



# AWS のカスタムシリコン AWS Graviton プロセッサ

滝口 開資 (Haruyoshi Takiguchi)

アマゾン ウェブ サービス ジャパン合同会社

シニアソリューションアーキテクト フレキシブルコンピューティングスペシャリスト

2023/07/20

# 自己紹介

## • 滝口 開資 (はるよし)

シニアソリューションアーキテクト フレキシブルコンピューティングスペシャリスト

## • 普段の業務

Amazon EC2の起動方式とコストの最適化に取り組むお客様を技術面からサポート

## • 好きな AWS サービス

- Amazon EC2 Auto Scaling
- AWS Support



# 今日のアジェンダ

AWS Graviton の特徴とイノベーション

AWS Graviton の技術的特徴

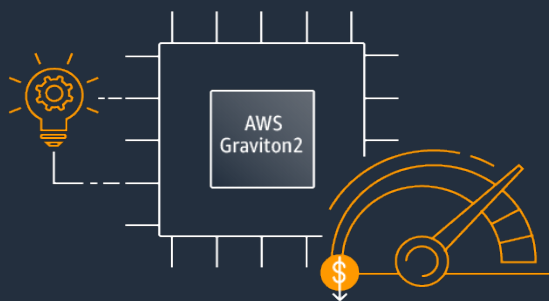
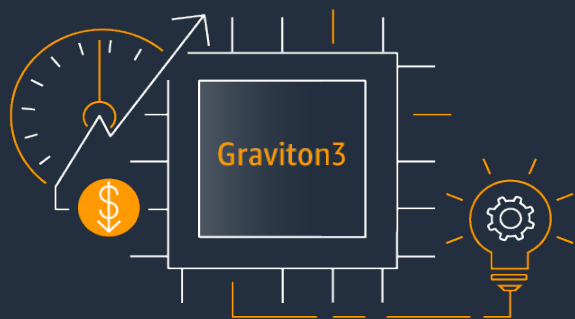
AWS Graviton の活用事例とワークロード

AWS Graviton 導入成功の秘訣

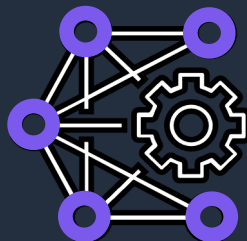
# AWS Graviton の特徴と イノベーション



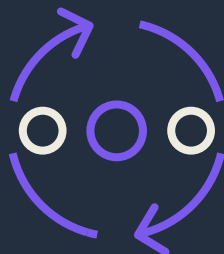
# AWS Graviton プロセッサ



64 ビット Arm プロセッサコア搭載  
カスタム AWS シリコン



クラウドネイティブなワークロードに最適化



お客様に代わって迅速なイノベーション、ビルド、  
イテレートを実施

# AWS Graviton を選択する価値



幅広いワークロードで価格対性能比が最大 40% 向上



同等の x86 ベースのインスタンスよりも最大 20% 安価\*



同等の x86 ベースのインスタンスと比較して  
エネルギー効率が最大 60% 向上

\* オンデマンドインスタンス価格での比較



# AWS Graviton の加速するイノベーション

## Graviton

2018 リリース

AWS で搭載された第 1 世代  
スケールアウトできる  
ワークロードに大幅なコスト  
最適化をもたらした

## Graviton2

2019 リリース

Core:Neoverse-N1

同等のx86ベースのインスタンス  
と比べて価格パフォーマンスが  
最大 40% 向上し、  
幅広いワークロードに対応

## Graviton3

2022 リリース

Core:Neoverse-V1

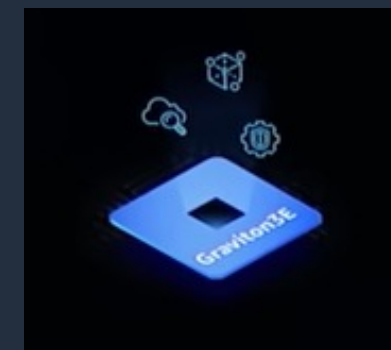
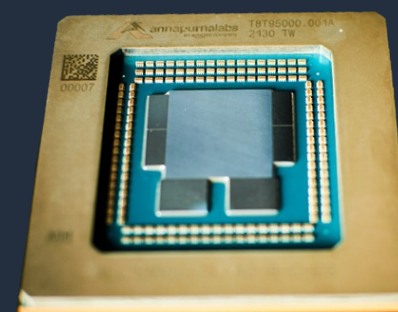
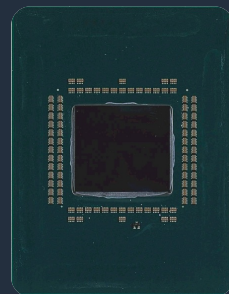
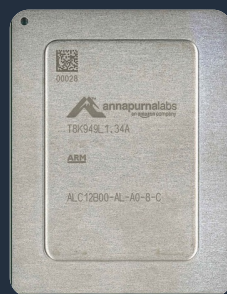
Graviton2 より25% 性能向上  
x86 インスタンスより  
60% 消費電力削減

## Graviton3E

2023 リリース

Core:Neoverse-V1

Graviton3 を  
HPC 向けに最適化  
最大35% 計算性能向上



<https://aws.amazon.com/jp/ec2/graviton/>



# なぜ自分たちでチップを作るのか？



## 最適化

AWS の仕様に合わせて  
ハードウェアを最適化  
高い電力効率



## スピード

製品の仕様化から導入  
までエンドツーエンド  
の開発プロセス



## 運用

信頼性・可用性  
動作監視・自己回復機能  
をチップレベルで実装



## イノベーション

より多くの価値を創造  
エンドツーエンドでの  
最適化



# AWS Graviton の 抑えておきたい技術的特徴



# Amazon EC2 で選択できるプロセッサと 命令セットアーキテクチャ



Intel® Xeon  
Scalable  
processors

m6i.xlarge



AMD EPYC  
processors

m6a.xlarge



AWS Graviton  
processors

m7g.xlarge



Apple M1  
processors

mac2.metal

x86

Arm64



# AWS Graviton 搭載 EC2 インスタンス

2023-07-20現在

ファミリー	Graviton2	Graviton3	Graviton3E
一般用途向け General Purpose	M6g, M6gd, T4g	M7g	
コンピューティング最適化 Compute Optimized	C6g, C6gd, C6gn	C7g	C7gn (New)
メモリ最適化 Memory Optimized	R6g, R6gd, X2gd	R7g	
高速コンピューティング Accelerated Computing	G5g (GPU 搭載)		
ストレージ最適化 Storage Optimized	Im4gn, Is4gen, I4g		
HPC最適化 HPC Optimized			HPC7g (New)

橙字：東京・大阪対応

白字：東京対応・大阪未対応

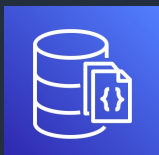
青字：東京・大阪未対応



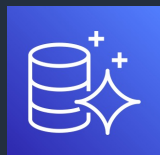
# マネージドサービスでも選択できる AWS Graviton

AWS GRAVITON のコストパフォーマンスをマネージドサービスにも拡大

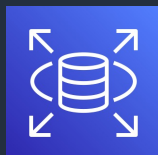
## データベース



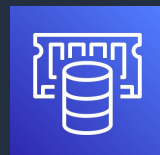
Amazon DocumentDB



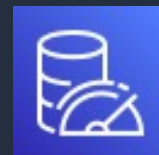
Amazon Aurora



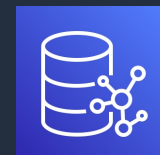
Amazon RDS



Amazon ElastiCache

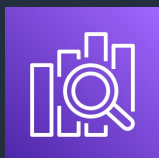


Amazon MemoryDB

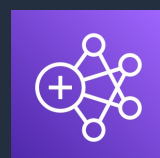


Amazon Neptune

## アナリティクス



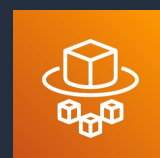
Amazon OpenSearch Service



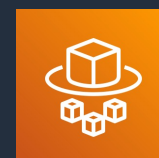
Amazon EMR



AWS Lambda



AWS Fargate



AWS Elastic Beanstalk



Amazon SageMaker

## コンピューート

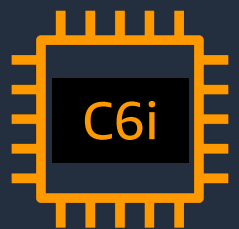
## 機械学習

aws-graviton-getting-started/managed\_services.md at main · aws/aws-graviton-getting-started · GitHub

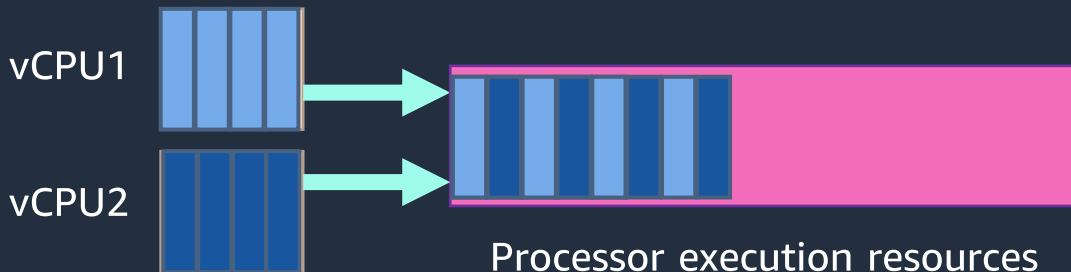
[https://github.com/aws/aws-graviton-getting-started/blob/main/managed\\_services.md](https://github.com/aws/aws-graviton-getting-started/blob/main/managed_services.md)



# Graviton のパフォーマンスのひみつ



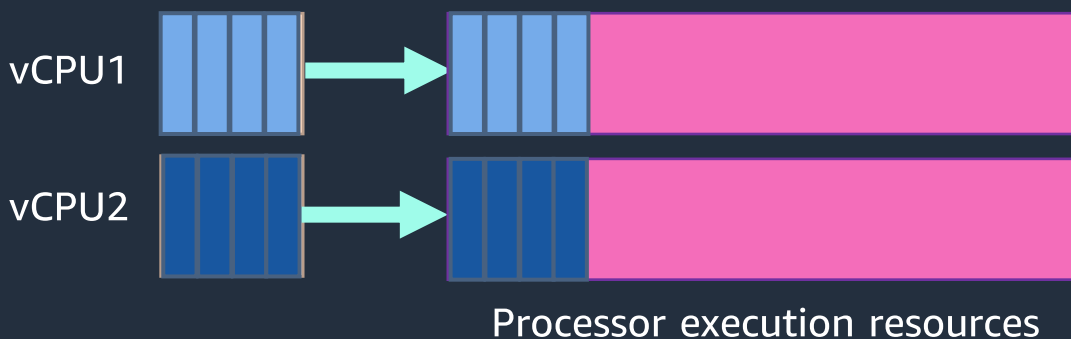
C6i instance



x86 系インスタンス : 2 vCPU = 1 物理コア  
(例) c6i.16xlarge : 64 vCPU = 32 物理コア



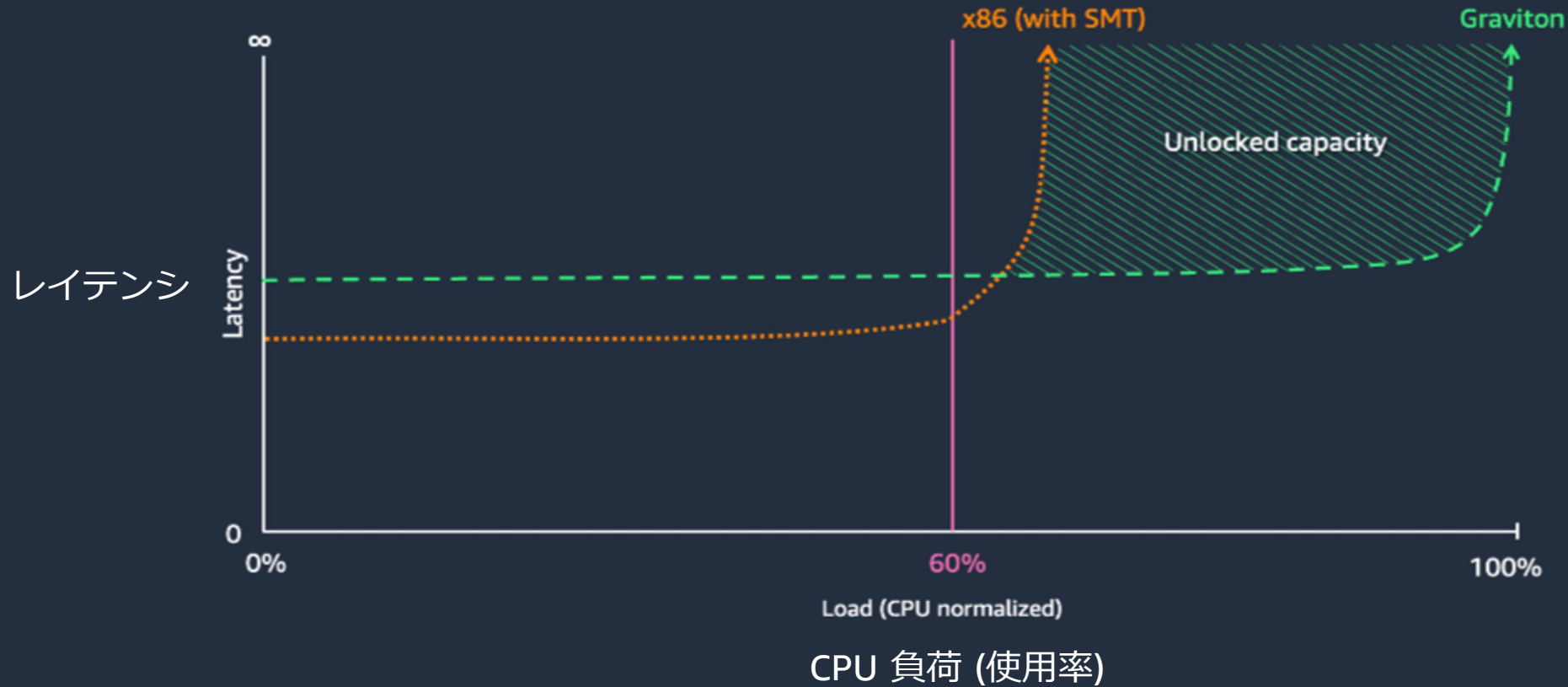
C7g instance



Graviton インスタンス : 1 vCPU = 1 物理コア  
(例) c7g.16xlarge : 64 vCPU = 32 物理コア

- 全体的なパフォーマンスが高く、より重い負荷でも良好に動作
- 複数のワークロードが稼働する環境で、より一貫したパフォーマンス
- 物理コアが独立した “medium” サイズのインスタンスを提供

# マルチスレッドなアプリケーションに最適なパフォーマンス

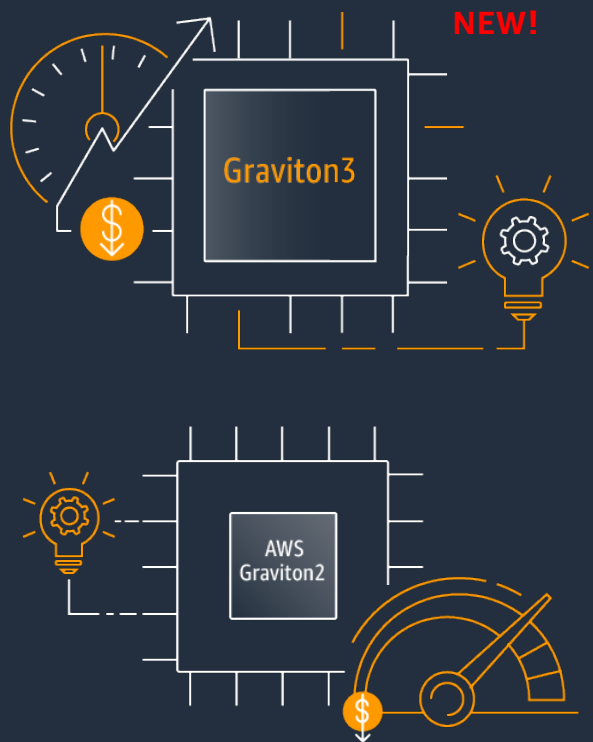


- Graviton は CPU への負荷が極めて高いときにもほぼ変わらないレイテンシを示す
- 自動スケーリングの条件に CPU 使用率を使っている場合、より厳しくできる

# AWS Graviton の 活用事例とワークロード



# Global 50 社のうち 48 社が AWS Graviton を採用



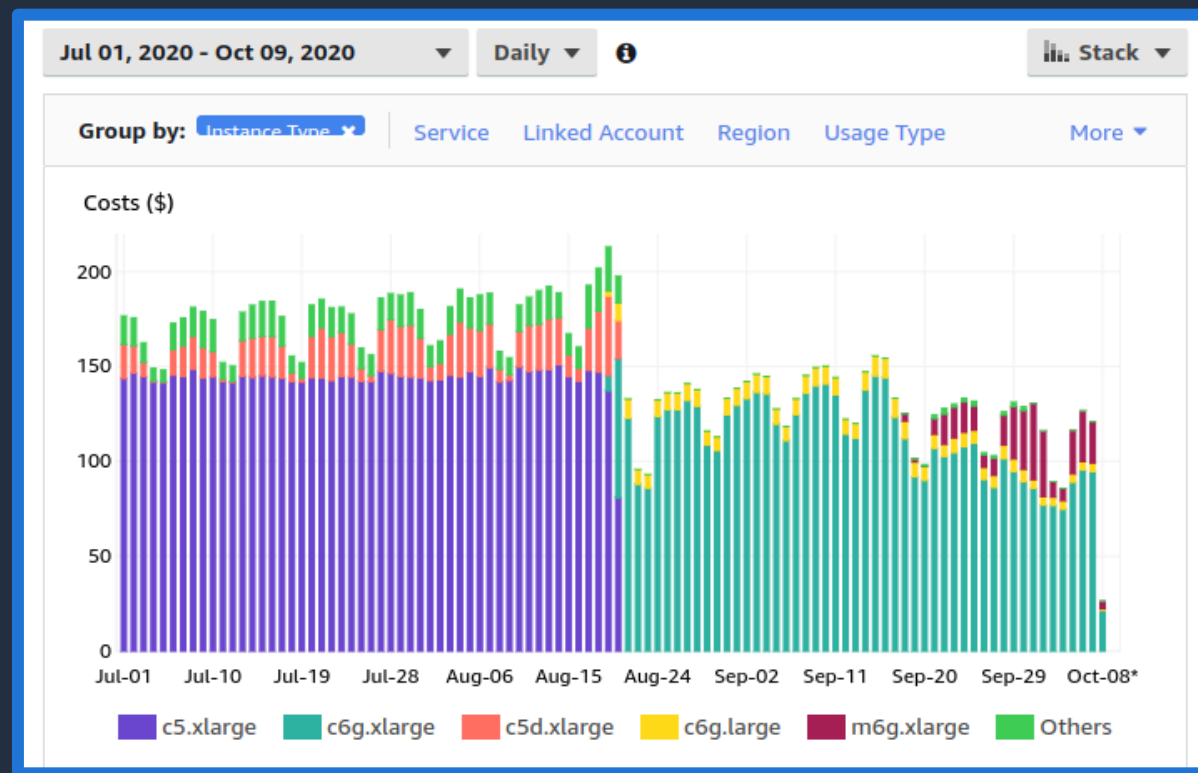
Amazon EC2 のグローバルトップ 50 社の顧客のうち 48社が、2019 年 12月にプレビューを開始した AWS の Arm ベース Graviton2 チップを採用しています



# Honeycomb.io 様の費用最適化：35% の節減



“自社向けワークロードの 100% を M6g に移行した結果、C5 を使用していたときと比較して 35% 少ないインスタンス数で済むようになった”



<https://www.honeycomb.io/blog/observations-on-arm64-awss-amazon-ec2-m6g-instances/>

<https://twitter.com/lizthegrey/status/1314618333140971521?s=20>

<https://www.honeycomb.io/blog/graviton2-one-year-retrospective/>



# ナビタイムジャパン様

- "NAVITIME is the leading provider of navigation technology and services in Japan. We completed the deployment of 4 services, the common API, map data, tile map, and full-text search distribution services, running on x86 based instances to Graviton2-based **M6g and C6g instances within just 3 months for the purpose of cost reduction**. These services are running on Java8 and C/C++ applications with Amazon EKS/ECS. Migration to M6g instance provided up to **15% higher throughput** at **20% lower cost**. We are investigating additional workloads to migrate to Graviton2 instances."
- Mr. Katsuhide Kayashima, senior engineer, NAVITIME JAPAN Co., Ltd.

- 「NAVITIMEは、日本におけるナビゲーション技術とサービスの大手プロバイダーです。コスト削減を目的として、x86ベースのインスタンスで実行される共通API、マップデータ、タイルマップ、全文検索配信サービスの4つのサービスを、わずか**3か月**でGraviton2ベースの**M6gおよびC6gインスタンス**に移行しました。これらのサービスは、Amazon EKS/ECS を搭載した Java8 および C/C++ アプリケーションで実行されています。M6g インスタンスへの移行により、**スループットが最大15% 向上**し、**コストは20% 削減**されました。現在、Graviton2 インスタンスに移行するための追加のワークロードを調査しています。」

<https://aws.amazon.com/jp/ec2/graviton/customers/>

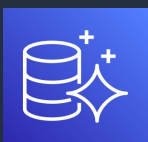


# AWS Graviton 導入成功の秘訣

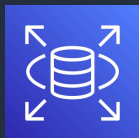


# AWS Graviton のターゲットワークロード

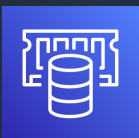
## AWS マネージドサービス



Amazon Aurora



Amazon RDS



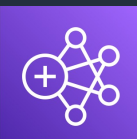
Amazon Elasticache



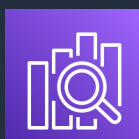
Amazon EKS



Amazon ECS



Amazon EMR



Amazon OpenSearch



© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

## EC2 でのセルフマネージ型ワークロード

Web + ゲームサーバー



分析



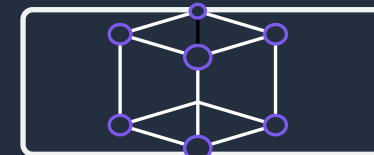
オープンソース  
データベース



マイクロサービス/コンテナ



HPC と AI/ML



## ワークロード特性

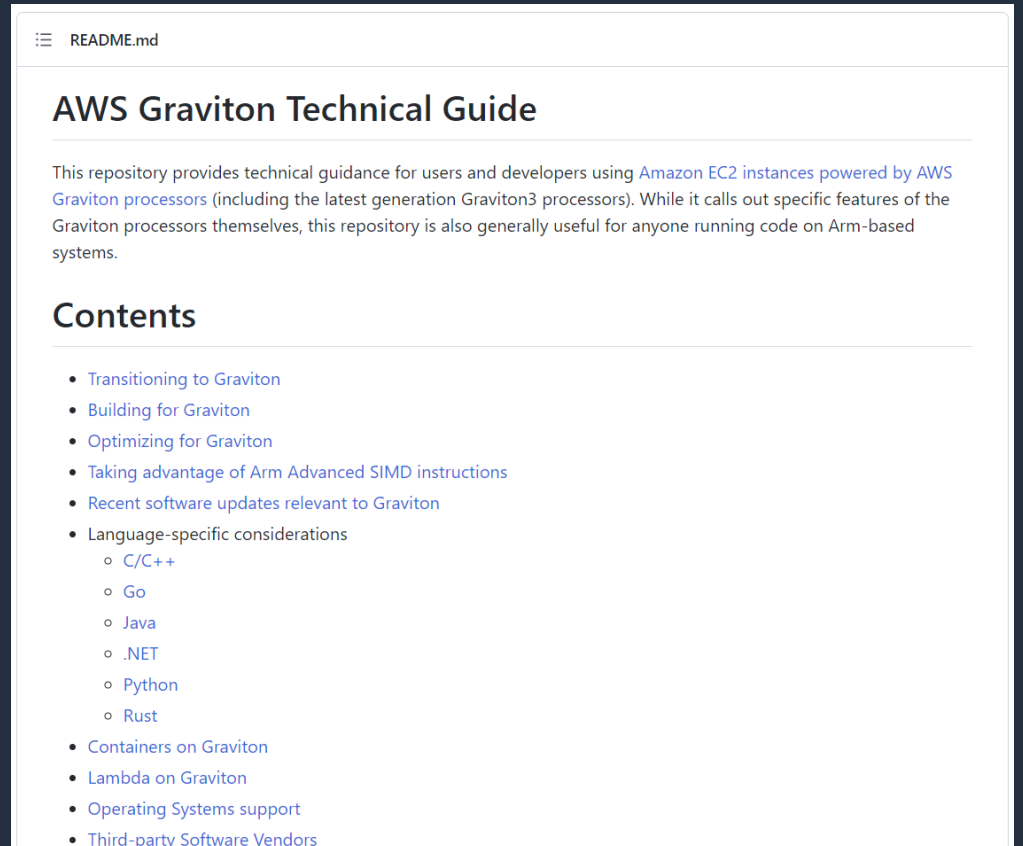
- ✓ オープンソース、Linux ベースである (2023年7月時点で Windows版AMIは存在しない)
- ✓ 最新バージョンのソフトウェア (プログラミング言語、OS、データベースエンジンなど) を実行できる
- ✓ 上位レベルのプログラミング言語 (Java、PHP、node.JS など) で書かれている→容易
- ✓ コンパイル言語 (C/C++、Python、Go など) で書かれている→やや注意が必要

# AWS Graviton Technical Guide

**AWS Graviton 利用時に最初にご確認いただきたいドキュメント**

<https://github.com/aws/aws-graviton-getting-started>

- プログラミング言語別考慮事項
  - 推奨コンパイルオプション
  - アーキテクチャ間の差異
- OS サポート情報
- 各種アプリケーションでの推奨バージョン、設定
- デバッグ・プロファイル
- etc..



☰ README.md

## AWS Graviton Technical Guide

This repository provides technical guidance for users and developers using [Amazon EC2 instances powered by AWS Graviton processors](#) (including the latest generation Graviton3 processors). While it calls out specific features of the Graviton processors themselves, this repository is also generally useful for anyone running code on Arm-based systems.

### Contents

- [Transitioning to Graviton](#)
- [Building for Graviton](#)
- [Optimizing for Graviton](#)
- [Taking advantage of Arm Advanced SIMD instructions](#)
- [Recent software updates relevant to Graviton](#)
- [Language-specific considerations](#)
  - [C/C++](#)
  - [Go](#)
  - [Java](#)
  - [.NET](#)
  - [Python](#)
  - [Rust](#)
- [Containers on Graviton](#)
- [Lambda on Graviton](#)
- [Operating Systems support](#)
- [Third-party Software Vendors](#)



# Arm アーキテクチャである AWS Graviton への 移行は大変？



プログラミング言語環境を含む  
様々な OSS アプリケーションが対応済み  
多くのお客様から想像よりずっと簡単だったとの声

# AWS Graviton 導入成功の秘訣

- 移行方針・組織のマインドセット
  - マネージドサービスから始める
  - ここで問題が出なければ Graviton の導入実績になり、EC2 環境にも適用しやすくなる
- 新規開発のワークロードは Graviton ファーストで考える
- 移行検証・動作検証を入念に実施する
  - 現行 x86 インスタンス上で動作しているソフトウェアを洗い出す
  - ソースコードレベルでの自動チェックに Porting Advisor for Graviton
    - aws/porting-advisor-for-graviton — <https://github.com/aws/porting-advisor-for-graviton>



# まとめ

AWS Graviton の特徴とイノベーション

AWS Graviton の技術的特徴

AWS Graviton の活用事例とワークロード

AWS Graviton 導入成功の秘訣





# Thank you!