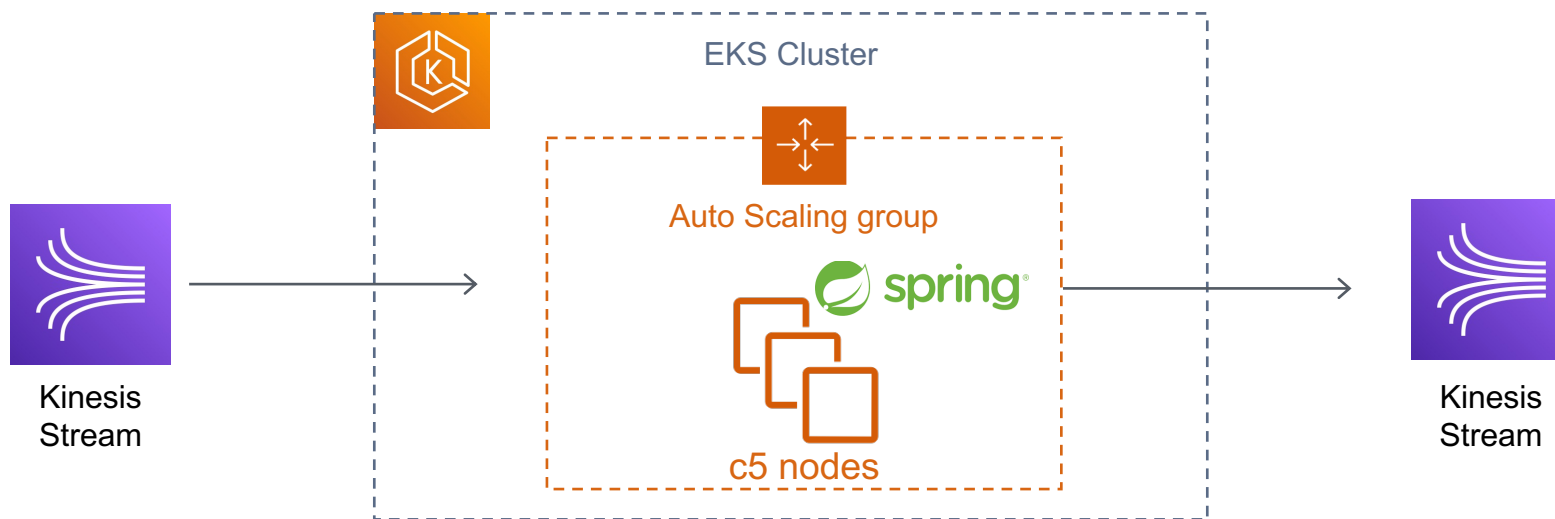
- アプリケーションロジック・アーキテクチャの改良
- **Gravitonインスタンスへの移行**
- Reserved Instances/Savings Plansの活用

コスト削減の取り組みの1つとして行った
Gravitonインスタンスへの移行事例について紹介

Gravitonへの移行事例

》移行対象のシステム

- EKSクラスター上に展開されたJavaアプリケーション
 - アプリケーションは複数のエリア毎に分割
 - QA・パフォーマンステスト用のEKS Clusterも存在
- c5.4xlarge (vCPU:16, メモリ: 32GiB) × 100台の規模



》 移行の流れ

- arm用コンテナイメージの作成
- パフォーマンステスト
- Node Groupの拡張とデプロイ

》arm用コンテナイメージの作成

- Gravitonインスタンスで動作可能なコンテナイメージに更新
 - アプリケーションのベースイメージをarm対応のものに更新
 - 必要に応じて内部で使用しているライブラリ等も更新
- 切り戻しなどの運用を考慮して、**Multi-arch Image**としてビルド

TAG

[13-jdk-slim](#) ? Not scanned for Log4Shell

Last pushed 3 years ago by [doijanky](#)

DIGEST

[dd65efeba8b2](#)

TAG

[17-jdk-slim](#) ✓ Log4Shell CVE not detected

Last pushed 5 months ago by [doijanky](#)

DIGEST

[779635c0c3d2](#)

[d732b25fa8a6](#)

OS/ARCH

linux/amd64

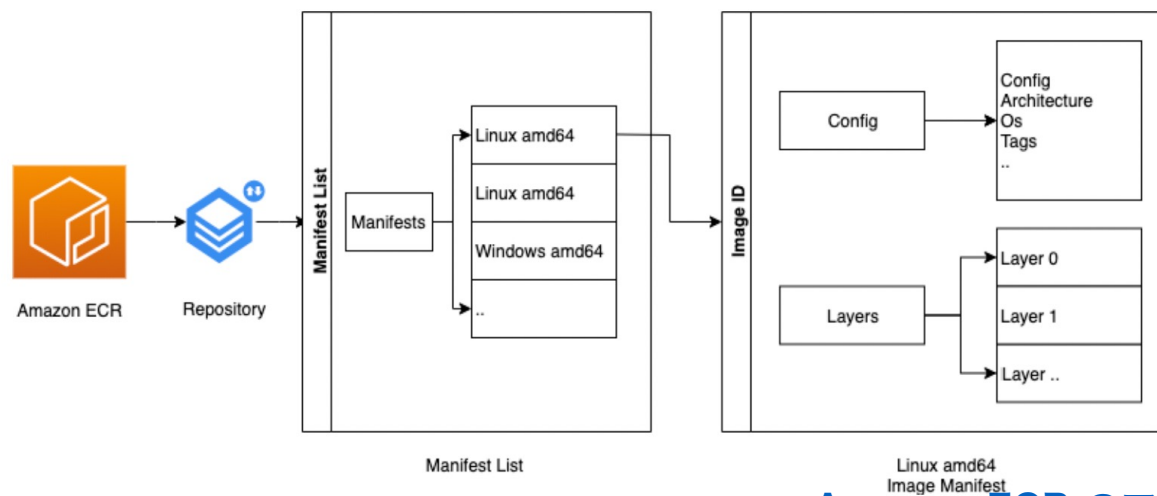
OS/ARCH

linux/amd64

linux/arm64/v8

Multi-arch Image

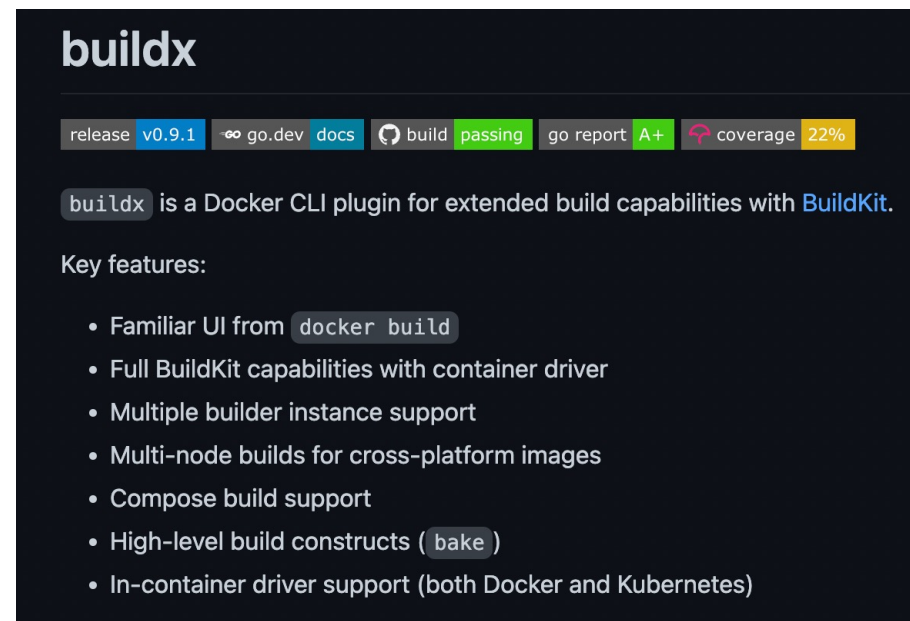
- 複数のOS/CPUアーキテクチャに対応したコンテナイメージ
 - 中身はアーキテクチャ毎の複数のイメージをバンドルしたManifest List
 - 複数のImageを単一のImageのように扱うことができる
- コンテナの可搬性をさらに高め、デプロイ先のNodeについて意識せずに運用することができる
 - 同じImageTagで通常のインスタンス⇔Gravitonインスタンス間の切り替えが可能に



》 Buildxの活用

• Multi-arch Imageの作成に Buildxを利用

- BuildxはBuildkitの機能を利用するためのDockerのCLIプラグイン
- コマンド一つでMulti-arch Imageのビルドを実行可能



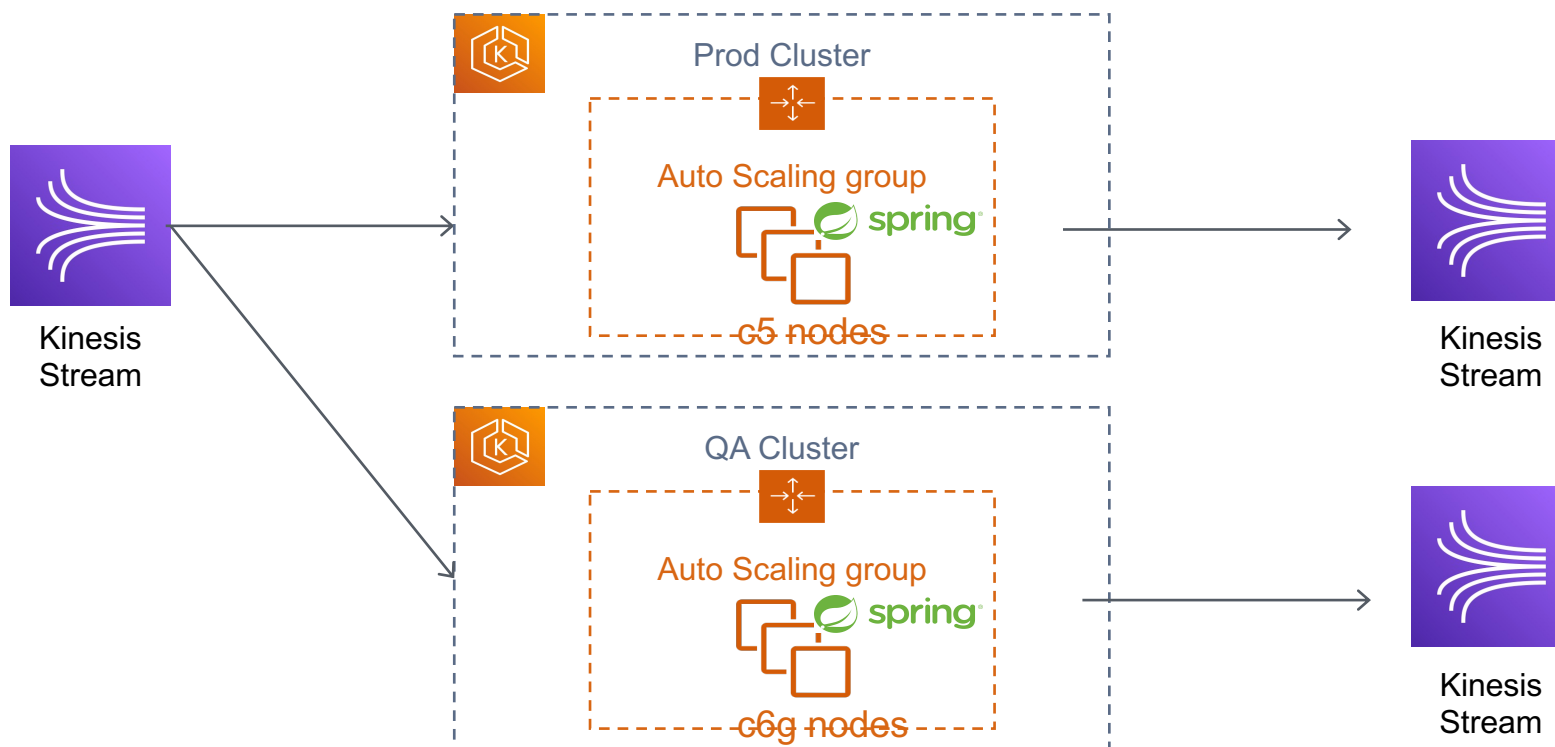
The screenshot shows the GitHub repository for Docker Buildx. At the top, it displays the repository name 'buildx' and a navigation bar with links for 'release v0.9.1', 'go.dev docs', 'build passing', 'go report A+', and 'coverage 22%'. Below this, a description states: 'buildx is a Docker CLI plugin for extended build capabilities with BuildKit.' Underneath, the 'Key features:' section lists several capabilities: 'Familiar UI from docker build', 'Full BuildKit capabilities with container driver', 'Multiple builder instance support', 'Multi-node builds for cross-platform images', 'Compose build support', 'High-level build constructs (bake)', and 'In-container driver support (both Docker and Kubernetes)'.

<https://github.com/docker/buildx>

```
docker buildx build --platform linux/amd64,linux/arm64 . -t ${IMAGE_TAG} --push
```

》パフォーマンステスト

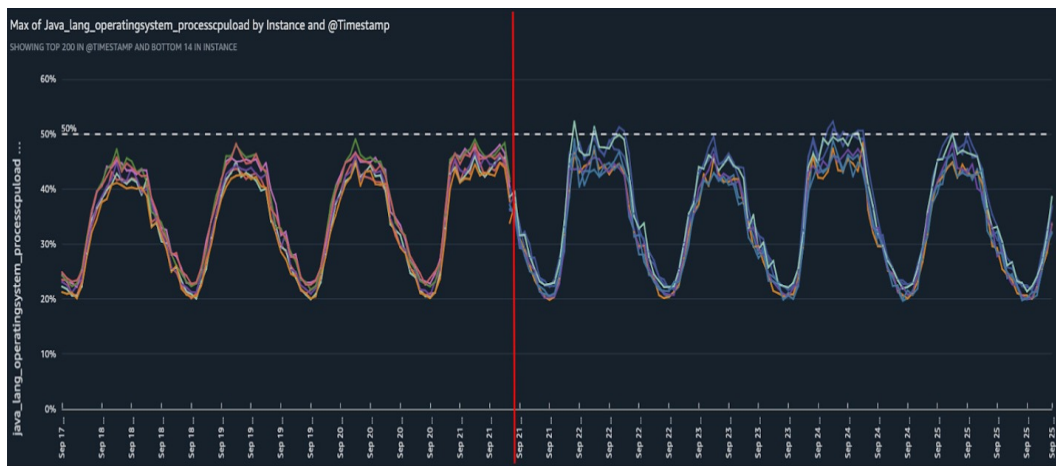
Graviton2インスタンスでの動作確認・リソースサイジングを行うために、本番と同一のデータで試験実施可能な環境でパフォーマンステストを実施



パフォーマンステスト

CPU使用率やヒープメモリ使用量を比較した結果、c5と同量のリソースサイズで安定動作することを確認

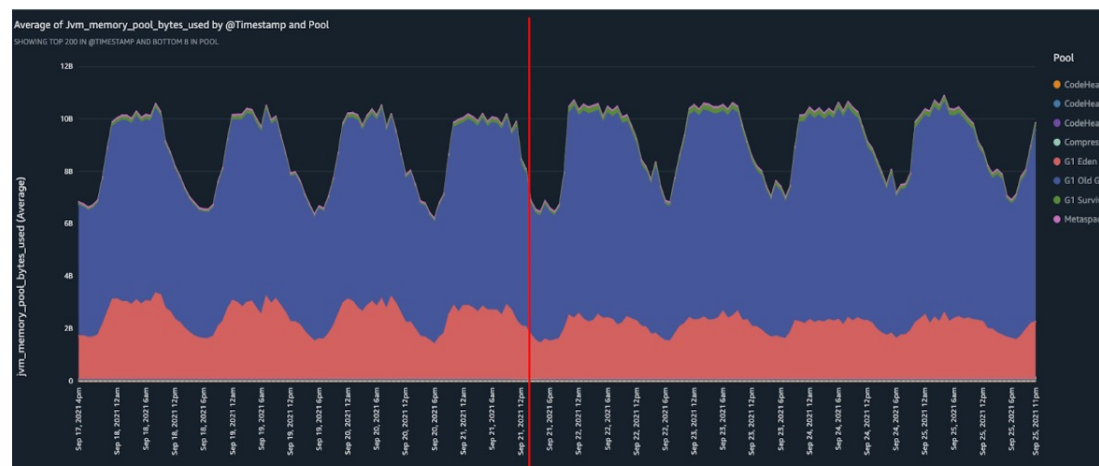
CPU使用率



ARM (c6g)

Intel (c5)

ヒープメモリ使用量



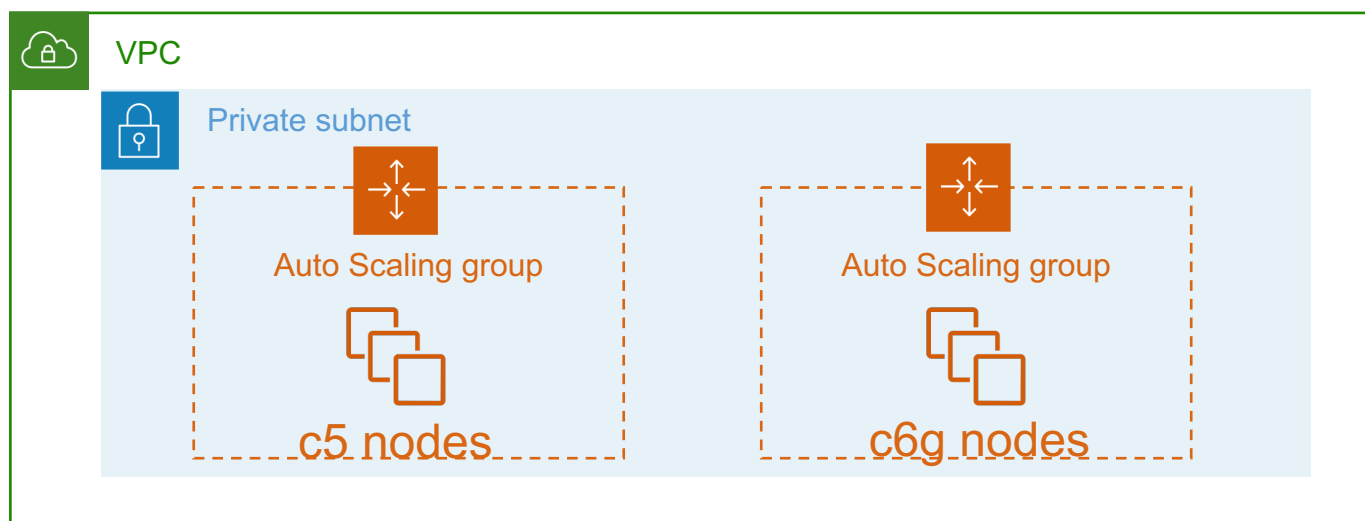
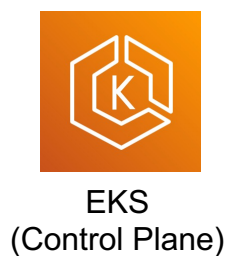
ARM (c6a)

Intel (c5)

c6gへの移行により、EC2コストを約20%削減できるため移行を決定

》 Node Groupの拡張

- 同一クラスター上にGraviton2インスタンス用のNode Groupを追加
 - arm用のEKS最適化AMI を使用してASGを作成



》》 デプロイ

下記の流れで実施

1. 移行先NodeGroupのASG台数を拡張
2. エリア毎に順次デプロイ
 - I. 移行元のアプリケーションを停止
 - II. 移行先Node Group上に再デプロイ
3. 全てのアプリケーションの移行が完了したら移行元NodeGroupのASGを縮小

》 nodeSelectorを用いたScheduling

c6gインスタンスへのデプロイは nodeSelectorを活用

- ワーカーノードにEKS最適化AMI
を利用している場合、
instanceType Labelが付与
- nodeSelectorにinstanceTypeを
指定することで意図したインスタ
ンスへをデプロイすることが可能

```
apiVersion: v1
kind: Pod
metadata:
  name: app
spec:
  containers:
  - name: app
    image: app_image
  nodeSelector:
    instanceType: ${INSTANCE_TYPE}
    . . .
```

》 Node Group閉塞時に起こったトラブル

- 全てのアプリケーションでc6gへの移行が完了したため、c5のNode Groupを閉塞したところ数分間の障害が発生
- 調査の結果、c5に残っていたcorednsが削除されサービスディスクバリが機能しなくなったことが原因と判明



閉塞前にドレーン処理を必ず実施する

参考：

[新しいノードグループへの移行](#)

》 移行の結果

- アプリケーションにほとんど手を加えることなく約20%のEC2コスト削減に成功
 - サーバー台数が多いため、費用対効果が高い取り組みに
- Multi-arch Imageに対応したCI/CDパイプラインを整備することができた
 - 性能の良いインスタンスへすぐに移行可能な状態に

》まとめ

- Gravitonインスタンスへの移行は費用対効果の高いコスト削減の取り組みになり得るため検討の価値あり
- Multi-arch Imageの活用は、アプリケーションの可搬性をさらに高め、サーバー移行を容易にする
- 移行後のNode Groupの閉塞には注意が必要