AWS Black Belt Online Seminar

AWS Batch
入門編

小林 広志

Solutions Architect
2023/10

本セミナーの対象者

- HPC や機械学習などのバッチ処理をクラウド上で実行することについて興味のある方
- 次の AWS サービスの概要レベルの知識が前提になります
  - Amazon VPC / Amazon EC2 / Amazon S3 などの AWS 基礎サービス

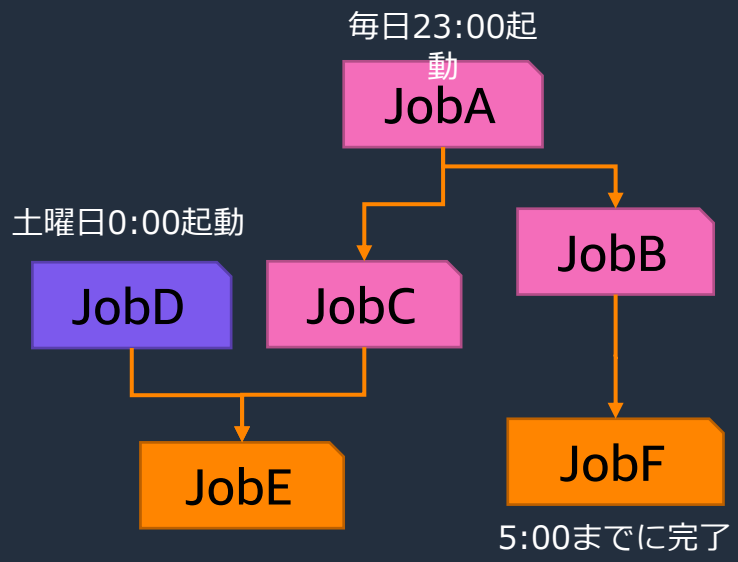
アジェンダ

1. バッチ処理とは
2. AWS Batch サービス概要
3. AWS Batch デザインパターン
4. まとめ

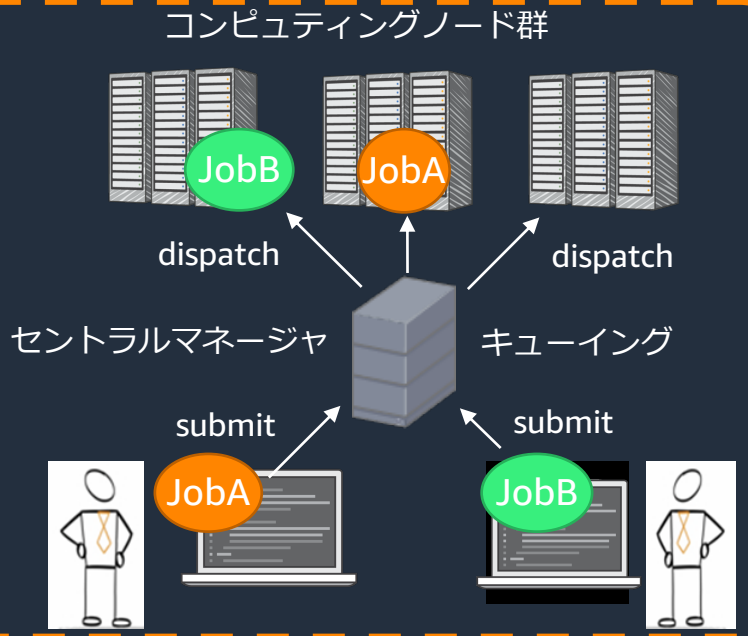
バッチ処理とは

似て非なる『バッチ処理』

A) 定型業務における『バッチ処理』
夜間バッチのように、予めジョブの起動や順序を定義しておいてジョブの実行・管理をするシステム（ジョブネット等）
ミドルウェア例：Hinemos、SOS JobScheduler etc.



B) 大規模計算における『バッチ処理』
スーパーコンピュータ等で行う大規模科学計算、メディア処理、CGレンダリング、機械学習における学習プロセス等アドホックな計算
ミドルウェア例：Slurm、PBS、LSF etc.



さまざま分野での大規模バッチ処理

自動運転の機械学習と
シミュレーション

遺伝子解析
新薬探索

ビッグデータ

機械学習

金融のリスク分析

流体解析
半導体設計

再生可能エネルギー
オイル・ガス探索

気象予報

大規模データ処理を支える共通基盤に AWS を活用。最大 30,000CPU コアの処理を実現し、ビジネスと開発の加速に貢献

ビジネス課題

- さまざまなサービスを運用するうえで、広告データ、ログデータ、属性データなどを利用したデータ分析は必要不可欠。社内組織や社内プロジェクトで利用可能なデータ処理基盤として『Crois』を開発。

ソリューション

- コンテナの実行基盤には AWS Batch と Amazon ECS on AWS Fargate を採用。機械学習のような CPU、メモリー、ディスクを大量に消費するタスクや GPU インスタンスを利用するタスクでは AWS Batch。データウェアハウスへのクエリーのような軽量なタスクには短時間でスケールができる Amazon ECS on AWS Fargate を利用

導入効果

- 大規模な処理を複数同時に実行することが可能で、最大 30,000CPU コアの処理に対応
- 『Crois』を導入することで、汎用モジュールの利用やユーザーが自作したモジュールの再利用などにより開発効率が上がり、リードタイムは 2 分の 1 に短縮。管理運用も AWS のマネージドサービスの利用により、年間で最大 10 人月程度の運用工数の削減の見込

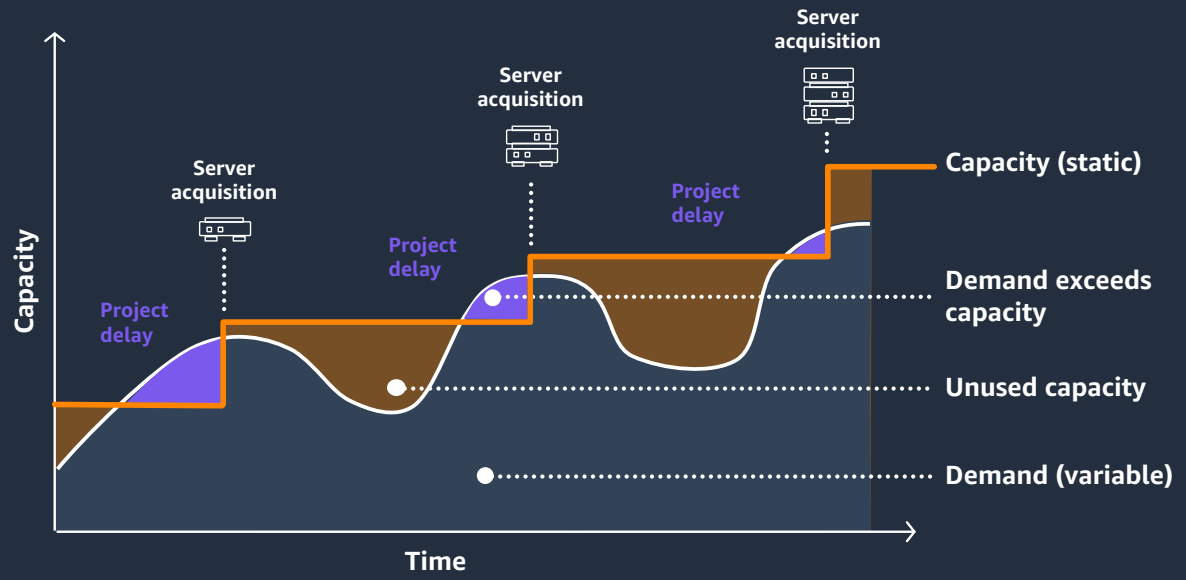
“ 社内向け大規模データ処理基盤『Crois』に AWS のマネージドサービスを活用することで、リクルートのビジネスに求められる高いアジリティとスケーラビリティを実現しています “

大石 壮吾 氏
株式会社リクルート　プロダクト統括本部 プロダクト開発統括室　データ推進室長

https://aws.amazon.com/jp/solutions/case-studies/recruit-case-study/

大規模バッチ処理の課題

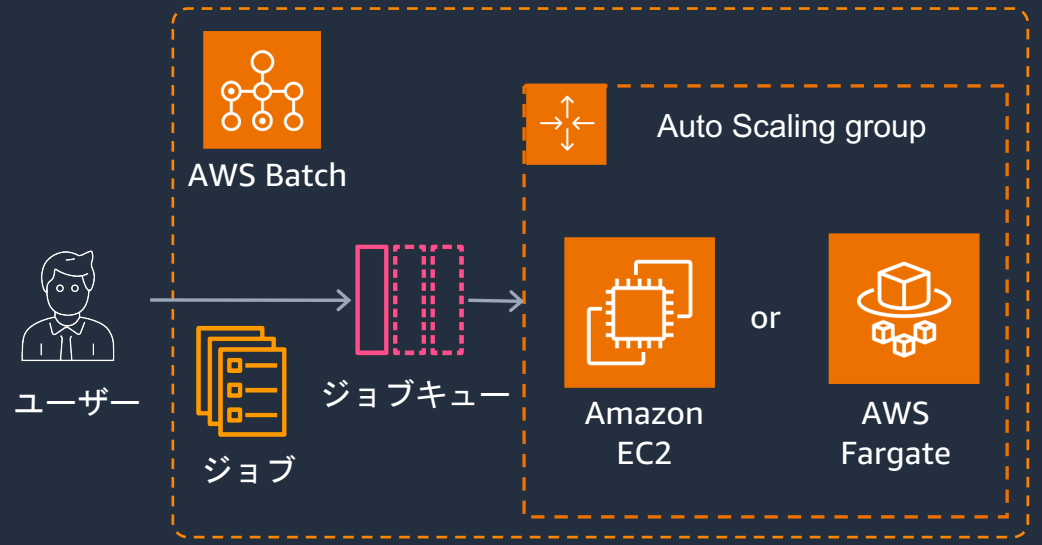
- 計算需要はビジネスの状況によって変動し、大規模な計算を行いたいことがあっても、計算資源不足で長いキュー待ちが発生
- HPC や機械学習のアプリケーションは、多くのパッケージやライブラリを必要とし、複雑な依存関係を管理していくのが困難。またアプリケーションごとに必要なリソースも異なる



AWS Batch

大規模バッチ処理のため環境をフルマネージドで提供

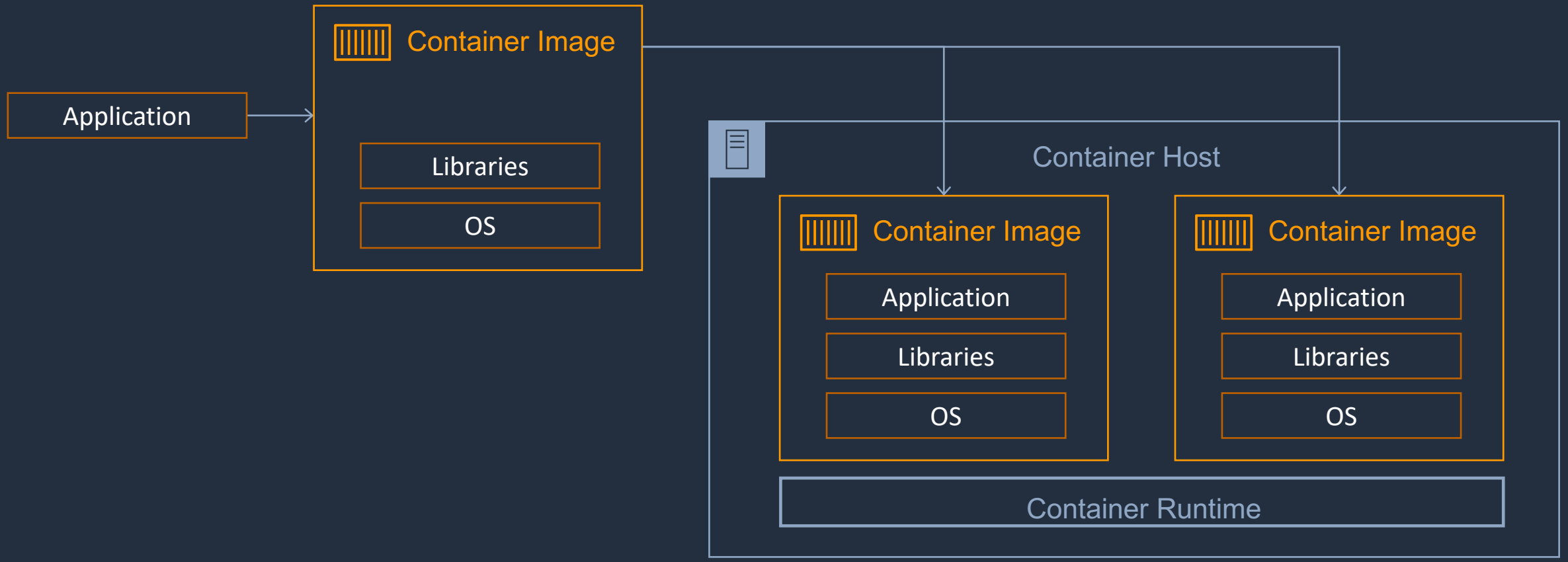
- AWS Batch がインスタンスの起動や停止を行うため、**スケジューラや計算ノードなどの管理が不要**
- ジョブは **Docker コンテナイメージ**を元に作成し、自動でスケールするコンピューティング環境で実行する
- コンピューティング環境ではインスタンスタイプや vCPU 数、スポットインスタンス利用有無などを任意に指定可能



コンテナイメージを用意するだけでスケーラブルな大規模バッチ処理環境が得られる

コンテナとは

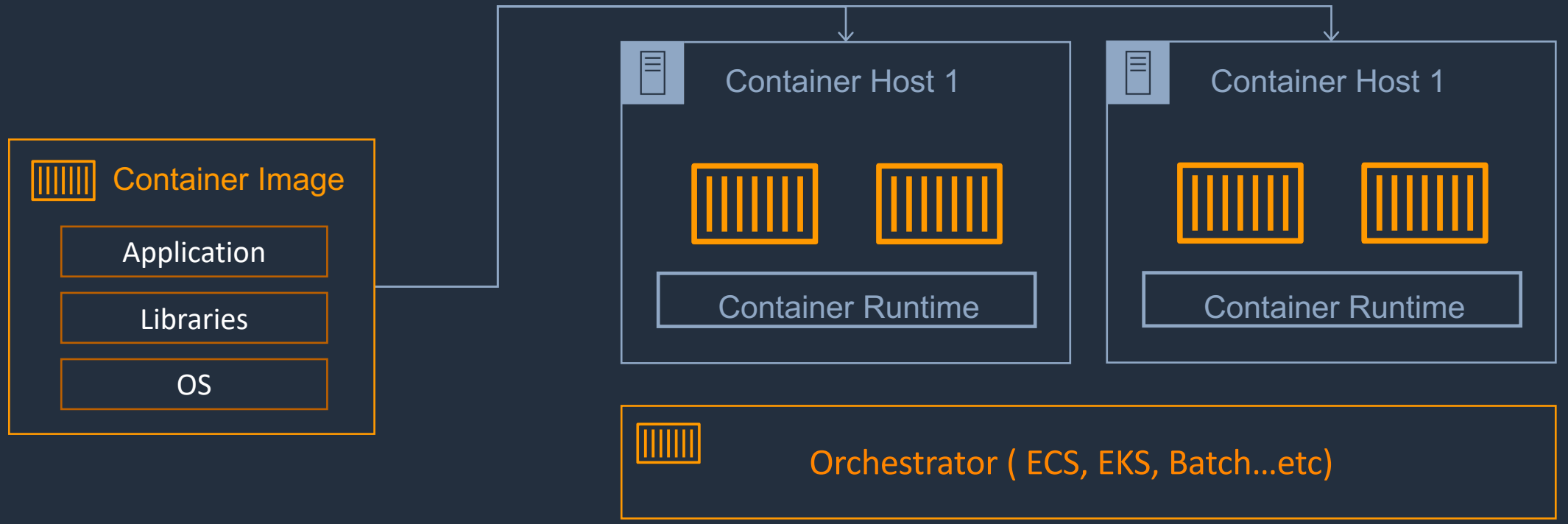
- アプリケーションが稼働するための依存関係をイメージにパッケージングして、マシンに配布して動かすための技術



コンテナ化のメリット

- 依存関係のカプセル化
- 複雑なデプロイメントの簡素化
- 可搬性

AWS Batch によりたくさんの計算機を使ってコンテナを動かし、大量の計算を実行



AWS Batch サービス概要

AWS Batch の機能と特徴

ジョブスケジューラ

リソース
オーケストレータ

フルマネージド

他のAWSサービス
との統合

最適化された
リソース提供

高い
コスト効率

大規模計算に対応

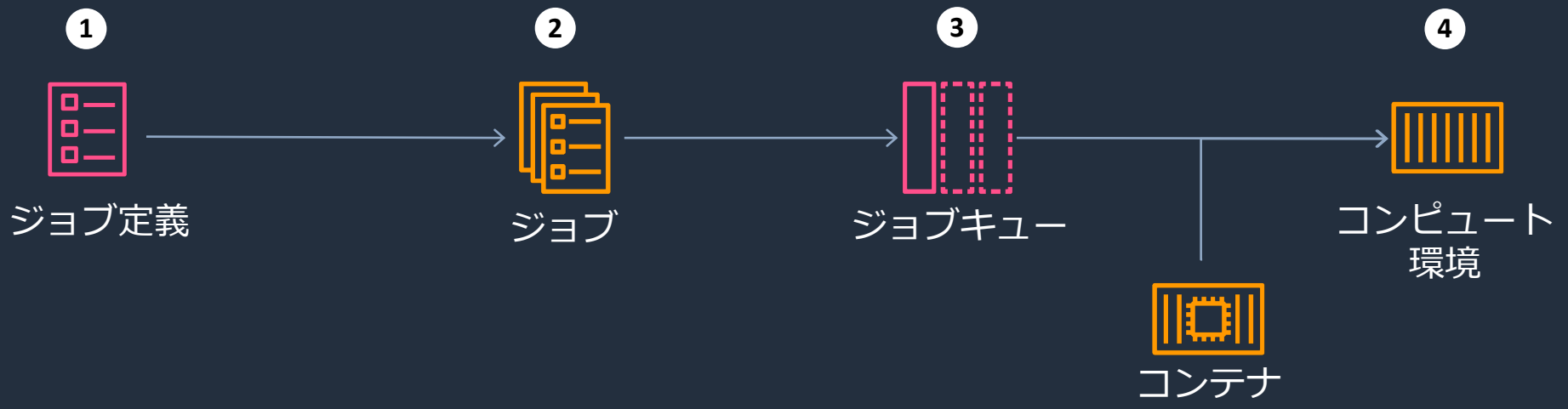
AWS Batch のキーコンポーネント

1. **ジョブ定義**
   ジョブの属性情報（コンテナイメージ、IAM ロール、vCPU、メモリ要件など）を持つテンプレート

2. **ジョブ**
   実際の計算作業
   各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能

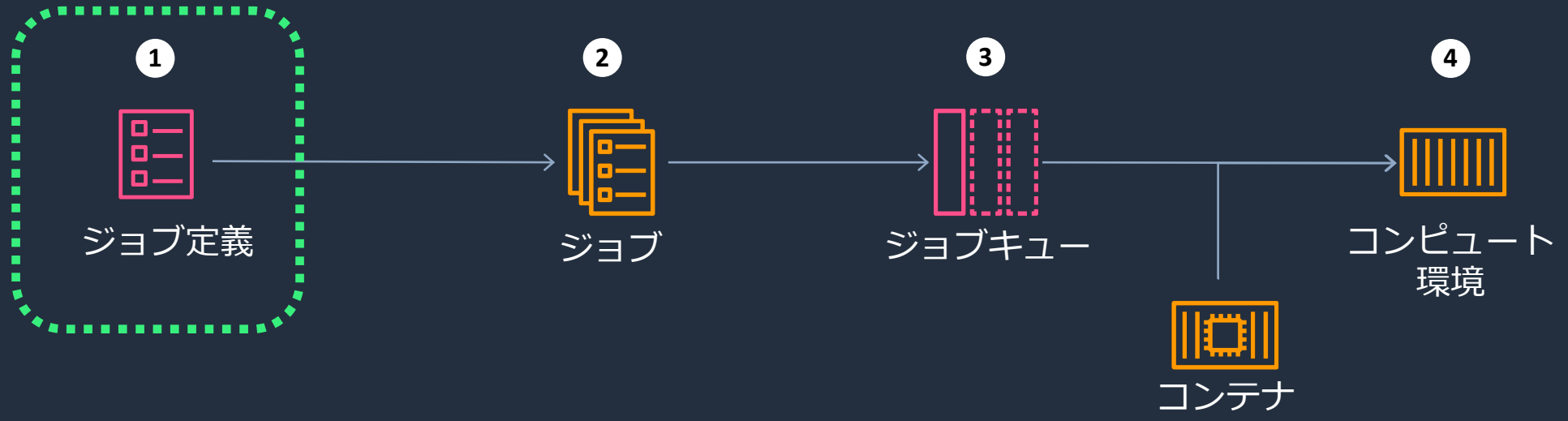
3. **ジョブキュー**
   キューは優先順位を決定し、各ジョブキューは1つ以上のコンピュート環境

4. **コンピュート環境 (CE)**
   オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



AWS Batch のキーコンポーネント

1. **ジョブ定義**
   ジョブの属性情報（コンテナイメージ、IAM ロール、vCPU、メモリ要件など）を持つテンプレート
2. **ジョブ**
   実際の計算作業
   各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能
3. **ジョブキュー**
   キューは優先順位を決定し、各ジョブキューは1つ以上のコンピート環境
4. **コンピート環境 (CE)**
   オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



ジョブ定義

ジョブ定義は、ジョブ実行のテンプレート

代表的な設定項目
- ジョブのコンテナで使用する Docker イメージ
- コンテナで使用する vCPU の数とメモリの量
- コンテナにマウントするボリュームとパス
- コンテナの開始時に渡す環境変数
- ジョブで AWS アクセス許可の取得に使用する IAM ロール
- ジョブのリトライ上限

各ジョブはジョブ定義を参照する必要があるが、多くのパラメータはジョブ投入時に上書き可能

https://docs.aws.amazon.com/batch/latest/userguide/job_definitions.html

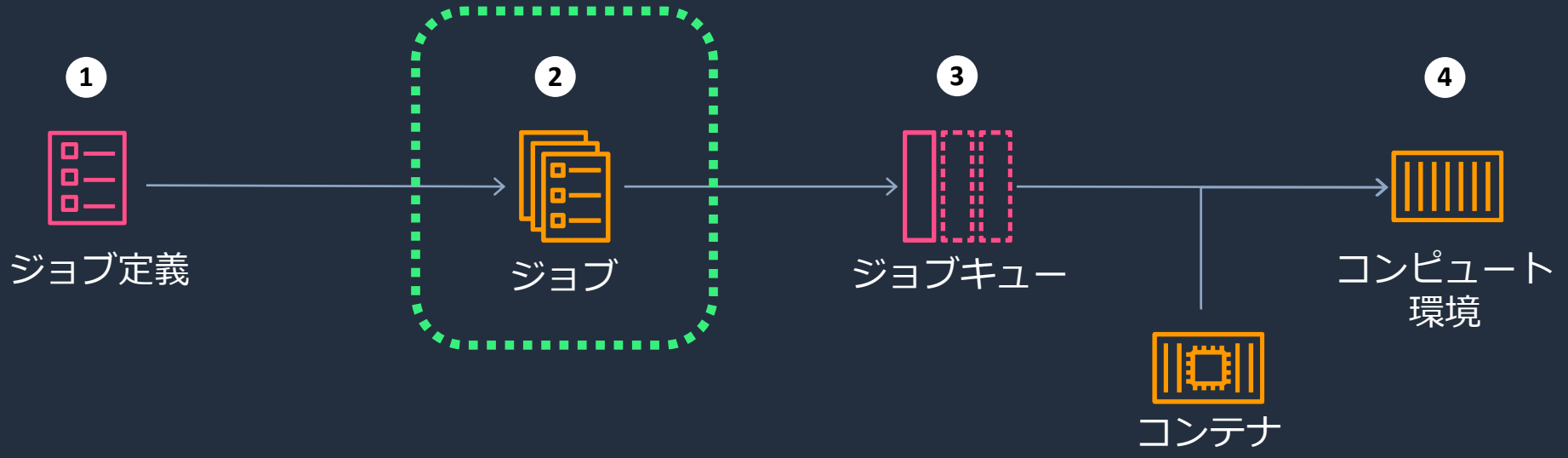
AWS Batch のキーコンポーネント

1. ジョブ定義
ジョブの属性情報（コンテナイメージ、IAMロール、vCPU、メモリ要件など）を持つテンプレート

2. ジョブ
実際の計算作業
各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能

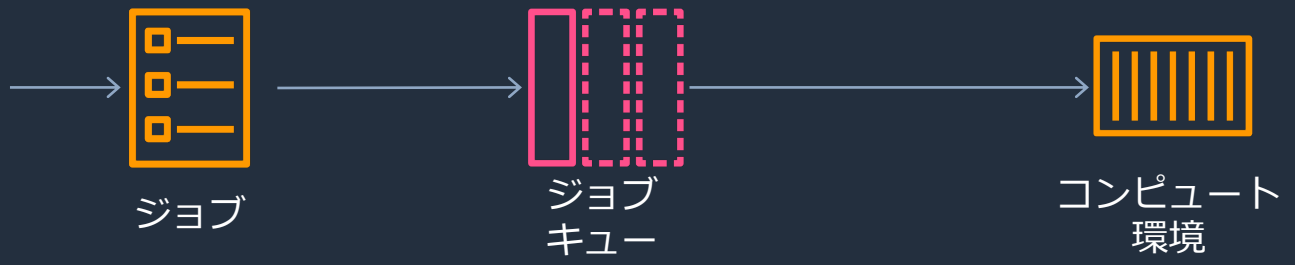
3. ジョブキュー
キューは優先順位を決定し、各ジョブキューは1つ以上のコンピュート環境

4. コンピュート環境 (CE)
オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



ジョブ

- ジョブは AWS Batch で実行される計算の作業単位、コンテナ化されたアプリケーション
- ジョブ投入時やジョブ定義で定義された、CPU、GPU、メモリなどのパラメータによって、計算に使われるインスタンスタイプが選択される
- ジョブのタイプ
  - スタンダード：1つづつ投入されるジョブ
  - 配列ジョブ：共有するパラメータを持ち、グループで実行されるジョブ
  - マルチノード並列(Multi-Node Parallel=MNP)：MPI or NCCL、複数ノード上で実行されるジョブ
- ジョブ間の依存関係



https://docs.aws.amazon.com/batch/latest/userguide/jobs.html

ジョブの状態

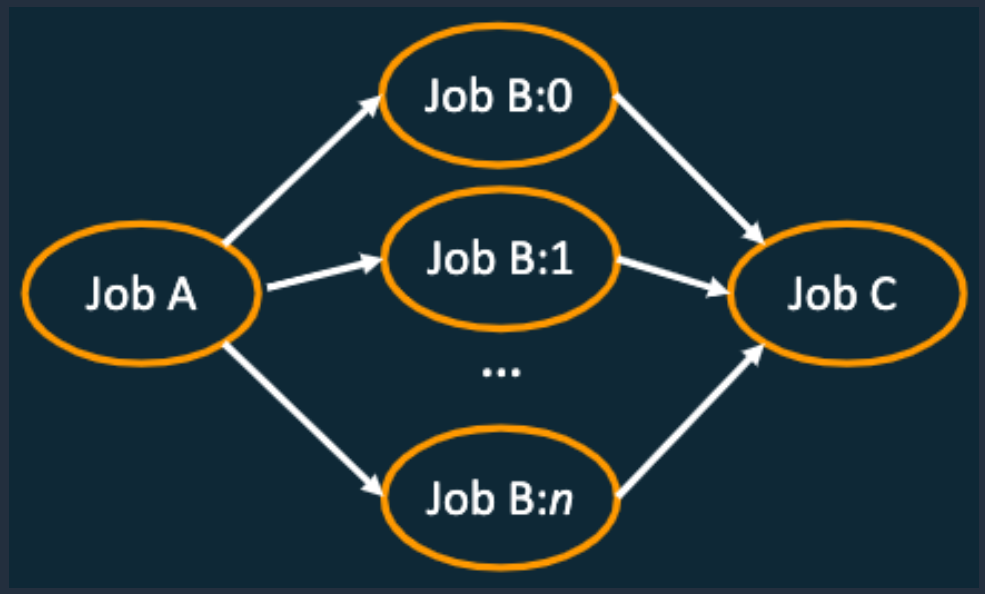
- SUBMITTED: キューに送信されたが、まだスケジューラによって評価されていない
- PENDING: ジョブはキュー内にあり、別のジョブやリソースへの依存関係があるため、まだ実行できていない
- RUNNABLE: スケジューラの評価が完了し、ホストにスケジューリングされる準備ができて
- STARTING: ジョブはホストにスケジュールされており、関連するコンテナ初期化操作が進行中
- RUNNING: ジョブが実行中
- SUCCEEDED: 終了コード 0 でジョブが正常に終了
- FAILED: 利用可能な再試行すべてで、終了コード 0 以外でジョブが終了

配列ジョブ

ジョブ定義、vCPU、メモリなどの共通パラメータを共有するジョブです。複数のホストに分散されたり、同時に実行されたりする可能性のある、関連しているが個別の基本ジョブの集合として実行されます。

たくさんのパラメータを持つジョブを実行するのに効率的

- パラメータスウィープ
- モンテカルロシミュレーション
- 大量の計算対象オブジェクトを持つ計算



https://docs.aws.amazon.com/batch/latest/userguide/array_jobs.html

ワークフロー

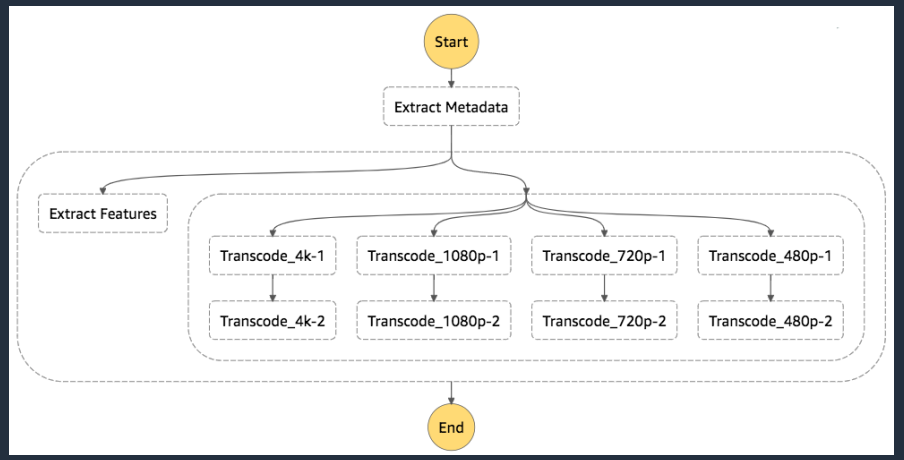
ワークフローツール

AWS Step Functions

ステートマシンという単位で記述された一連の個別ステップを実行するワークフローオーケストレーター

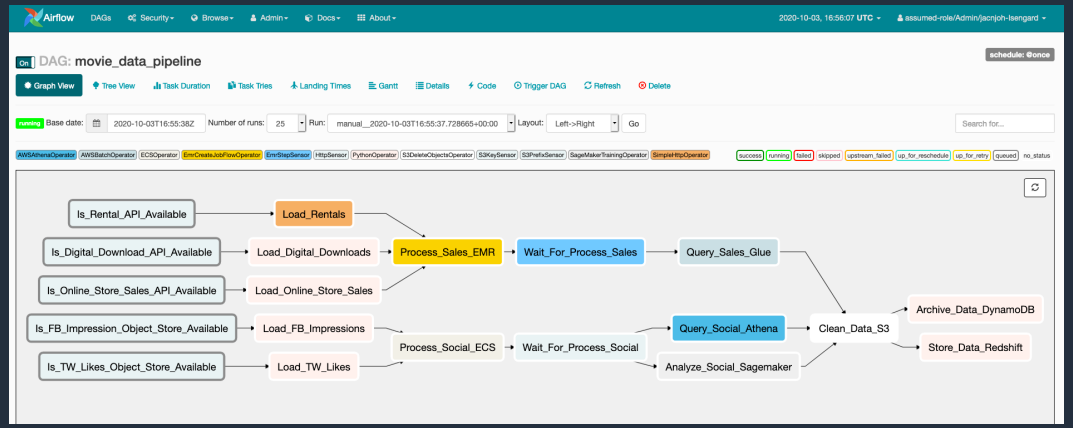
タスク　並列処理

条件分岐処理　コールバック待機



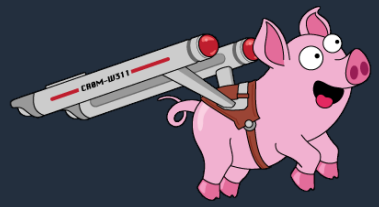
Amazon Managed Workflows for Apache Airflow (MWAA)

パイプラインのセットアップと運用を行う Apache Airflow 用のマネージドワークフローオーケストレーションサービス



3rd party ワークフローツールとの統合

3rd Party
Open Source



Cromwell

Airflow

Nextflow

Luigi

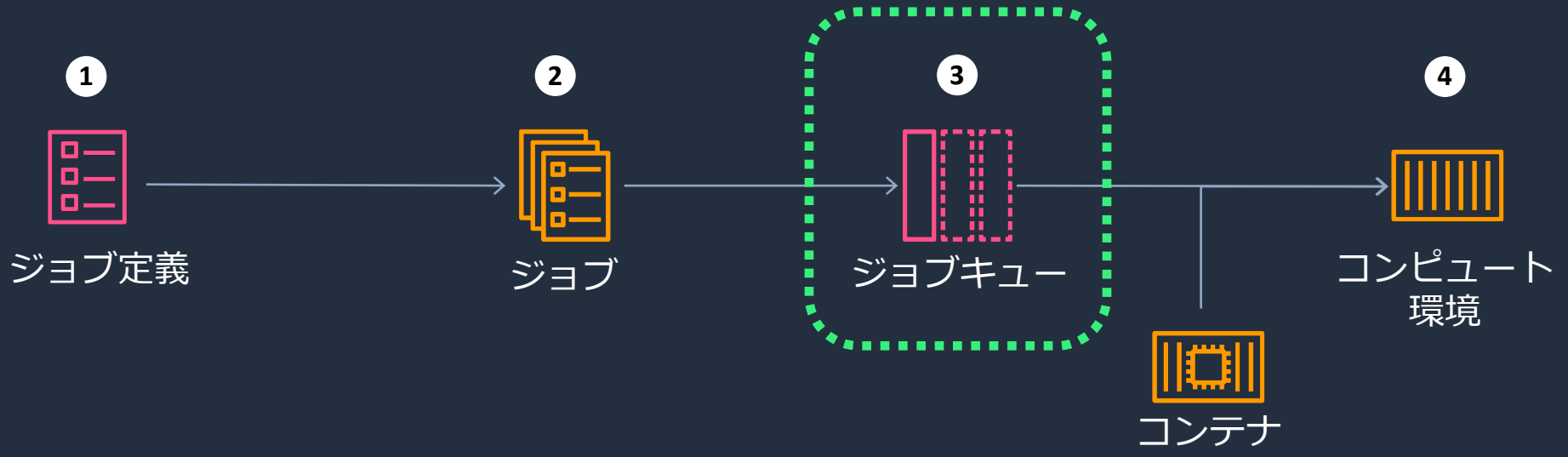
AWS Batch のキーコンポーネント

1. ジョブ定義
ジョブの属性情報（コンテナイメージ、IAMロール、vCPU、メモリ要件など）を持つテンプレート

2. ジョブ
実際の計算作業
各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能

3. ジョブキュー
キューは優先順位を決定し、各ジョブキューは1つ以上のコンピュート環境

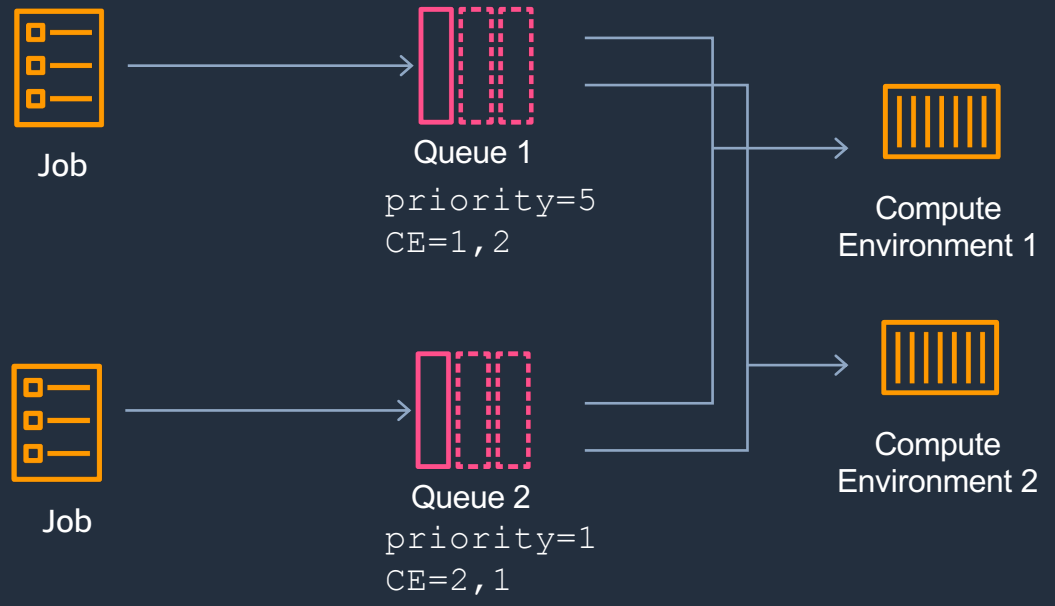
4. コンピュート環境 (CE)
オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



ジョブキュー

ジョブキューはジョブが投入された後、実際にコンピューティング環境で実行するようにスケジュールされるまでのジョブの待機場所

- キューはひとつまたは複数のコンピュート環境に接続可能。
  またコンピュート環境は複数のキューで共有可能
- 設定項目
  - 優先度: 複数のキューで共有されるコンピュート環境に
    ジョブを割り当てる優先度を評価（数が大きいと優先度高）
  - スケジューリングポリシー: FIFO (デフォルト) or Fair-Share
  - 状態: ENABLED or DISABLED
  - コンピュート環境の順番: 利用されるコンピュート環境
    の順番（数が小さい方が優先的に使われる）
  - タグ: キューの分類と整理



https://docs.aws.amazon.com/batch/latest/userguide/job_queues.html

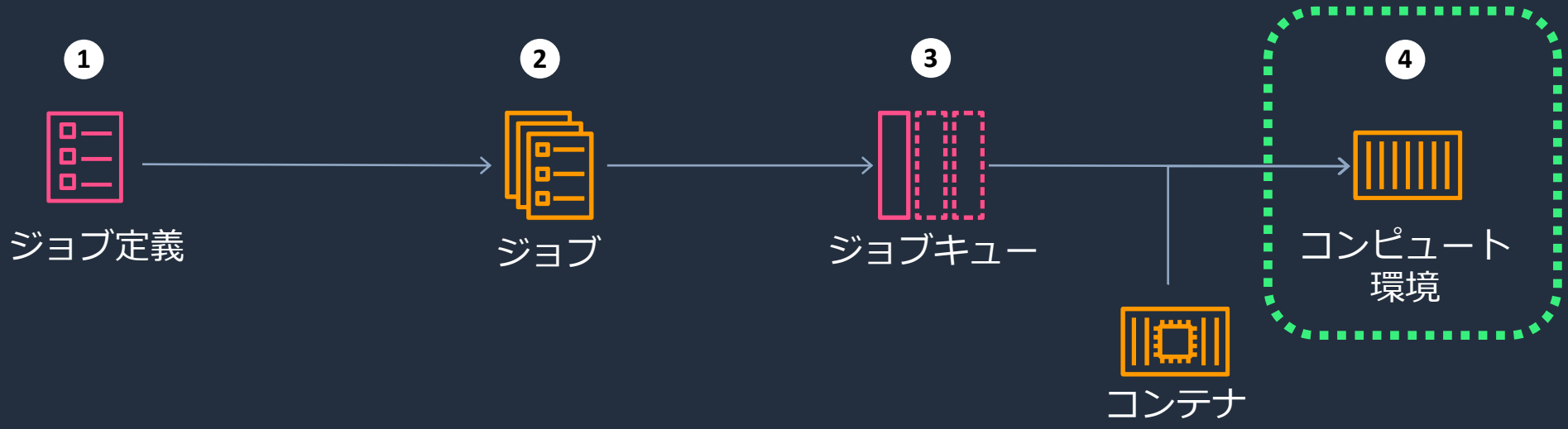
AWS Batch のキーコンポーネント

1. ジョブ定義
ジョブの属性情報（コンテナイメージ、IAMロール、vCPU、メモリ要件など）を持つテンプレート

2. ジョブ
実際の計算作業
各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能

3. ジョブキュー
キューは優先順位を決定し、各ジョブキューは1つ以上のコンピート環境

4. コンピート環境 (CE)
オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



コンピュート環境 (Compute Environment = CE)

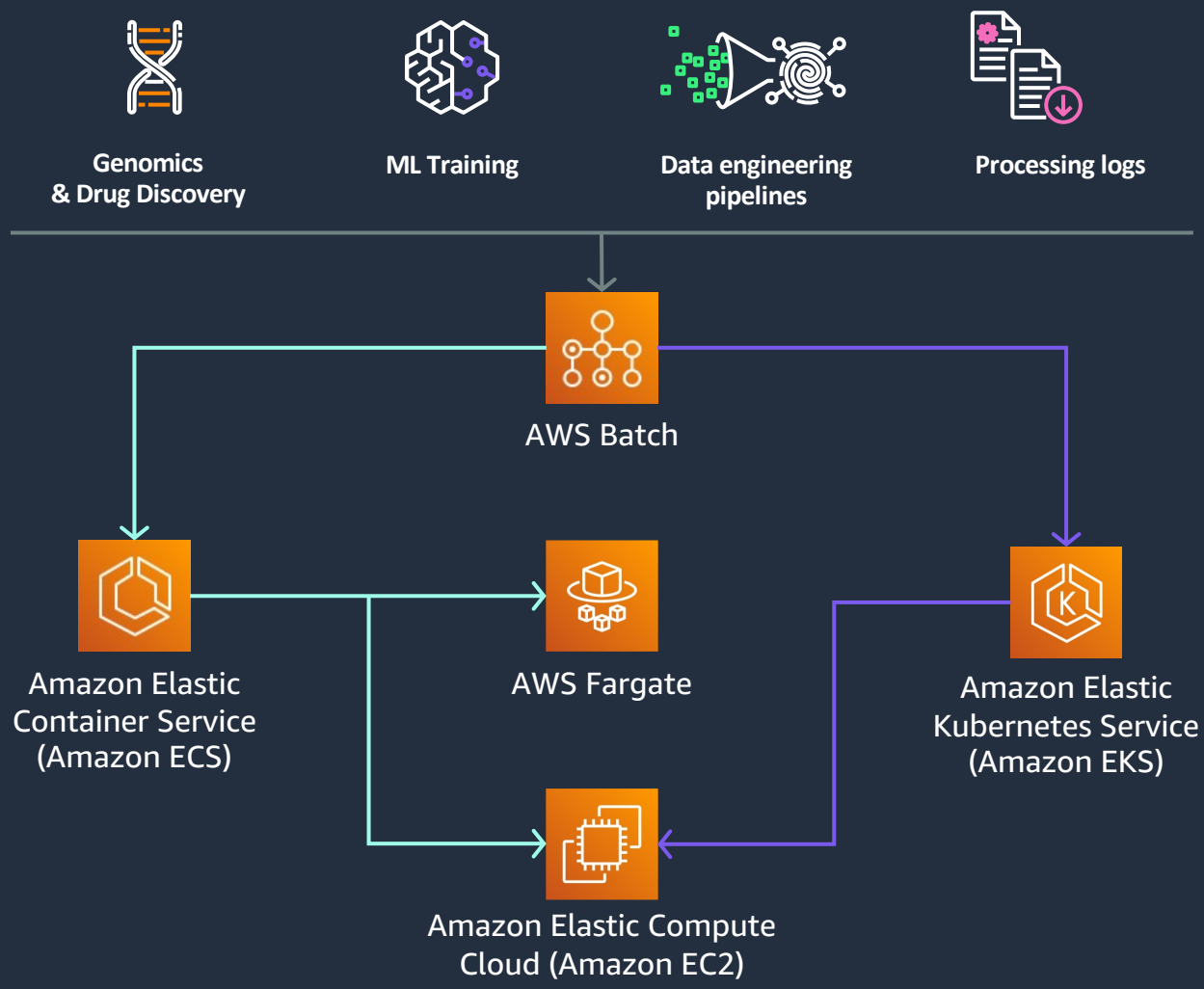
AWS が自動でスケールする計算環境を提供

代表的な設定項目:

- ECS / EKS
- オンデマンド / スポット
- インスタンスタイプ
- 最小 / 最大 vCPU
- Amazon Machine Image (AMI)
- 配分戦略

https://docs.aws.amazon.com/batch/latest/userguide/compute_environments.html

AWS Batch – コンテナプラットフォーム



Container Orchestration

- Amazon ECS or Amazon EKS

Fargate

- サーバレスな計算環境
- コンテナの起動が早く、ジョブの開始が早い
- 1つのジョブで使用可能リソース上限は
  16vCPU 128GiB memory

EC2

- 全てのインスタンスタイプが利用可能
- EC2 のカスタマイズも可能
- 超大規模計算にも対応

https://docs.aws.amazon.com/batch/latest/userguide/eks.html

EC2 Spot Instances are how you can do more for less

On-Demand

Pay for compute capacity by the second with no long-term commitments

Spiky workloads, to define needs

Savings Plans and Reserved

Make a 1 or 3 year commitment and receive a significant discount off On-Demand prices

Committed and steady-state usage

Spot Instances

Spare Amazon EC2 capacity at savings of up to 90% off On-Demand prices

Fault-tolerant, flexible, stateless workloads

Spot is more than cost savings…

More Compute  + Faster Results  = Accelerated Innovation 

aws

配分戦略

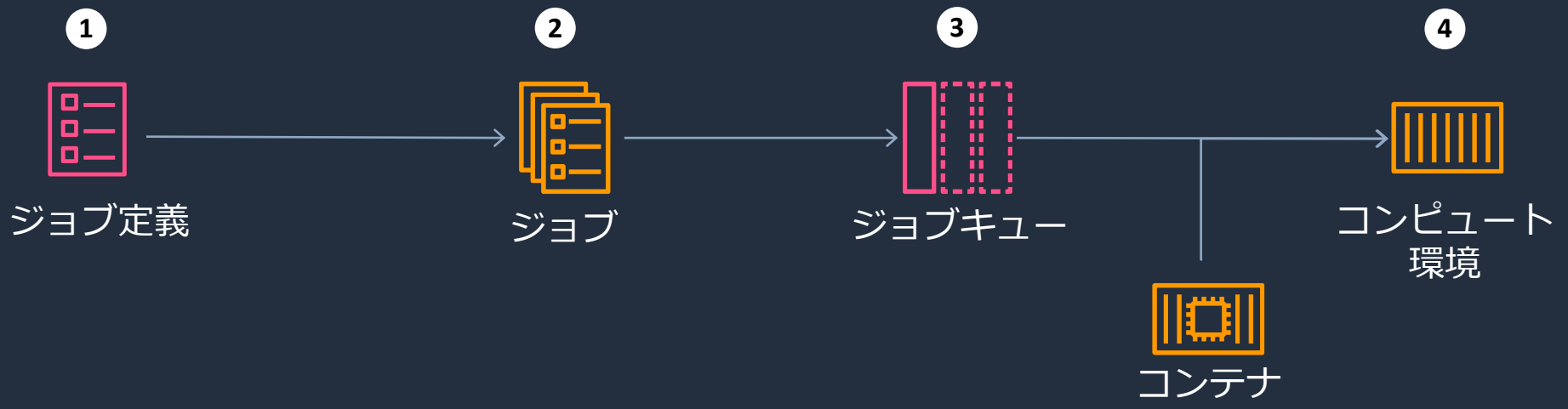
AWS Batch はジョブのニーズに最も適したインスタンスタイプを、指定された インスタンスタイプから選択します。配分戦略は、AWS Batch が追加のキャパシティーを必要とする場合の動作を定義します。
(Fargate リソースで実行されるジョブには適用されません)

- BEST_FIT (デフォルト)
  最もコストの低いインスタンスタイプを優先して、ジョブのニーズに最も適したインスタンスタイプを選択します。選択したインスタンスタイプの追加インスタンスが利用できない場合、AWS Batch は追加インスタンスが利用可能になるまで待機します。
- BEST_FIT_PROGRESSIVE
  キュー内のジョブの要件を満たすのに十分な大きさのインスタントタイプを追加で選択します。単位 vCPU あたりのコストが低いインスタンスタイプが優先されます。
- SPOT_CAPACITY_OPTIMIZED (スポットインスタンス向け)
  キュー内のジョブの要件を満たすのに十分な大きさのインスタンスタイプを 1 つ以上選択します。中断する可能性が低いインスタンスタイプが優先されます。
- SPOT_PRICE_CAPACITY_OPTIMIZED (スポットインスタンス向け)
  価格とキャパシティの両方を考慮して、中断される可能性が最も低く、価格ができるだけ低いスポットインスタンスプールを選択します。この配分戦略は、スポットインスタンスのコンピューティングリソースでのみ使用できます。

https://docs.aws.amazon.com/batch/latest/userguide/allocation-strategies.html

AWS Batch のキーコンポーネント (再掲)

1. **ジョブ定義**
   ジョブの属性情報（コンテナイメージ、IAM ロール、vCPU、メモリ要件など）を持つテンプレート
2. **ジョブ**
   実際の計算作業
   各ジョブはジョブ定義を参照するが、多くのパラメータはサブミットの時にオーバーライド可能
3. **ジョブキュー**
   キューは優先順位を決定し、各ジョブキューは1つ以上のコンピート環境
4. **コンピート環境 (CE)**
   オンデマンドとスポット、その他のインスタンス・タイプを定義。CEは複数のジョブキューに接続可能



AWS Batch デザインパターン

パターン1：大量パラメータ探索

ワークロードの特徴

- 小規模な処理を大量に実行するために、複数のノードで独立したジョブを処理する
- 基本的にノード間の通信が発生しないため、スケールさせやすい

- 例 ：パラメータ最適化、創薬ドッキングシミュレーション、半導体設計シミュレーション (Electronic Design Automation)

計算環境構築の注意点

- 広大な計算領域をカバーするための
  - 計算資源のキャパシティ確保
  - 計算資源のコスト最適化

利用サービス

- Amazon Elastic Compute Cloud (Amazon EC2), Amazon EC2 Spot instances, AWS Batch

Arm Accelerates Speed to Market by Migrating EDA Workflows to AWS

arm

Challenge

Arm wanted to modernize its offerings for intellectual property design because its on-premises infrastructure could not grow with the pace of its engineering requirements.

Solution

Arm uses AWS Batch and Amazon EC2 Spot Instances to optimize its compute—decreasing the turnaround time for verification jobs, increasing engineer productivity, and accelerating product speed to market.

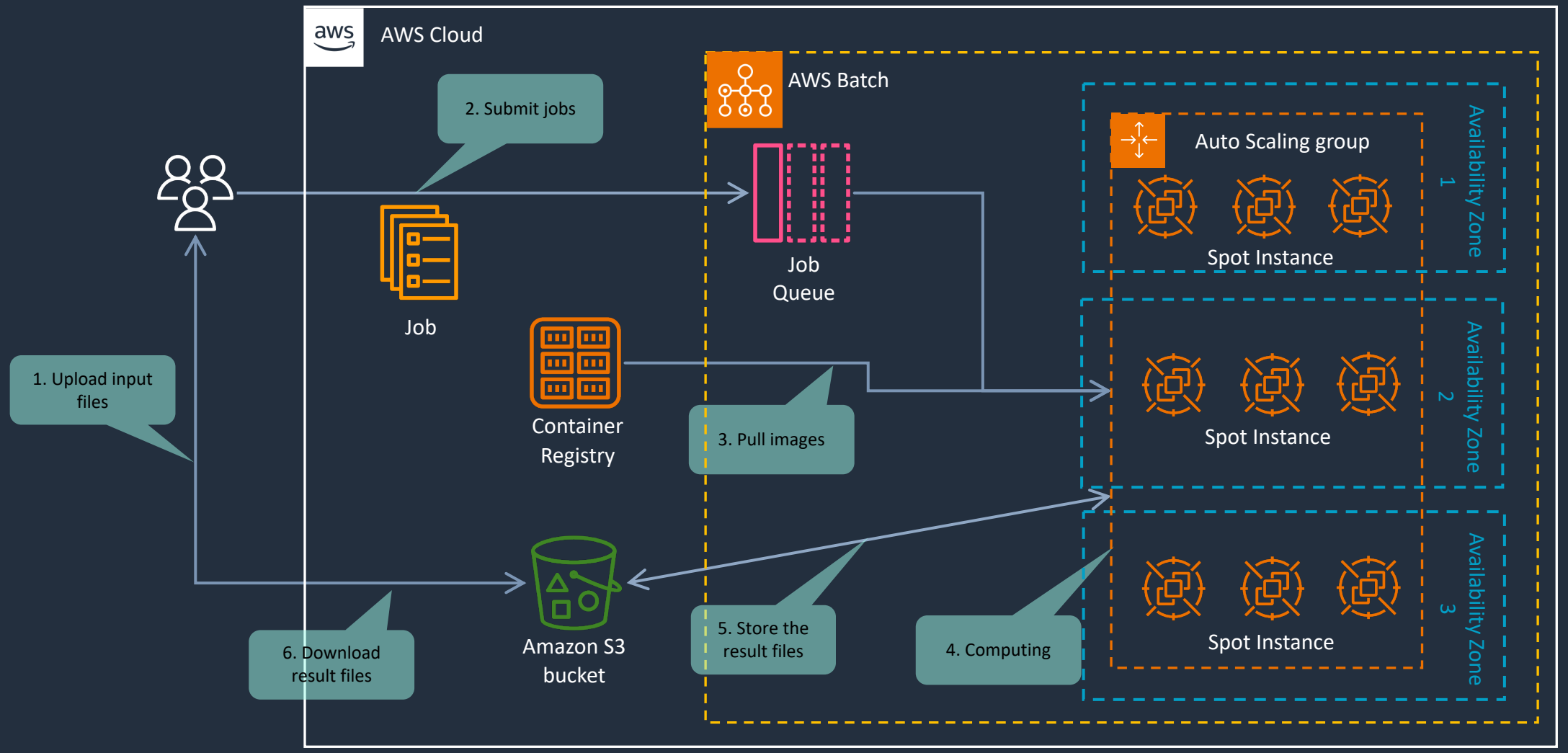
Benefits

- Can run more than 53 million jobs per week
- Scaled up to 350,000 virtual CPUs
- Decreased turnaround time for verification jobs

"Using AWS Batch facilitates selecting different instance types and mixing them together. That helps us to achieve the scalability that we need."

—**Zhifeng Yun**, technical director, Arm Limited

大量パラメータ探索 アーキテクチャ



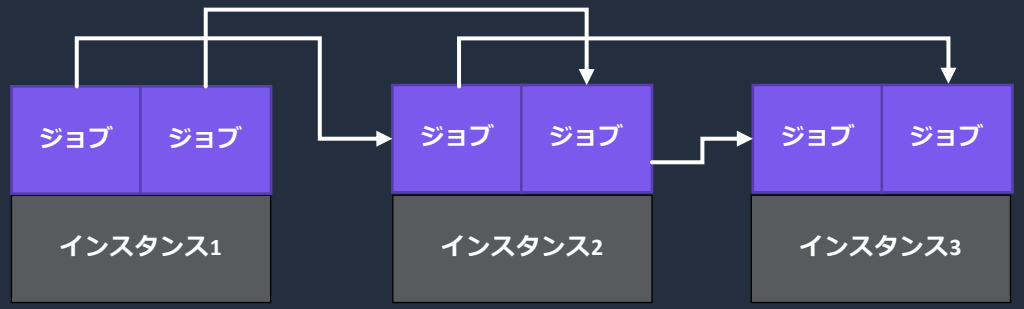
参考：AWS 秋の Amazon EC2 Deep Dive 祭り 2022

パターン２：複雑ワークフロー

ワークロードの特徴

- 小規模な処理を大量に実行するために、複数のノードで独立したジョブを処理する
- 基本的にノード間の通信が発生しないため、スケールさせやすい
- ただし複雑なワークフローを持つ

- 例：ゲノム解析、金融リスク計算



計算環境構築の注意点

- 複雑なワークフローの管理とそれに付随する計算インスタンスの管理

利用サービス

- Amazon EC2, AWS Batch, AWS Step Functions, Amazon Managed Workflows for Apache Airflow
- 3rd パーティのワークフローツール

Fred Hutch performs 7 years of compute time in 7 days

CHALLENGE

Accelerate research processes on AWS toward developing therapeutics to fight cancer. Analyzing and processing an immense number of whole genome datasets and translating gigabytes of raw microbiome genomic data into insights about which specific microbes are present in a person is a computationally intensive task requiring highly scalable technology.

SOLUTION

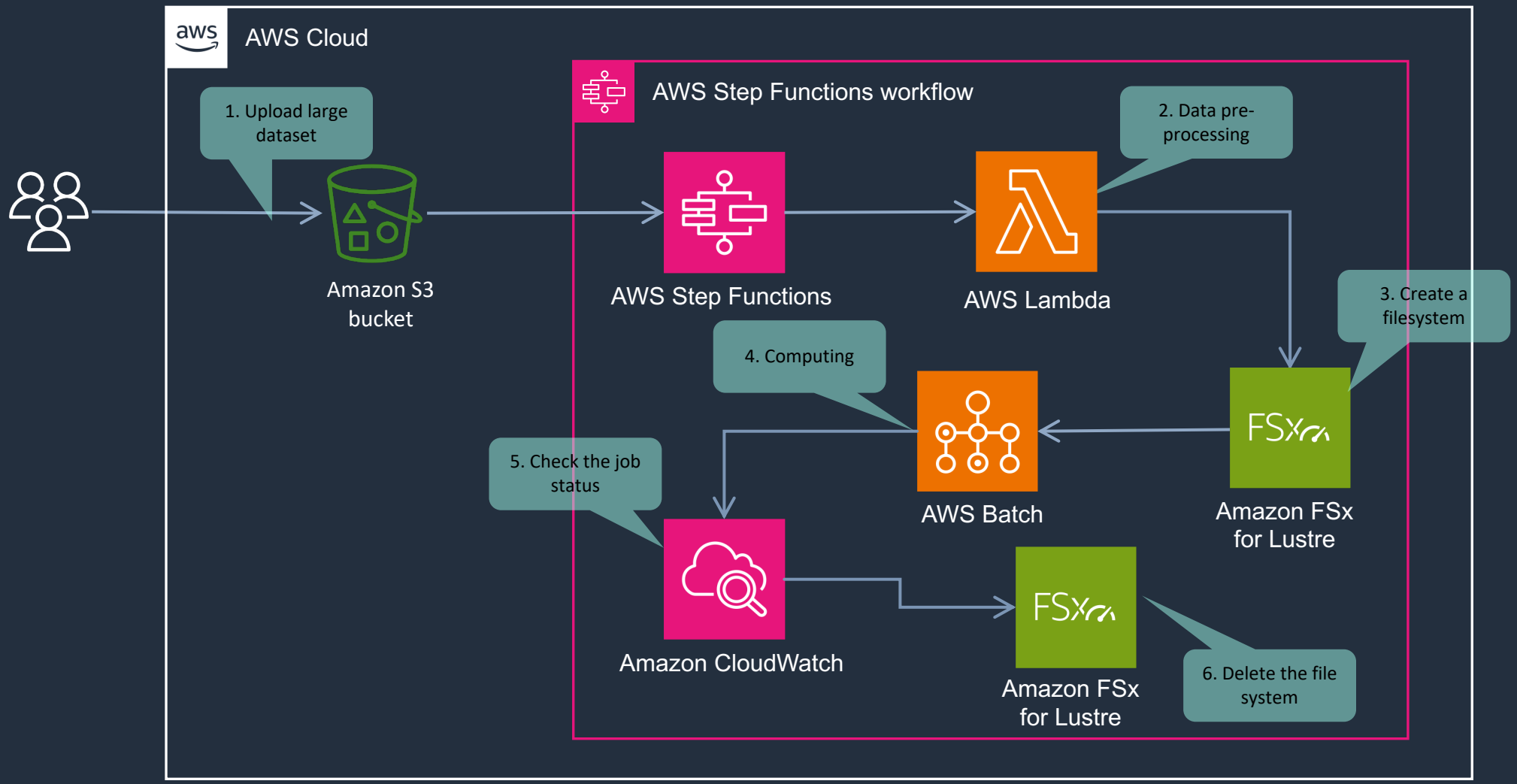
Their HPC platform accelerates processing time using Amazon EC2 instances to run computational workloads and Amazon S3 to store research data. Researchers execute computational analysis using the Nextflow framework to orchestrate AWS Batch which integrates well with Nextflow.

OUTCOME

- ✓ Fred Hutch processes data from 10,000 biological samples
- ✓ With AWS Batch and Spot Instances, they decreased the time it took to analyze this data from 7 years to 7 days

KEY SERVICE(S): AWS Batch, Amazon EC2, Amazon EC2 Spot instances

複雑ワークフロー アーキテクチャ



まとめ

- AWS Batch を利用することで、フルマネージドでスケーラブルなバッチ処理環境を簡単に構築
- スポットインスタンスを有効活用し、コスト効率よく大量の計算を実行したり。AWS Step Functions や 3rd party のワークフローツールを利用することで複雑なワークフローを実現
- HPC、ゲノム解析、機械学習など様々なワークロードで活用

AWS Batch Tips

- Fair share scheduling
  - https://aws.amazon.com/blogs/hpc/deep-dive-on-fair-share-scheduling-in-aws-batch/
- Fetch & Run
  - https://aws.amazon.com/blogs/compute/creating-a-simple-fetch-and-run-aws-batch-job/
- Multi-Node Parallel
  - https://docs.aws.amazon.com/ja_jp/batch/latest/userguide/multi-node-parallel-jobs.html
- How to use FSx for Lustre with AWS Batch
  - https://github.com/aws-samples/aws-batch-cloudformation-examples/blob/feature/example_fsx_for_lustre/examples/ec2/amazon-fsx-lustre/README.md
  - https://docs.aws.amazon.com/ja_jp/batch/latest/userguide/launch-templates.html
- AWS Fargate の AWS Batch で CLI/SDK の Linux ARM64 コンテナと Windows x86 コンテナのサポートを開始
  - https://aws.amazon.com/jp/about-aws/whats-new/2023/07/aws-batch-fargate-linux-arm64-windows-x86-containers-cli-sdk/
- ジョブの fail を Amazon SNS で通知
  - https://docs.aws.amazon.com/batch/latest/userguide/batch_sns_tutorial.html
- AWS Batch コンソールにダッシュボードのカスタマイズ機能を追加
  - https://aws.amazon.com/jp/about-aws/whats-new/2023/05/aws-batch-dashboard-customization-console/
- ベストプラクティス
  - https://docs.aws.amazon.com/batch/latest/userguide/best-practices.html

Thank you!

内容についての注意点

- 本資料では 2023 年 10 月時点のサービス内容および価格についてご説明しています。AWS のサービスは常にアップデートを続けているため、最新の情報は AWS 公式ウェブサイト (https://aws.amazon.com/) にてご確認ください
- 資料作成には十分注意しておりますが、資料内の価格と AWS 公式ウェブサイト記載の価格に相違があった場合、AWS 公式ウェブサイトの価格を優先とさせていただきます
- 価格は税抜表記となっています。日本居住者のお客様には別途消費税をご請求させていただきます
- 技術的な内容に関しましては、有料の AWS サポート窓口へお問い合わせください
- 料金面でのお問い合わせに関しましては、カスタマーサポート窓口へお問い合わせください (マネジメントコンソールへのログインが必要です)

AWS Black Belt Online Seminar とは

- 「サービス別」「ソリューション別」「業種別」などのテーマに分け、アマゾン ウェブ サービス ジャパン合同会社が提供するオンラインセミナーシリーズです
- AWS の技術担当者が、 AWS の各サービスやソリューションについてテーマごとに動画を公開します
- 以下の URL より、過去のセミナー含めた資料などをダウンロードすることができます
  - https://aws.amazon.com/jp/aws-jp-introduction/aws-jp-webinar-service-cut/
  - https://www.youtube.com/playlist?list=PLzWGOASvSx6FIwIC2X1nObr1KcMCBBlqY

ご感想は X(旧Twitter) へ！ハッシュタグは以下をご利用ください
#awsblackbelt