アプリケーション観点での AWS Graviton へのワークロード移行

2023/05/25

Kyosuke Ochimizu

Amazon Web Services Japan G.K.

自己紹介

落水 恭介 (Ochimizu Kyosuke)

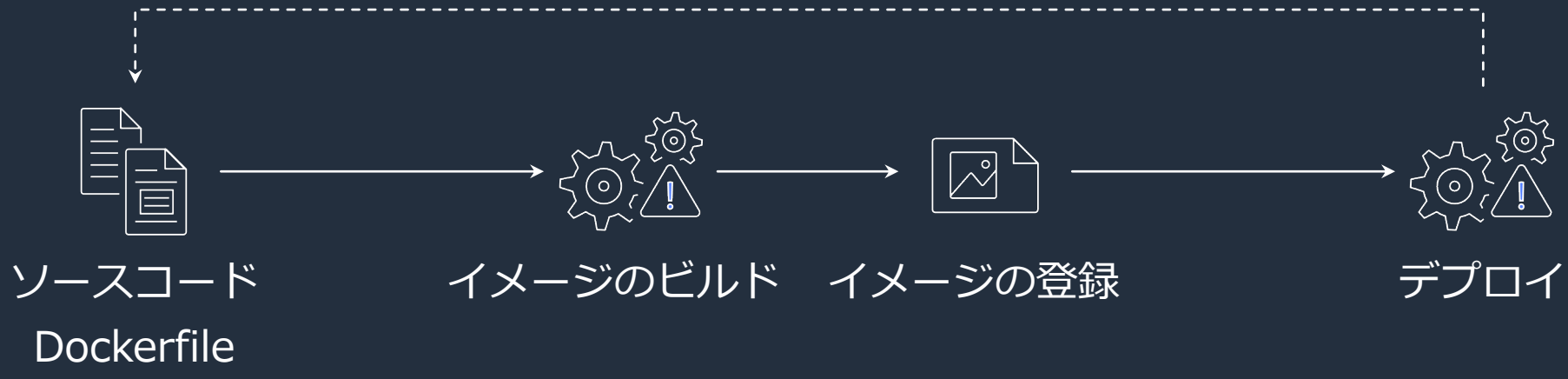
コンテナスペシャリストソリューションアーキテクト

- SIer
- 教育業界ベンチャー
- Cloud Integrator
- サポートチーム / アマゾン ウェブ サービス ジャパン
- 現在のロール

好きな AWS サービス: AWS Fargate

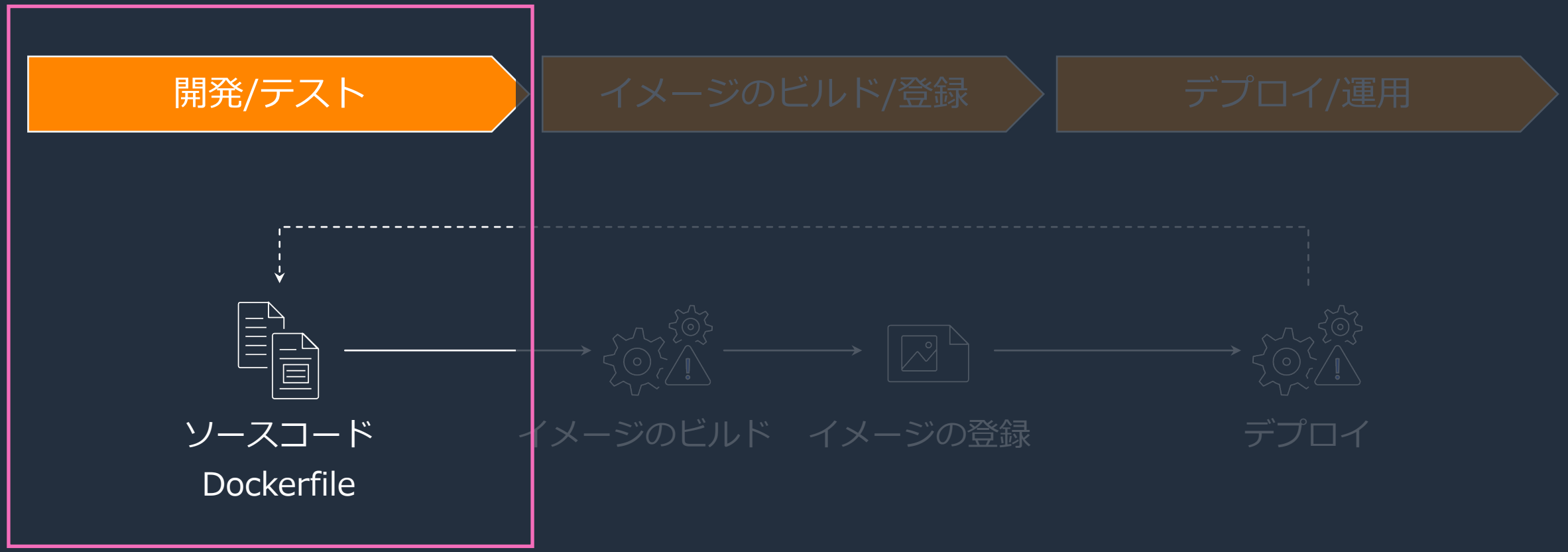
コンテナにおけるソフトウェア開発ライフサイクル

開発/テスト → イメージのビルド/登録 → デプロイ/運用



本セッションでお話すること

AWS Graviton へのワークロード移行における、
アプリケーションソースコードへの対応についてご紹介します



Graviton への移行ステップ

Graviton への移行に必要な作業とは？

移行ステップ

1. アプリケーションに関連するソフトウェアの状況を整理
2. Graviton でアプリケーション実行環境を構築
3. 機能テスト / パフォーマンステストの実施
4. Graviton を利用するようにインフラストラクチャを更新

[Step 1]
アプリケーションに関連する
ソフトウェアの状況を整理

現在のソフトウェア利用状況を一覧化する

ダウンロードしたソフトウェア

オープンソースプロジェクト、コンテナイメージ、ライブラリ

自分達で開発したソフトウェア

アプリケーションコード

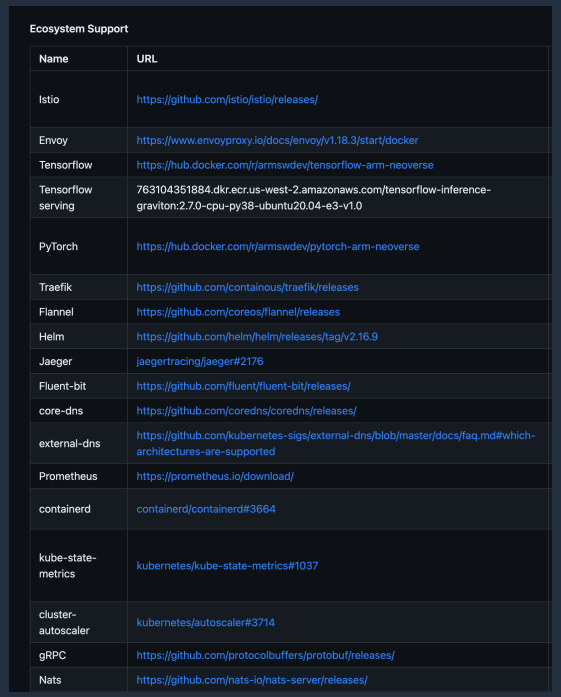
購入したソフトウェア

パッケージ、SaaS エージェント

ダウンロードしたソフトウェアの状況整理

- 利用しているソフトウェアが ARM64 対応しているか確認
  - ARM64 対応のリリース
  - ARM64 向けにビルドされたコンテナイメージ

Container-based workloads on Graviton



Ecosystem Support

Name	URL
Istio	https://github.com/istio/istio/releases/
Envoy	https://www.envoyproxy.io/docs/envoy/v1.18.3/start/docker
Tensorflow	https://hub.docker.com/r/armswdev/tensorflow-arm-neoverse
Tensorflow serving	763104351884.dkr.ecr.us-west-2.amazonaws.com/tensorflow-inference-graviton:2.7.0-cpu-py38-ubuntu20.04-e3-v1.0
PyTorch	https://hub.docker.com/r/armswdev/pytorch-arm-neoverse
Traefik	https://github.com/containous/traefik/releases
Flannel	https://github.com/coreos/flannel/releases
Helm	https://github.com/helm/helm/releases/tag/v2.16.9
Jaeger	jaegertracing/jaeger#2176
Fluent-bit	https://github.com/fluent/fluent-bit/releases/
core-dns	https://github.com/coredns/coredns/releases/
external-dns	https://github.com/kubernetes-sigs/external-dns/blob/master/docs/faq.md#which-architectures-are-supported
Prometheus	https://prometheus.io/download/
containerd	containerd/containerd#3664
kube-state-metrics	kubernetes/kube-state-metrics#1037
cluster-autoscaler	kubernetes/autoscaler#3714
gRPC	https://github.com/protocolbuffers/protobuf/releases/
Nats	https://github.com/nats-io/nats-server/releases/

- 必要に応じてソフトウェアのバージョンアップを実施
  - 例) version 2.1.0 以降で ARM64 に対応

- 公式には ARM64 が未サポート場合は回避策を検討
  - 公式が未サポート ≠ ARM64 環境で動かない
  - Issue / Pull Request の作成

https://github.com/aws/aws-graviton-getting-started/blob/main/containers.md

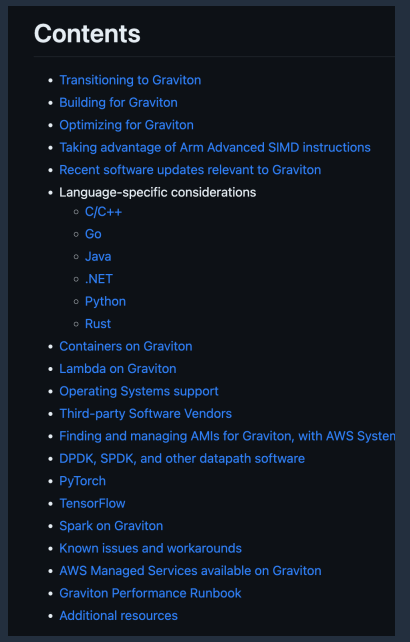
自分達で開発したソフトウェアの状況整理

- OS が Graviton (ARM64) をサポートしているか確認
  - （コンテナの場合）ベースイメージが提供されているか

AWS Graviton Technical Guide

- アプリケーションの関連コンポーネントのサポート状況を確認
  - 例) ランタイム、ライブラリ、アプリケーションフレームワーク

- コンパイラやテストツールのサポート状況を確認
  - C や C++、Go などは ARM64 向けに再コンパイルが必要
  - プログラミング言語ごとの推奨アクションをガイドとして提供（右記）



https://github.com/aws/aws-graviton-getting-started/blob/main/README.md

購入したソフトウェアの状況整理

- パッケージや SaaS エージェントの ARM64 サポート状況を確認
  - ベンダーからの ARM64 対応バージョンの提供有無
  - ARM64 非対応の場合、オープンソースプロジェクトと異なり、自身での対応は難しい（ライセンス、サポート、etc）

- Graviton (ARM64) をサポートする ISV の製品リストを確認

Category	ISV	Product	Resources
Analytics	Databricks	Databricks Lakehouse	Announcement: https://databricks.com/blog/2022/04/18/announcing-databricks-su graviton2-with-up-to-3x-better-price-performance.html Documentation: https://docs.databricks.com/clusters/graviton.html
Analytics	InterSystems	IRIS (Community Edition)	Release: https://aws.amazon.com/marketplace/pp/B08629M8S4 Announcement: https://community.intersystems.com/post/intersystems-iris-now-a graviton2-based-amazon-ec2-instances
Analytics	Splunk	Splunk Cloud Platform	Announcement: https://www.splunk.com/en_us/blog/platform/splunk-embarks-on- journey-with-amazon-ec2-im4gn-and-is4gen-instances.html
Analytics	Starburst	Starburst Enterprise	Release: https://aws.amazon.com/marketplace/pp/prodview-2bpppdqlesn6w Announcement: https://docs.starburst.io/361-e/release/release-360-e.html

https://github.com/aws/aws-graviton-getting-started/blob/main/isv.md

コンテナにおけるソフトウェア利用状況の確認

1. Dockerfile の確認

例) AWS Copilot CLI の Dockerfile

```
FROM golang:1.20
# We need to have both nodejs and go to build the binaries.
# We could use multi-stage builds but that would require significantly changing our Makefile.
RUN apt-get update
RUN curl -sL https://deb.nodesource.com/setup_10.x | bash -
RUN apt-get update && apt-get install -y nodejs

WORKDIR /copilot
COPY . .
RUN go env -w GOPROXY=direct
RUN make release
```

https://github.com/aws/copilot-cli/blob/mainline/Dockerfile

コンテナにおけるソフトウェア利用状況の確認

1. Dockerfile の確認 : ベースイメージ

例) AWS Copilot CLI の Dockerfile

```
FROM golang:1.20
# We need to have both nodejs and go to b
# We could use multi-stage builds but tha
RUN apt-get update
RUN curl -sL https://deb.nodesource.com/s
RUN apt-get update && apt-get install -y

WORKDIR /copilot
COPY . .
RUN go env -w GOPROXY=direct
RUN make release
```

- ベースイメージの ARM64 対応を確認
- マルチアーキテクチャイメージとして提供されている場合も

```
$ docker manifest inspect golang:1.20
...
"platform": { "architecture": "amd64", "os": "linux” }
...
"platform": { "architecture": "arm64", "os": "linux", "variant": "v8” }
```

https://github.com/aws/copilot-cli/blob/mainline/Dockerfile

コンテナにおけるソフトウェア利用状況の確認

1. Dockerfile の確認：ビルド時に取得しているツールやパッケージ

例) AWS Copilot CLI の Dockerfile

```
FROM golang:1.20
# We need to have both nodejs and go to build the binaries.
# We could use multi-stage builds but that would require significantly changing our Makefile.
RUN apt-get update
RUN curl -sL https://deb.nodesource.com/setup_10.x | bash -
RUN apt-get update && apt-get install -y nodejs

WORKDIR /copilot
COPY . .
RUN go env -w GOPROXY=direct
RUN make release
```

Dockerfile 命令で取得している
ツールやパッケージの ARM64 対応を確認

https://github.com/aws/copilot-cli/blob/mainline/Dockerfile

コンテナにおけるソフトウェア利用状況の確認

1. Dockerfile の確認 : 依存ライブラリ

例) AWS Copilot CLI の Dockerfile

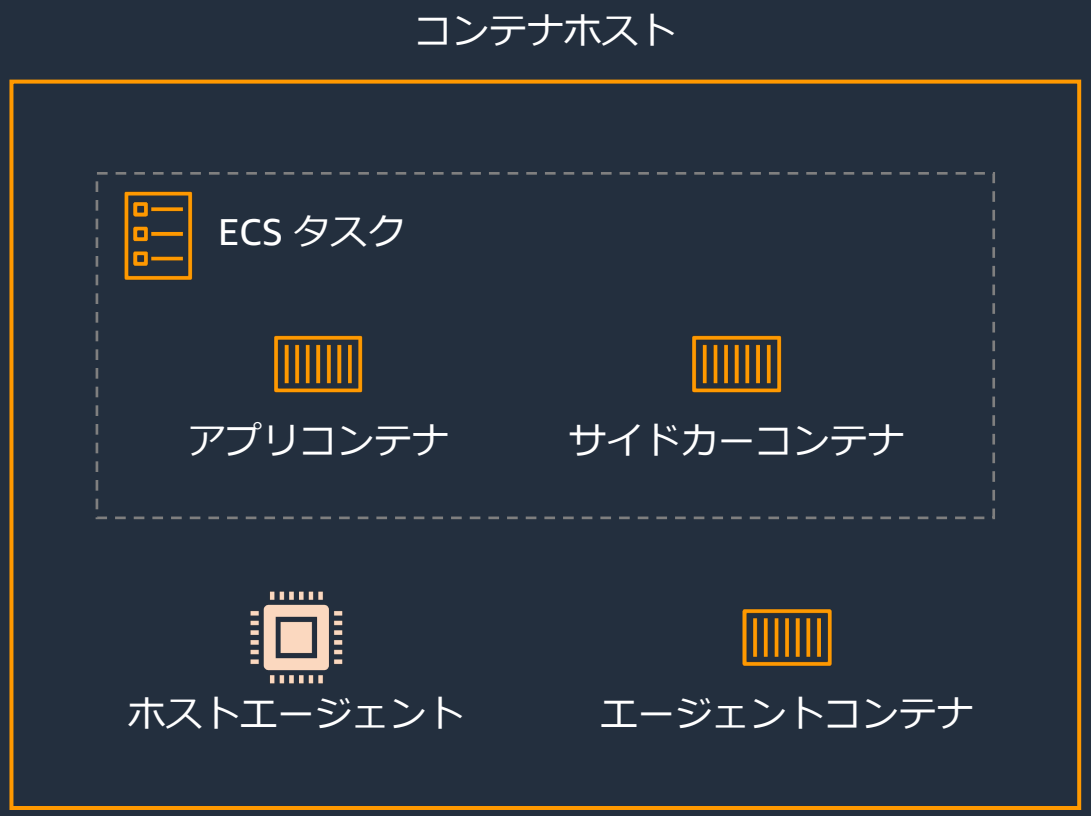
```
FROM golang:1.20
# We need to have both nodejs and go to build the binaries.
# We could use multi-stage builds but that would require significantly changing our Makefile.
RUN apt-get update
RUN curl -sL https://deb.nodesource.com/setup_10.x | bash -
RUN apt-get update && apt-get install -y nodejs

WORKDIR /copilot
COPY . .
RUN go env -w GOPROXY=direct
RUN make release
```

パッケージ管理システムで取得している
依存ライブラリの ARM64 対応を確認

https://github.com/aws/copilot-cli/blob/mainline/Dockerfile

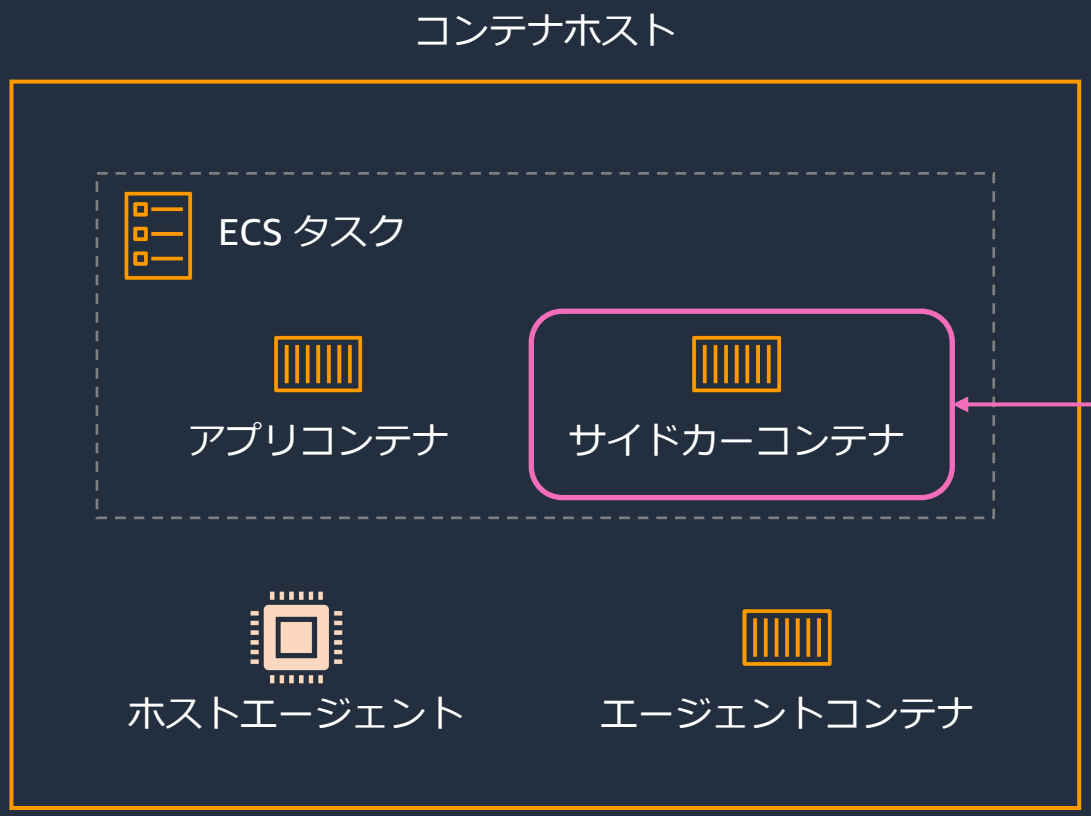
コンテナにおけるソフトウェア利用状況の確認

2. サイドカーコンテナ / ホストエージェント / コンテナホスト の確認



コンテナにおけるソフトウェア利用状況の確認

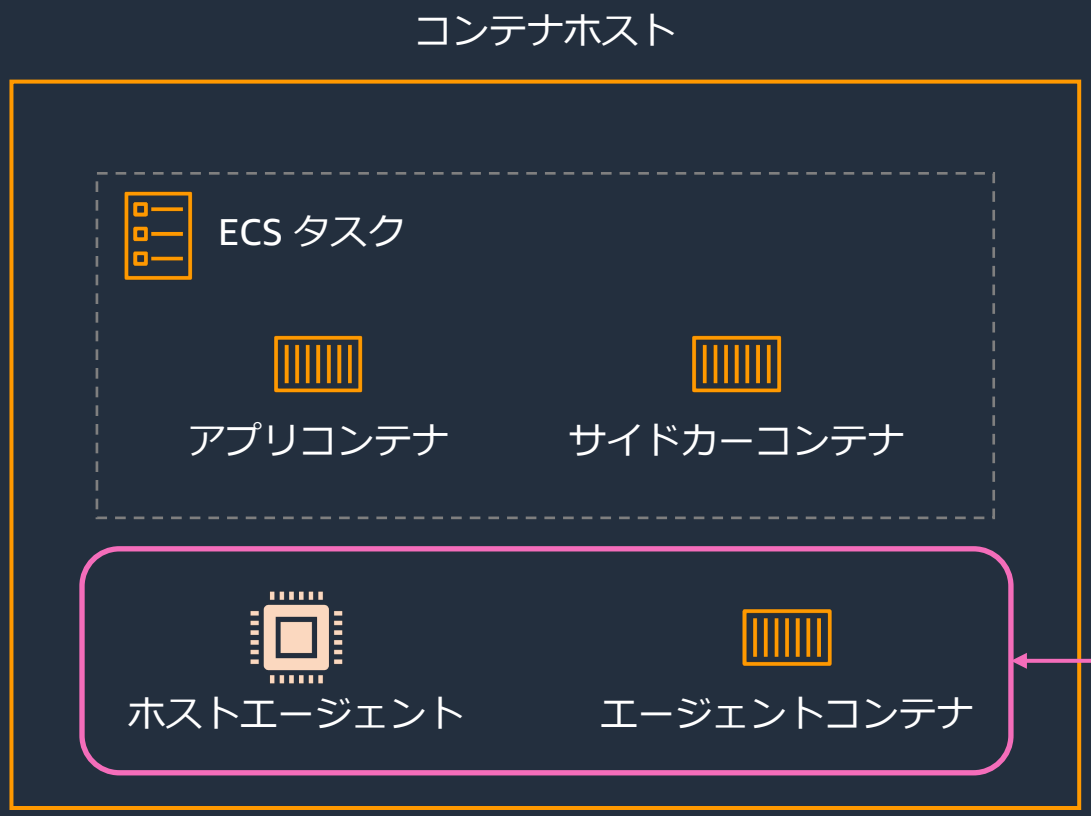
2. サイドカーコンテナ / ホストエージェント / コンテナホスト の確認



- 自分達で作成したコンテナの場合、
  アプリコンテナと同様に ARM64 対応を進める
- パブリックなコンテナイメージの場合、
  ARM64 対応のイメージの有無を確認

コンテナにおけるソフトウェア利用状況の確認

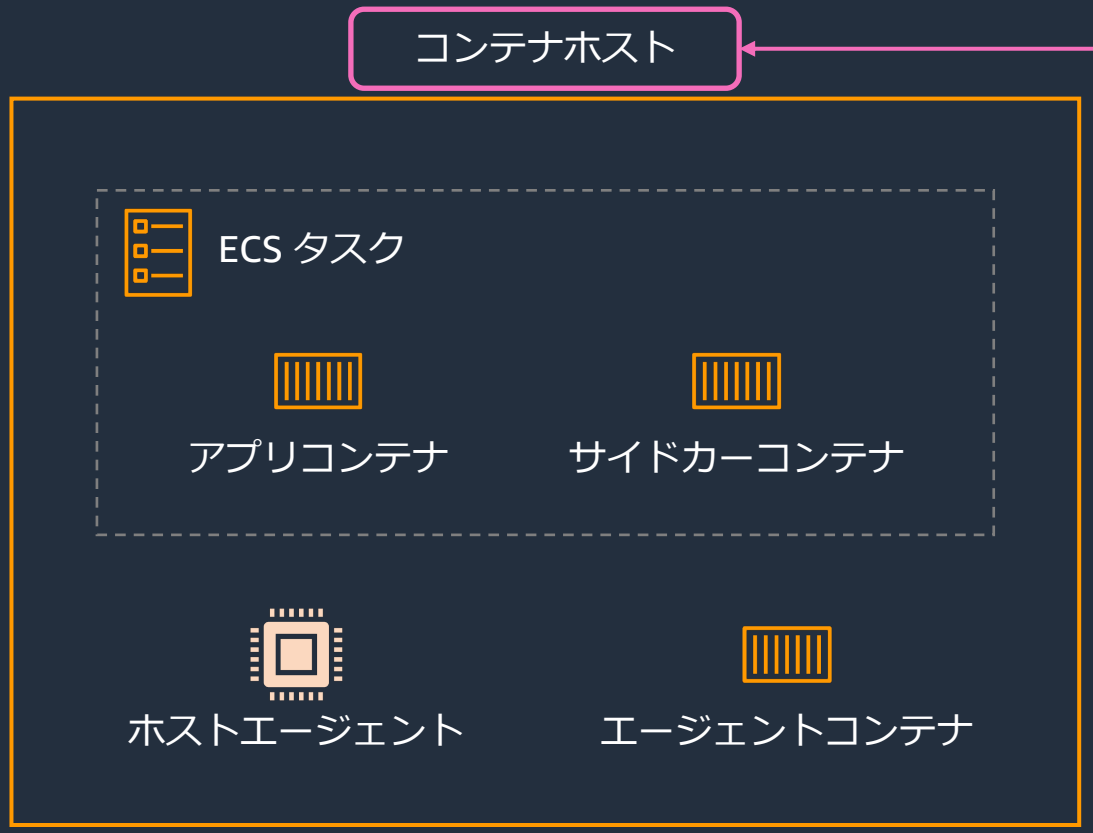
2. サイドカーコンテナ / ホストエージェント / コンテナホスト の確認



- 自分達で作成したコンテナの場合、アプリコンテナと同様に ARM64 対応を進める
- パブリックなコンテナイメージの場合、ARM64 対応のイメージの有無を確認
- コンテナホストへ直接インストールしているエージェントも要確認

コンテナにおけるソフトウェア利用状況の確認

2. サイドカーコンテナ / ホストエージェント / コンテナホスト の確認



- コンテナホストとして利用する AMI の ARM64 対応を確認
- ECS や EKS の場合、ARM64 対応の最適化 AMI を提供している

Porting Advisor for Graviton

言語ランタイムや依存ライブラリの必要・推奨バージョンを提案

```
./dist/porting-advisor-linux-x86_64 ./sample-projects/
| Elapsed Time: 0:00:03

Porting Advisor for Graviton v1.0.0
Report date: 2023-01-06 23:48:20

13 files scanned.
detected java code. we recommend using Corretto. see https://aws.amazon.com/corretto/ for more details.
detected python code. if you need pip, version 19.3 or above is recommended. we detected that you have version 22.2.1.
detected python code. min version 3.7.5 is required. we detected that you have version 3.10.6. see https://github.com/aws/aws-graviton-
getting-started/blob/main/python.md for more details.
./sample-projects/java-samples/pom.xml: dependency library: leveldbjni-all is not supported on Graviton
./sample-projects/java-samples/pom.xml: using dependency library snappy-java version 1.1.3. upgrade to at least version 1.1.4
./sample-projects/java-samples/pom.xml: using dependency library zstd-jni version 1.1.0. upgrade to at least version 1.2.0
./sample-projects/python-samples/incompatible/requirements.txt:3: using dependency library OpenBLAS version 0.3.16. upgrade to at least
version 0.3.17
detected go code. min version 1.16 is required. version 1.18 or above is recommended. we detected that you have version 1.15. see
https://github.com/aws/aws-graviton-getting-started/blob/main/golang.md for more details.
./sample-projects/java-samples/pom.xml: using dependency library hadoop-lzo. this library requires a manual build more info at:
https://github.com/aws/aws-graviton-getting-started/blob/main/java.md#building-multi-arch-jars
./sample-projects/python-samples/incompatible/requirements.txt:5: dependency library NumPy is present. min version 1.19.0 is required.
detected java code. min version 8 is required. version 11 or above is recommended. see https://github.com/aws/aws-graviton-getting-
started/blob/main/java.md for more details.

Use --output FILENAME.html to generate an HTML report.
```

https://github.com/aws/porting-advisor-for-graviton

Porting Advisor for Graviton

言語ランタイムや依存ライブラリの必要・推奨バージョンを提案

言語ランタイム・パッケージ管理ツールの最低必要バージョンと推奨バージョンを提案

```
./dist/porting-advisor-linux-x86_64 ./sample-projects/

13 files scanned.
detected java code. we recommend using Corretto. see https://aws.amazon.com/corretto/ for more details.
detected python code. if you need pip, version 19.3 or above is recommended. we detected that you have version 22.2.1.
detected python code. min version 3.7.5 is required. we detected that you have version 3.10.6. see https://github.com/aws/aws-graviton-getting-started/blob/main/python.md for more details.
./sample-projects/java-samples/pom.xml: dependency library: leveldbjni-all is not supported on Graviton
./sample-projects/java-samples/pom.xml: using dependency library snappy-java version 1.1.3. upgrade to at least version 1.1.4
./sample-projects/java-samples/pom.xml: using dependency library zstd-jni version 1.1.0. upgrade to at least version 1.2.0
./sample-projects/python-samples/incompatible/requirements.txt:3: using dependency library OpenBLAS version 0.3.16. upgrade to at least version 0.3.17
detected go code. min version 1.16 is required. version 1.18 or above is recommended. we detected that you have version 1.15. see https://github.com/aws/aws-graviton-getting-started/blob/main/golang.md for more details.
./sample-projects/java-samples/pom.xml: using dependency library hadoop-lzo. this library requires a manual build more info at: https://github.com/aws/aws-graviton-getting-started/blob/main/java.md#building-multi-arch-jars
./sample-projects/python-samples/incompatible/requirements.txt:5: dependency library NumPy is present. min version 1.19.0 is required.
detected java code. min version 8 is required. version 11 or above is recommended. see https://github.com/aws/aws-graviton-getting-started/blob/main/java.md for more details.

Use --output FILENAME.html to generate an HTML report.
```

https://github.com/aws/porting-advisor-for-graviton

Porting Advisor for Graviton

言語ランタイムや依存ライブラリの必要・推奨バージョンを提案

```
./dist/porting-advisor-linux-x86_64 ./sample-projects/
```

依存ライブラリの最低必要バージョンを提案（ARM64 サポートがないものはその旨を提示）

```
13 files scanned.
detected java code. we recommend using Corretto. see https://aws.amazon.com/corretto/ for more details.
detected python code. if you need pip, version 19.3 or above is recommended. we detected that you have version 22.2.1.
detected python code. min version 3.7.5 is required. we detected that you have version 3.10.6. see https://github.com/aws/aws-graviton-getting-started/blob/main/python.md for more details.
./sample-projects/java-samples/pom.xml: dependency library: leveldbjni-all is not supported on Graviton
./sample-projects/java-samples/pom.xml: using dependency library snappy-java version 1.1.3. upgrade to at least version 1.1.4
./sample-projects/java-samples/pom.xml: using dependency library zstd-jni version 1.1.0. upgrade to at least version 1.2.0
./sample-projects/python-samples/incompatible/requirements.txt:3: using dependency library OpenBLAS version 0.3.16. upgrade to at least version 0.3.17
detected go code. min version 1.16 is required. version 1.18 or above is recommended. we detected that you have version 1.15. see https://github.com/aws/aws-graviton-getting-started/blob/main/golang.md for more details.
./sample-projects/java-samples/pom.xml: using dependency library hadoop-lzo. this library requires a manual build more info at: https://github.com/aws/aws-graviton-getting-started/blob/main/java.md#building-multi-arch-jars
./sample-projects/python-samples/incompatible/requirements.txt:5: dependency library NumPy is present. min version 1.19.0 is required.
detected java code. min version 8 is required. version 11 or above is recommended. see https://github.com/aws/aws-graviton-getting-started/blob/main/java.md for more details.

Use --output FILENAME.html to generate an HTML report.
```

https://github.com/aws/porting-advisor-for-graviton

Graviton への移行とソフトウェアアップデート

- 可能であれば、できるだけ新しいソフトウェアバージョンへ更新することを推奨
  - 新しいほど高いパフォーマンスが期待される
  - 例) Java 11, GCC 11, Go 1.18

- ソフトウェアバージョンの更新は ARM64 へ移行する前に実施する
  - ARM64 へ移行する際の変更点を最小にするため、更新は x86 環境で実施しておく
  - x86 (現在のバージョン) -> x86 (最新バージョン) -> ARM64 (最新バージョン)

[Step 2]
Graviton でアプリケーション実行環境を構築

コンテナアプリケーションの実行環境の構築

ARM64 向けのコンテナイメージをビルドし、Graviton 環境でコンテナを実行

Dockerfile

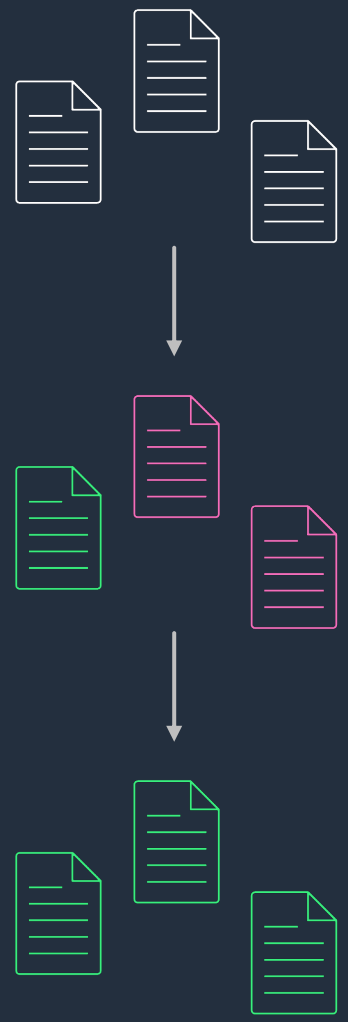
コンテナイメージ
(ARM64)

コンテナホスト
(Graviton)

[Step 3]
機能テスト / パフォーマンス テストの実施

機能テスト

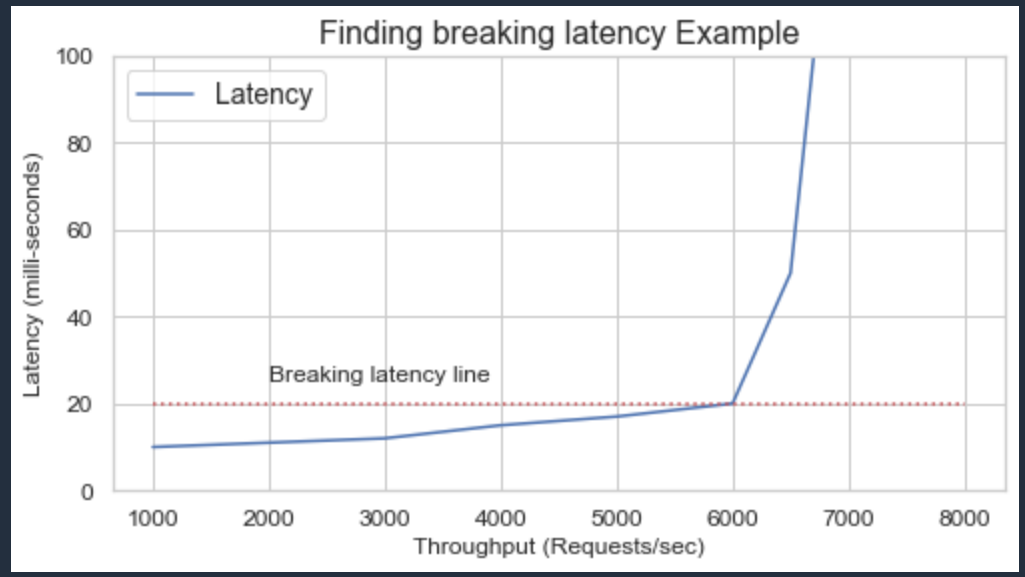
- 単体テスト、結合テスト、E2E テスト等がパスすることを確認
  - x86 環境と同じカバレッジを達成できるように
  - 自動テストが充実していると確認が容易に



- 多くのエラーは ARM64 移行に伴うソフトウェア変更に起因
  - 新規にインストールしたもの、バージョンアップデートしたもの
  - ソフトウェア変更箇所を少なくするために、アップデートは x86 環境での実施を推奨

パフォーマンステスト

- テストで評価したい項目を明確にする
  - (Bad) Java アプリケーションの性能比は？
  - (Good) p99 が 10 ms 以下で、XX のシナリオにおけるスループットの比較は？
- 負荷ツールで適切な負荷をかける
  - 大きめのインスタンスタイプ、分散環境の利用
  - 最大のスループットを測定、レイテンシやエラーレートが崩壊するポイントの特定



Flame Graph (on-CPU profiling)

プロファイラ (例: Linux perf) が取得したスタックを可視化

ボックス

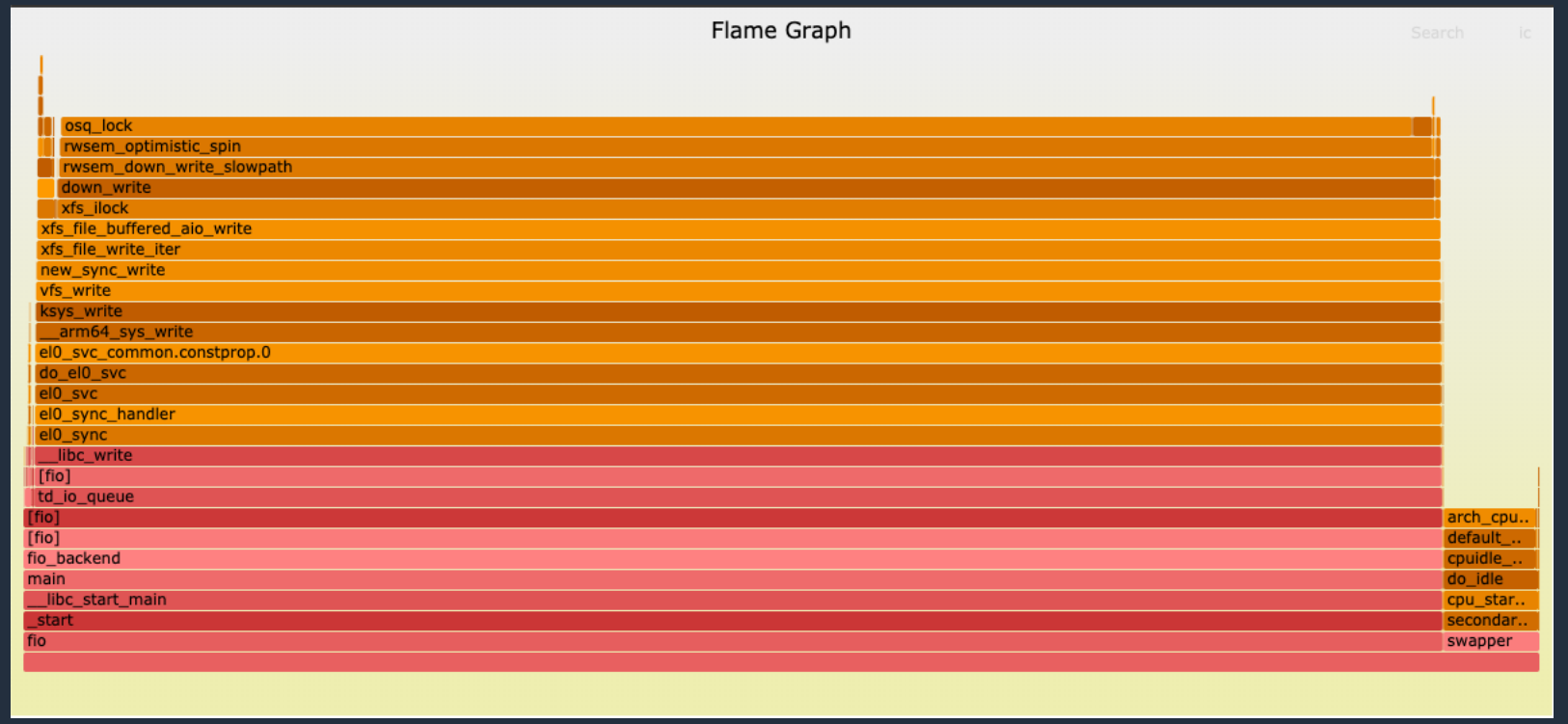
- 関数 (Stack frame)

X 軸 (ボックスの幅)

- 関数が on-CPU の時間
- 時系列ではない

Y 軸

- スタックの深さ
- 一番上のボックスが on-CPU だった関数



https://github.com/aws/aws-graviton-getting-started/blob/main/perfrunbook/debug_code_perf.md

[Step 4]
Graviton を利用するように
インフラストラクチャを更新

インフラストラクチャの更新手順

一般的な手順は次の通り:

- インスタンスタイプの変更
- AMI の更新
- Auto Scaling グループの設定変更

→ IaC (Infrastructure as Code) のコードを更新

- インスタンスの置き換え (x86 -> ARM64)
- トラフィックの移行

→ Blue / Green, Canary など ワークロード移行のデプロイを実施

まとめ

アプリケーション観点での Graviton 移行

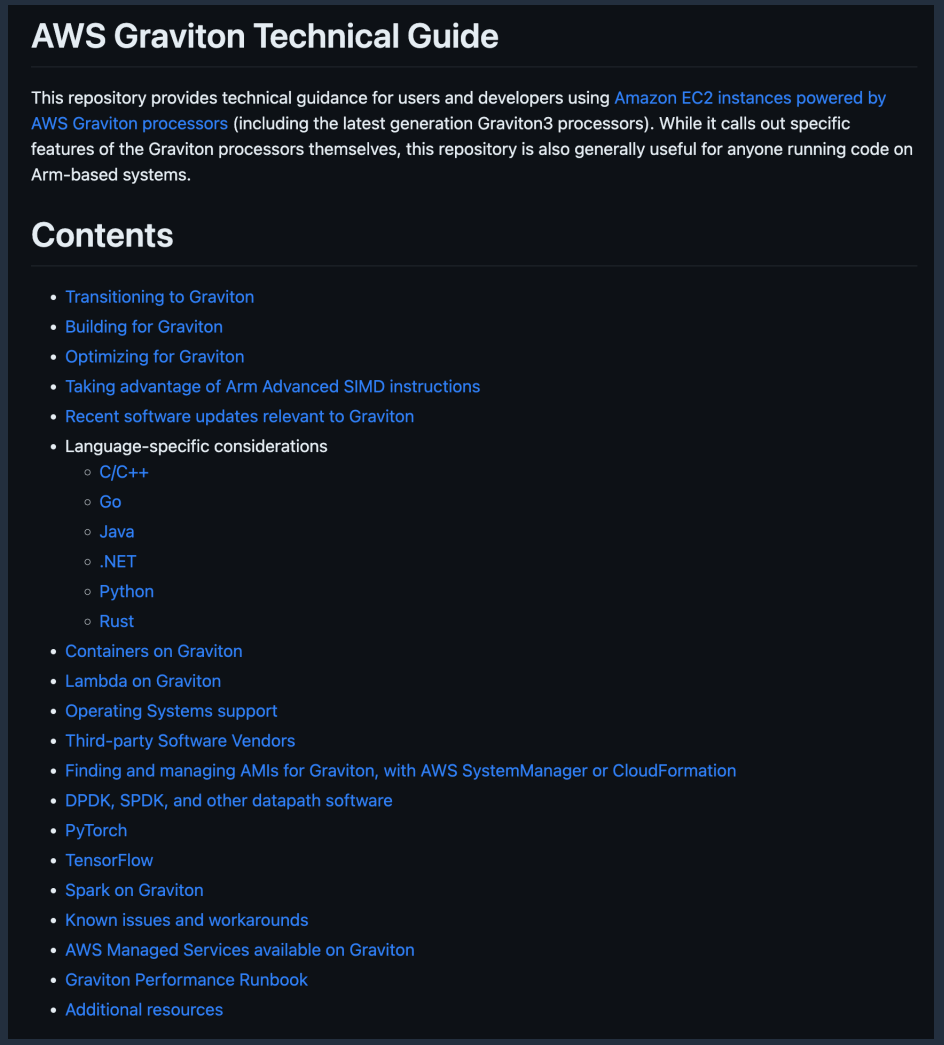
Graviton への移行ステップ

1. アプリケーションに関連するソフトウェアの状況を整理
2. Graviton でアプリケーション実行環境を構築
3. 機能テスト / パフォーマンステストの実施
4. Graviton を利用するようにインフラストラクチャを更新

移行 Tips

- 関連するソフトウェアのバージョンはできるだけ新しい方が望ましい
- 機能テストやパフォーマンステストの実施が容易になるほど、移行負荷が軽減
  - ➢ 自動テストの拡充、負荷テストの仕組み化

関連資料



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

https://github.com/aws/aws-graviton-getting-started/tree/main

Thank you!