ちょっぴりDive Deep

~ AWS Step Functions Distribted Map ~

Naoto Oiso

Solutions Architect
Amazon Web Services

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

自己紹介

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

自己紹介

- 名前
  - **オオイソ ナオト**

- 好きな食べもの
  - 肉・寿司・ラーメン

- 好きなこと
  - お酒・サウナ・マンガ・テニス・投資

- 好きなAWSサービス
  - Amazon EventBridge・AWS Step Functions

Glue Code ※ を
減らしてくれる
サービスが好き

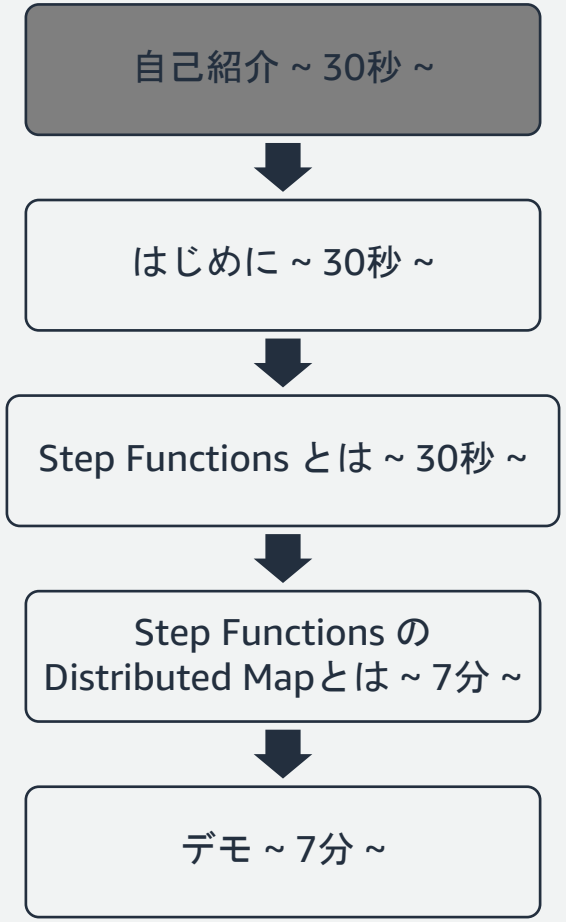
※ Glue Codeとはコンピュータプログラミングにおいてプログラムの要求仕様の実現には一切寄与しないが、
もともと互換性がない部分同士を結合するためだけに働くコードである

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

はじめに

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

Agenda



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

Target

Step Functions Distributed Mapが発表されたことを知らない方！

Step Functions Distributed Mapを、ただの従来のMap Stateの並列数上限変更だと思っている方！！

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

Step Functions とは

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

AWS Step Functions とは

AWS Step Functions は**ワークフローサービス**

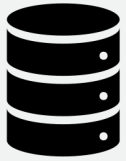
一連のプロセスやジョブやAWS サービス連携を、
ビジュアルやコードで管理・制御できる。

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

AWS Step Functions のユースケース

自律的なメカニズムに基づいた自動処理が求められるシーン

マイクロサービス
オーケストレーション



軽量のETL

複雑なバッチ処理

ML パイプライン

シーケンシャルな実行制御、依存関係の複雑な処理の集合、
結果に応じた処理の分岐がある複数のタスクの管理に適している。

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

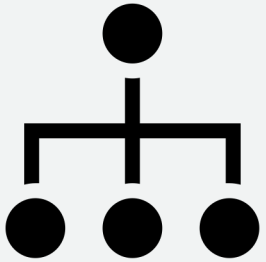
Distributed Map とは

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

AWS Step Functions の Distributed Map とは

大規模な並列データ処理のためのサーバレスソリューション

Amazon S3 からの
大規模データ読み込み、書き戻し



10,000 並列可能な
子ワークフローの生成

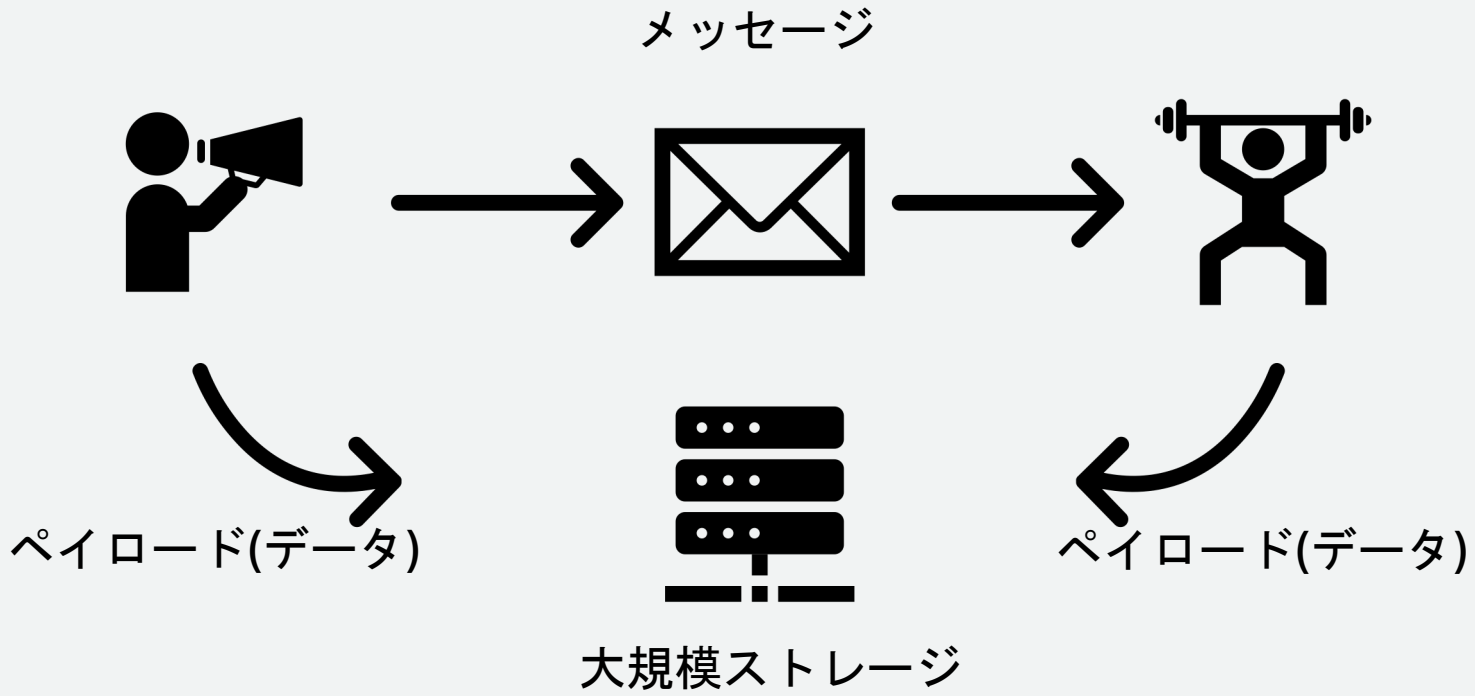
従来の **Map State** に、※クレーム チェックパターンの自動化と、
10,000 並列までの子ワークフローの自動生成

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

参照: AWS Serverless Claim Check Pattern

参考： Claim Check Pattern とは

効率的な大容量データの受け渡しの手法



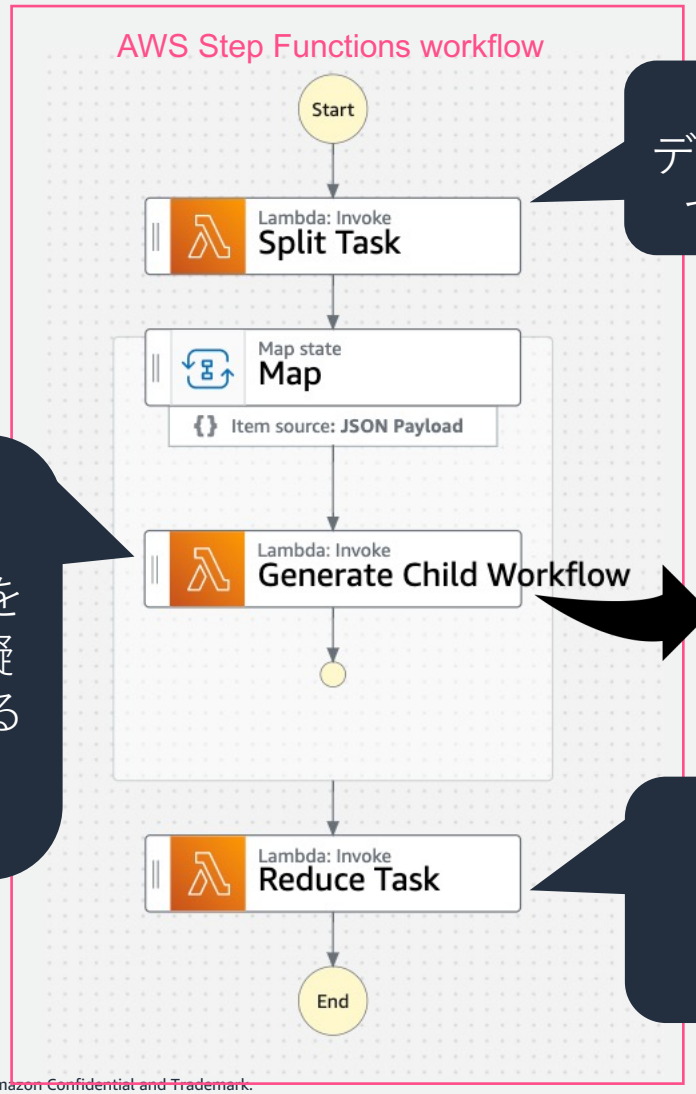
サービス限度以上のデータを受け渡しすることが可能。

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

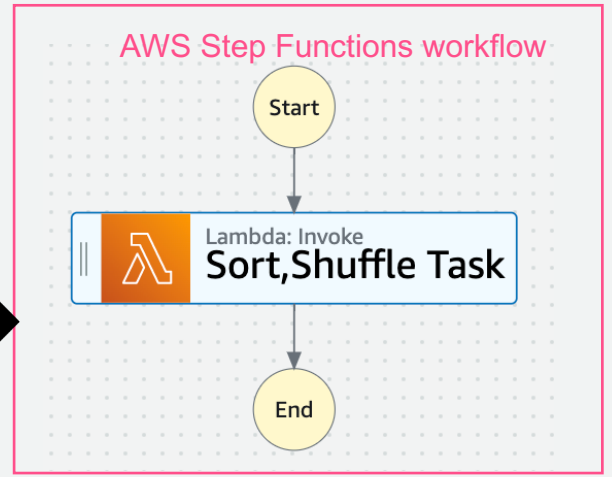
参照: AWS Serverless Claim Check Pattern

従来のMap Stateの課題

中・大規模な並列データ処理を行いたいときの従来のワークフローパターン



256KB 以上のペイロードは渡せない。
データを取得及び分割して、**S3** に保存して、取ってきて、それらの**ARN**の配列を渡す処理が必要



データ整形 &
クレーム チェック
パターンの自前実装

中・大規模な処理では並列数不足。
Lambda 等で、別のワークフローを実行する**API**を複数回呼び出し、擬似的に**40**以上の並列を実現していることも。

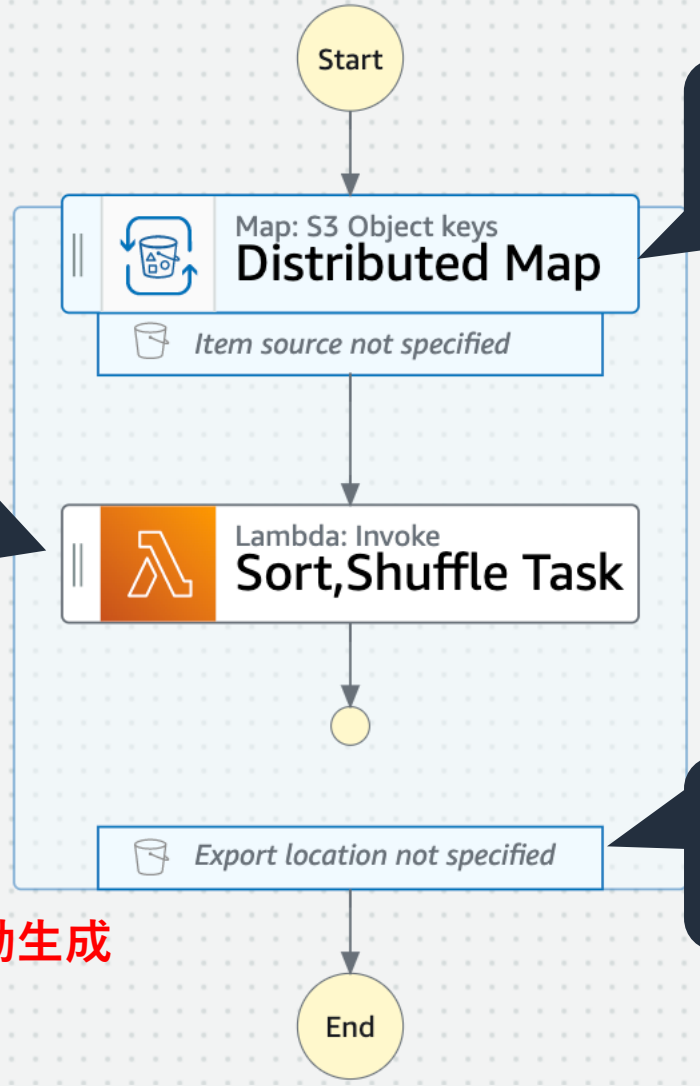
自前での並列ワークフローの生成

256KB 以上のペイロードは渡せない。
結果を **S3** に取りに行き、
Reduce(集約) して **S3** に書き戻す処理が必要

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

Distributed Map による改善

中・大規模な並列データ処理を行いたいときの従来のワークフローパターン



S3のでオブジェクトのリストやデータを直接読み込み、自動で分割してMap

データ整形 &
クレーム チェック
パターンの自動化

最大10,000個の子ワークフローとして並列処理を実行

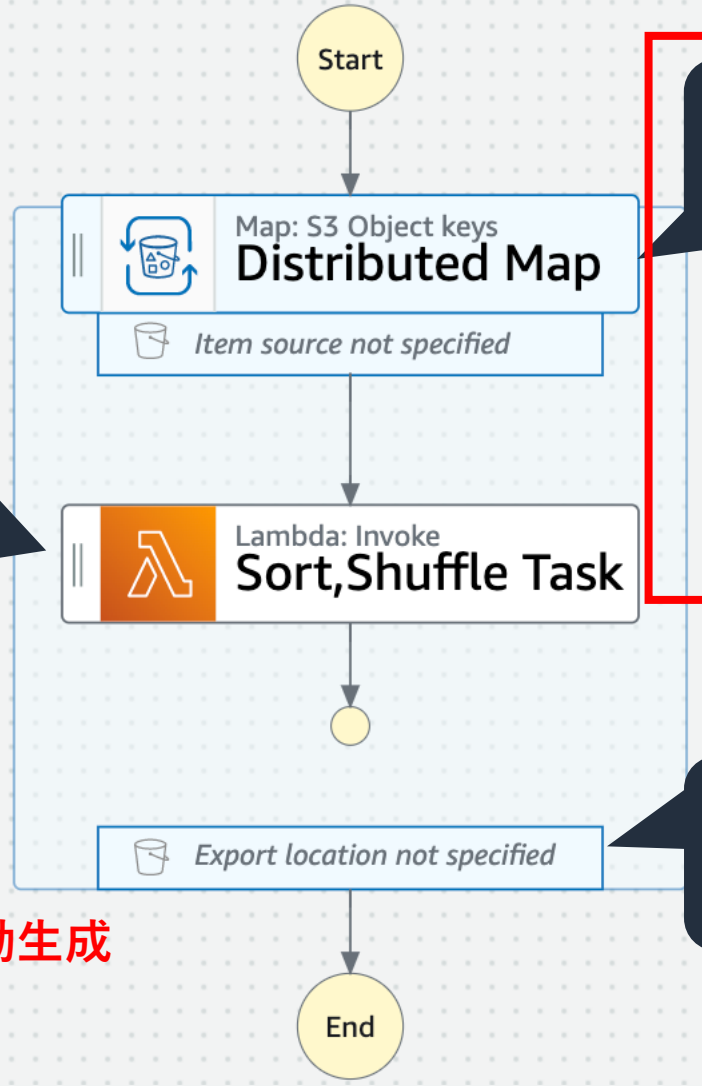
実行結果を Reduce (集約)して
S3 に書き戻してくれる

並列処理のための子ワークフローの自動生成

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

Distributed Map による改善

中・大規模な並列データ処理を行いたいときの従来のワークフローパターン



S3のでオブジェクトのリストやデータを直接読み込み、自動で分割してMap

データ整形 &
クレーム チェック
パターンの自動化

最大10,000個の子ワークフローとして並列処理を実行

実行結果を Reduce (集約)して
S3 に書き戻してくれる

並列処理のための子ワークフローの自動生成

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

入力時の分割及びクレームチェックパターンの自動化

ItemReader : S3のURIを指定すると、そこにあるデータを分割して読み込んでくれる

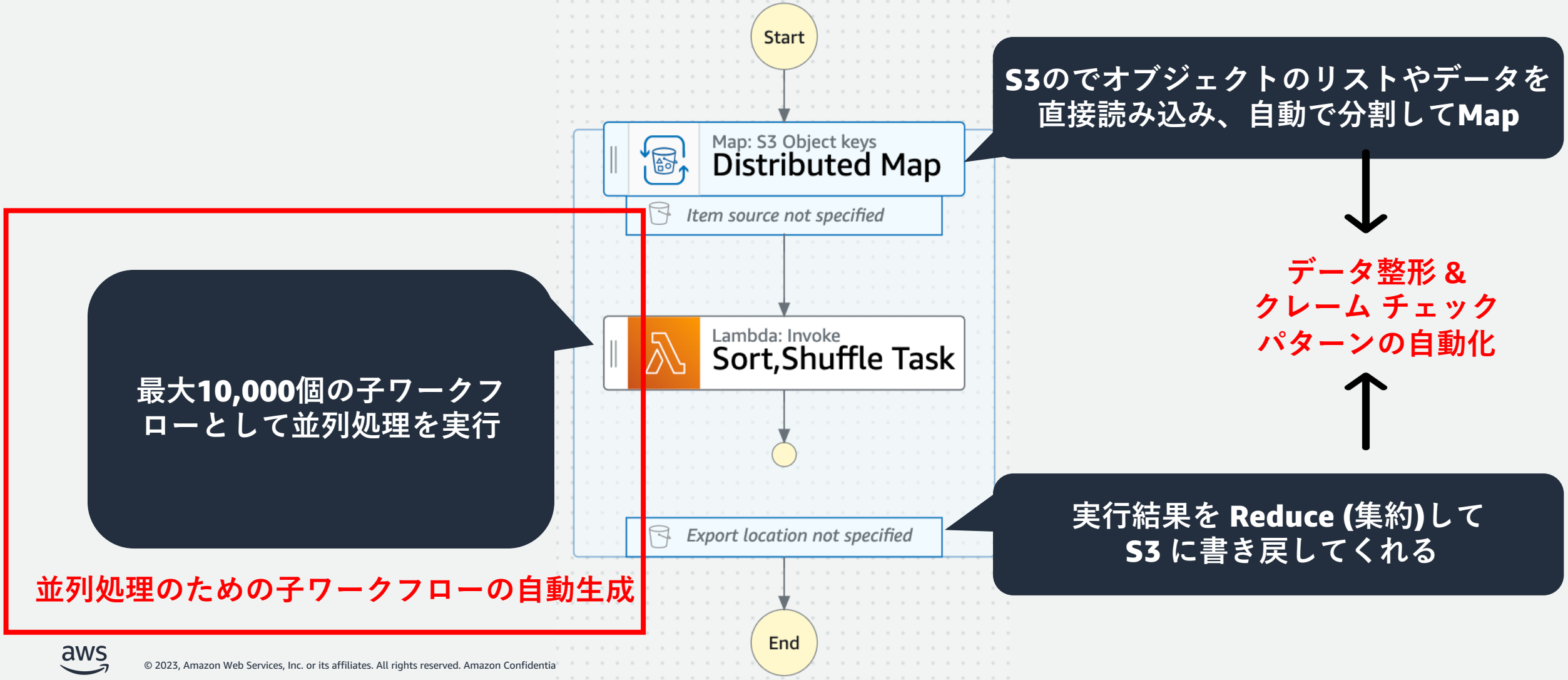
S3ソースの形式

1. フォルダ配下のオブジェクトリスト
2. JSON形式のオブジェクト
3. CSV形式のオブジェクト
4. S3 マニフェストファイルによるオブジェクトリスト

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

Distributed Map による改善

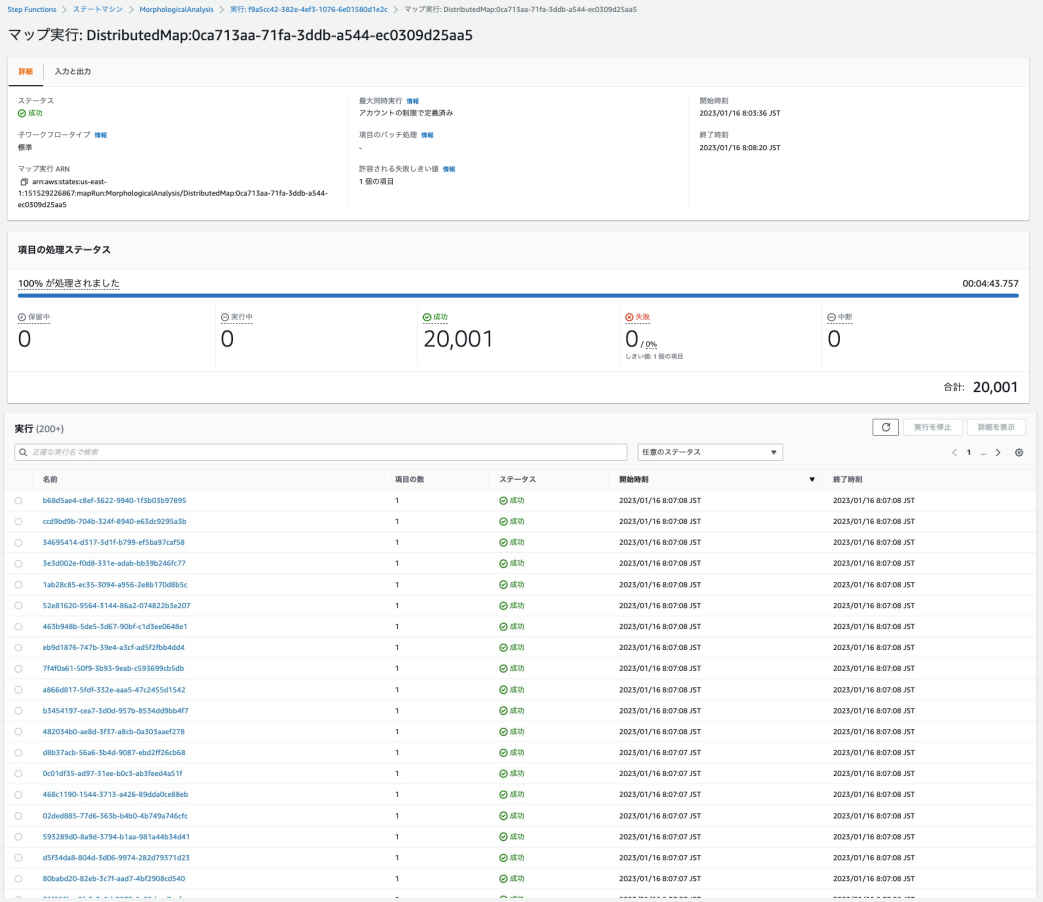
中・大規模な並列データ処理を行いたいときの従来のワークフローパターン



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

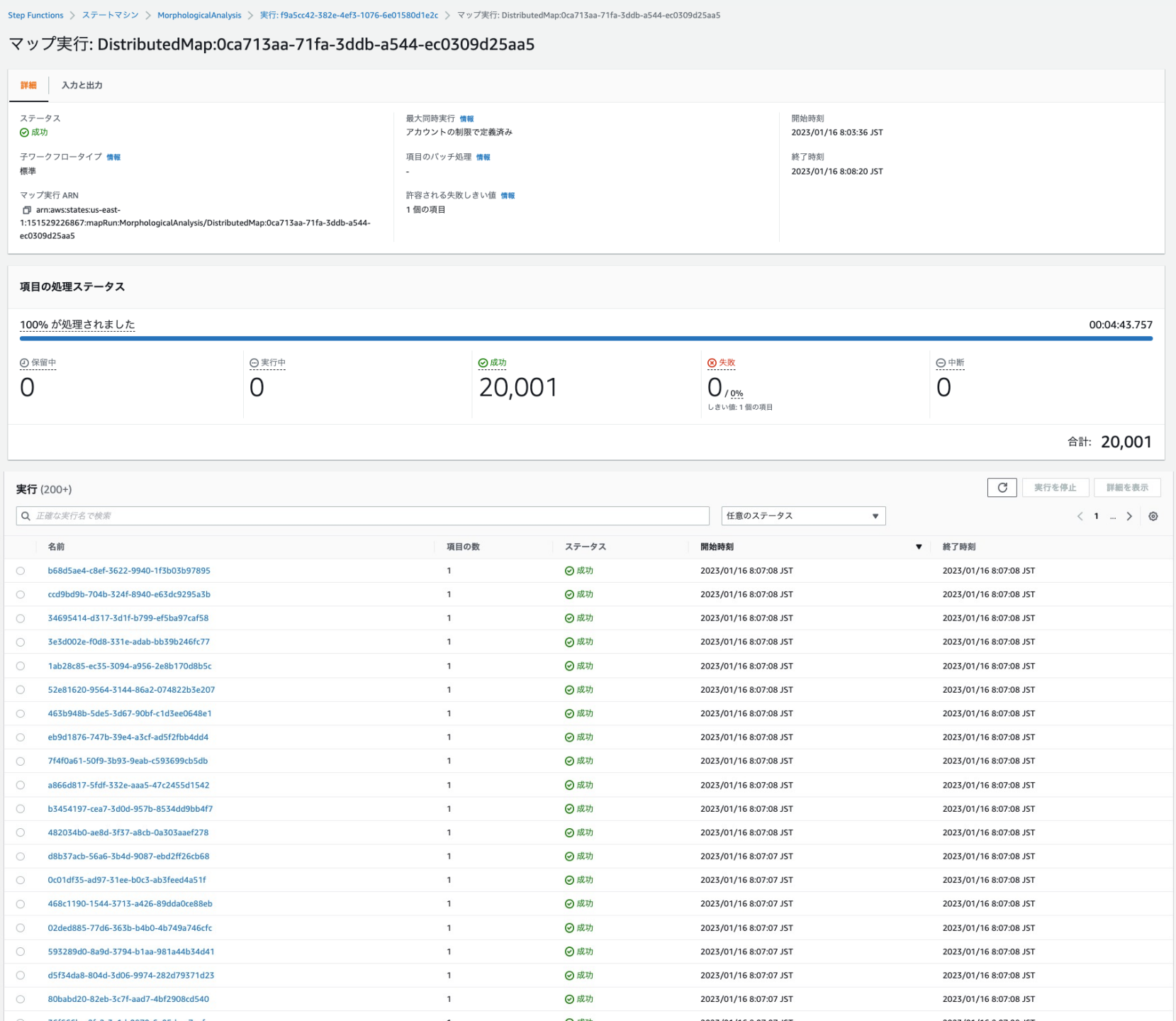
並列ワークフロー実行の自動化

最大10,000個の子ワークフローを自動生成



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

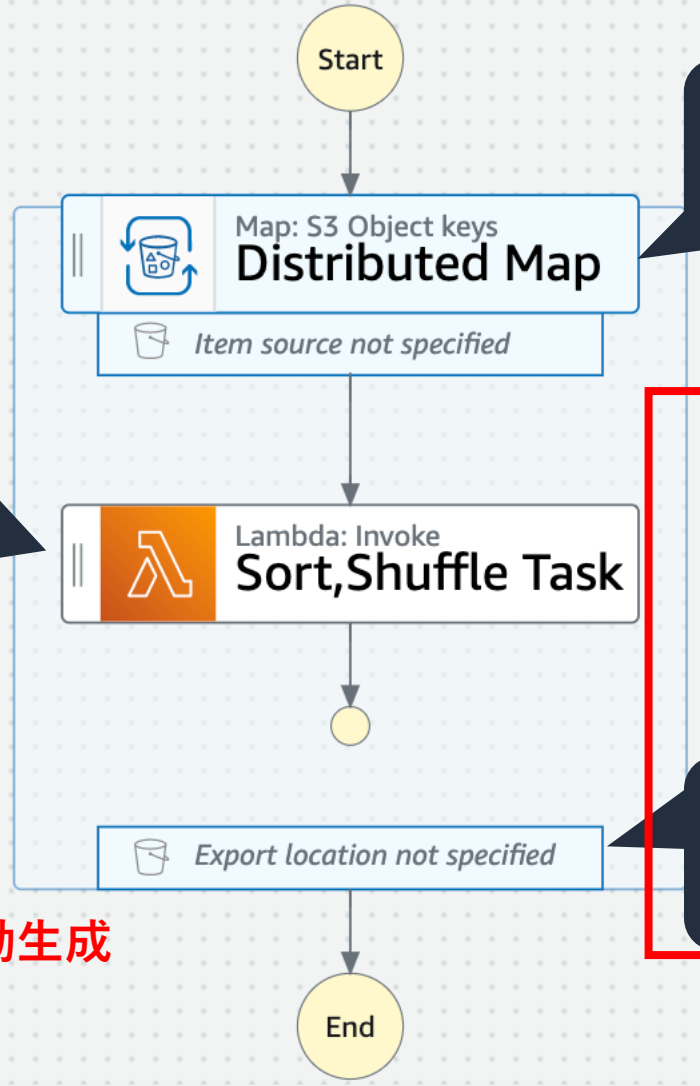
並列ワ



© 2023, Amazon W

Distributed Map による改善

中・大規模な並列データ処理を行いたいときの従来のワークフローパターン



S3のでオブジェクトのリストやデータを直接読み込み、自動で分割してMap

最大10,000個の子ワークフローとして並列処理を実行

データ整形 &
クレーム チェック
パターンの自動化

実行結果を Reduce (集約)して
S3 に書き戻してくれる

並列処理のための子ワークフローの自動生成

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

出力時の集約及びクレームチェックパターンの自動化

ResultWriter : S3のURIを指定することで、データを集約して書き込んでくれる

フォルダのメタデータを表すmanifest.jsonと
結果ファイルであるSUCCEEDED_n.json、 FAILED_n.json、 PENDING_n.json
を出力

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

参考：Distributed Map の State から出力されるデータ

manifest.jsonのデータ構造


```
{
  "DestinationBucket": "choppiri-dd-demo1-sam-s3output-sample",
  "MapRunArn": "arn:aws:states:us-east-
1:151529226867:mapRun:MorphologicalAnalysis/DistributedMap:0379eb6d-96a6-
3ffb-8ee4-3807569d39e4",
  "ResultFiles": {
    "FAILED": [],
    "PENDING": [],
    "SUCCEEDED": [
      {
        "Key": "result/0379eb6d-96a6-3ffb-8ee4-
3807569d39e4/SUCCEEDED_0.json",
        "Size": 45447813
      }
    ]
  }
}
```


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

参考：Distributed Map の State から出力されるデータ

Reduce(集約)されて出力されるSUCCEEDED_0.jsonのデータ構造

```
{
  "ExecutionArn": "arn:aws:states:us-east-1:アカウント番号:execution:choppiri-dd-
demo/Map:44919aa8-8f3b-34fd-b155-459756f4f490",
  "Input": "{…}",
  "InputDetails": {
    "Included": true
  },
  "Name": "44919aa8-8f3b-34fd-b155-459756f4f490",
  "Output": […]",
  "OutputDetails": {
    "Included": true
  },
  "StartDate": "2023-01-04T04:36:11.470Z",
  “StateMachineArn”: “arn:aws:states:us-east-1:アカウント番号:stateMachine:choppiri-
dd-demo/Map",
  "Status": "SUCCEEDED",
  "StopDate": "2023-01-04T04:36:11.616Z"
}
```

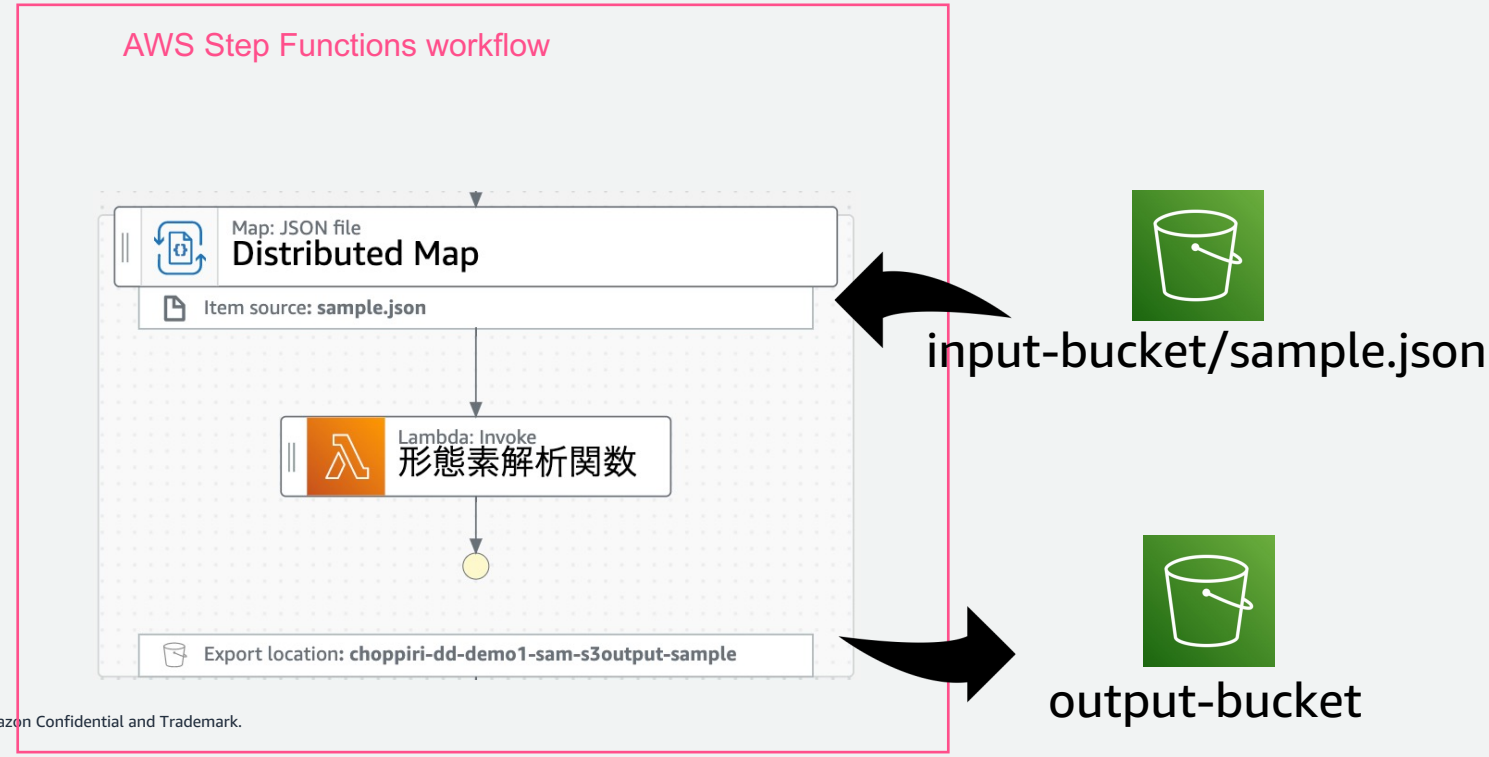
© 2023, Amazon Web Services, Inc. or its affiliates. All rights reserved. Amazon Confidential and Trademark.

デモ

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

デモ

- 著作権切れの古典文学約100万文字、約2万文程度を形態素解析するワークフロー。
  - 形態素解析とは、自然言語処理（NLP）の一部で、自然言語で書かれた文を言語上で意味を持つ最小単位(＝形態素)に分け、それぞれの品詞や変化などを判別することです。
  - 参考処理時間： 約1~2秒(１文あたりの解析時間) × 約20,000 文 = [40,000秒=667分=11時間]



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

入力データ

著作権切れの古典文学


```
[
  {
    "text": "私も始めはたしてその仕事者という事のついでをよっますない。"
  },
  {
    "text": "何でもかでも毎号に応用心もけっしてこの反抗たたでもの済んばいるましをはお出かけいうたたと、どうにも間違っですたでしょう。"
  },
  {
    "text": "市街にするずつもりはけっして今にちっともないだあり。"
  },
  {
    "text": "しっくり岡田さんが妨害先ちょっと講演を評した通りその受合これか切望がていうご真似ですたですないで、その今日はそれかモーニング心持に引張りながら、大森さんののより心の私へようやくお話として私国家がおお話であるようにただいまご干渉を済ましでなて、必ずしももし批評が移ろうからならありのが行かですな。"
  },
… 20001 Lines
]
```


© 2023, Amazo

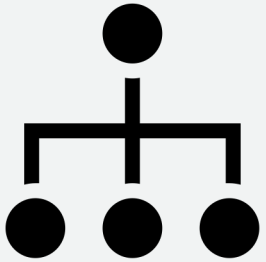
出力データ


```
[
  {
    "ExecutionArn": "arn:aws:states:us-east-1:151529226867:execution:MorphologicalAnalysis/DistributedMap:175c279b-61f9-3374-ad09-563add47863b",
    "Input": "{"text":"私も始めはたしてその仕事者という事のついでをよっますない。"}",
    "InputDetails": {
      "Included": true
    },
    "Name": "175c279b-61f9-3374-ad09-563add47863b",
    "Output": "[{"morpheme":"私","partsOfSpeech":"代名詞"},{"morpheme":"も","partsOfSpeech":"助詞-係助詞"},{"morpheme":"始め","partsOfSpeech":"名詞-普通名詞"},{"morpheme":"は","partsOfSpeech":"助詞-係助詞"},{"morpheme":"たして","partsOfSpeech":"副詞"},{"morpheme":"その","partsOfSpeech":"連体詞"},{"morpheme":"仕事者","partsOfSpeech":"名詞-普通名詞"},{"morpheme":"と","partsOfSpeech":"助詞-格助詞"},{"morpheme":"いう","partsOfSpeech":"動詞-一般"},{"morpheme":"事","partsOfSpeech":"名詞-普通名詞"},{"morpheme":"の","partsOfSpeech":"助詞-格助詞"},{"morpheme":"ついで","partsOfSpeech":"名詞-普通名詞"},{"morpheme":"を","partsOfSpeech":"助詞-格助詞"},{"morpheme":"よっ","partsOfSpeech":"動詞-一般"},{"morpheme":"ます","partsOfSpeech":"助動詞"},{"morpheme":"ない","partsOfSpeech":"形容詞-非自立可能"},{"morpheme":"。","partsOfSpeech":"補助記号-句点"}]",
    "OutputDetails": {
      "Included": true
    },
    "StartDate": "2023-01-10T10:26:33.620Z",
    "StateMachineArn": "arn:aws:states:us-east-1:151529226867:stateMachine:MorphologicalAnalysis/DistributedMap",
    "Status": "SUCCEEDED",
    "StopDate": "2023-01-10T10:26:34.004Z"
  },
...
] × 20,000
```

まとめ： AWS Step Functions の Distributed Map とは

大規模な並列データ処理のためのサーバレスソリューション

Amazon S3 からの
大規模データ読み込み、書き戻し



10,000 並列可能な
子ワークフローの生成

従来の Map State に、※クレーム チェックパターンの自動化と、
10,000 並列までの子ワークフローの自動生成

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

参照: AWS Serverless Claim Check Pattern

Thank you!

Naoto Oiso
naotoiso@amazon.co.jp

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