



【初級】 AWSで始めるNoSQL入門

Amazon DynamoDB Hands-on Lab

Tsutsumi, Hayato

Senior NoSQL Specialist Solutions Architect

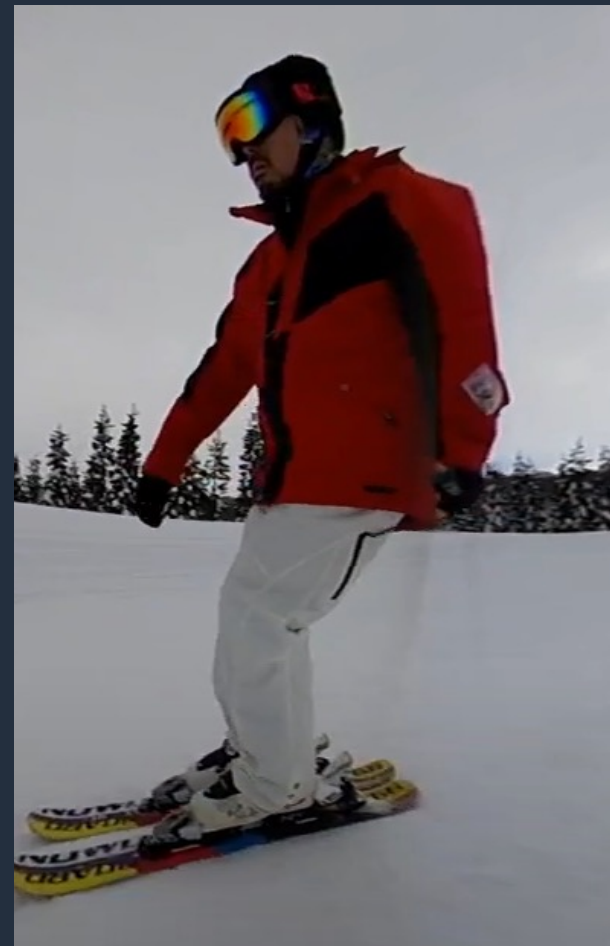
2024/02/14

自己紹介

- 堤 勇人 / Tsutsumi, Hayato
 - シニア NoSQL スペシャリスト ソリューションアーキテクト

サービス設計からフロントエンド、データベースまで一通り
Cassandraを10年ほど使っていました。

- 好きなAWSサービス
 - Amazon ElastiCache
 - Amazon MemoryDB for Redis
 - Amazon DynamoDB



本セッションのターゲット

- NoSQLの存在を知る
- NoSQLの一つであるAmazon **DynamoDB**の概要を知る
- DynamoDBの基礎的なデータアクセスについて**実際に触ってみる**

やらないこと

- 他のNoSQLデータベースの紹介
- DynamoDBのオペレーションベストプラクティス
- データモデリングの工夫
- DynamoDBのバックグラウンドアーキテクチャ

Agenda

- Amazon DynamoDBとは?
- DynamoDBのテーブル構造
- Scan, QueryとFilter
- Transaction
- Global Secondary Indexes (GSI)
- Backup

Amazon DynamoDBとは？

データの生成と活用のトレンド

データの爆発的増加



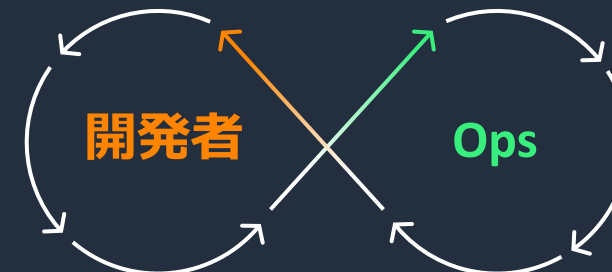
ネットワーク接続されたスマートデバイスなどの影響でデータはこの5年間で 10 倍に増加

データと分析の要件を変える マイクロサービス



マイクロサービスアーキテクチャにより、万能のデータベースの必要性が減少し、リアルタイムの監視と分析の必要性が高まる

DevOps による変化の速さ



IT から DevOps への移行により変化が加速

アプリケーションのアーキテクチャとパターンの進化

マイクロサービスは、クラウドでのアプリケーション構築を変える

Mainframe



Client Server



Three Tier



Microservices



Purpose-built databases



Purpose-built databases



DynamoDB

あらゆる規模に対応する高速で柔軟な NoSQL データベースサービス



大規模なパフォーマンス

- 1秒あたり数百万件のリクエストを処理
- 1桁のミリ秒のレイテンシーを実現
- グローバル・レプリケーション
- Amazon Kinesis データストリームを使用した新しい高度なストリーミング



サーバ管理なし

- メンテナンスフリー
- 自動スケーリング
- オンデマンドキャパシティモード
- AWS Lambda, Amazon Redshift, Amazon Opensearch Service との統合のための変更データキャプチャを提供



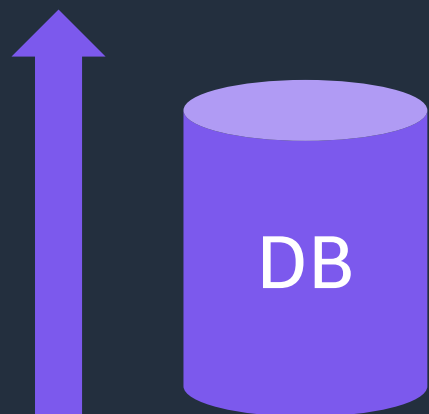
エンタープライズ対応

- ACID トランザクション
- 保管時の暗号化
- 継続的バックアップ (PITR)、およびオンデマンドのバックアップと復元
- NoSQL Workbench
- テーブルデータを S3 にエクスポートする
- PartiQL (SQL 互換のクエリ言語) のサポート

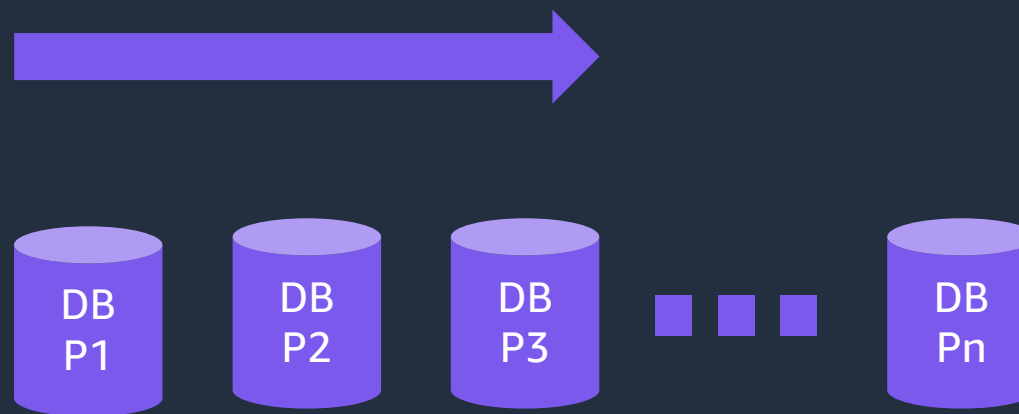
DynamoDBのテーブル構造

データベースのスケーリング

従来の SQL

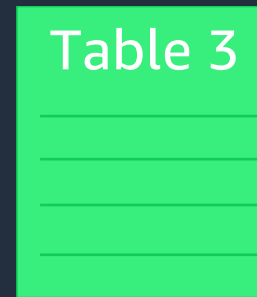
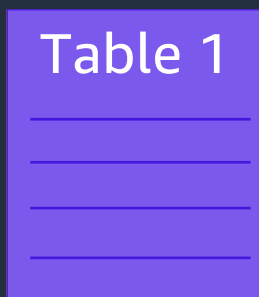


NoSQL



基本的な前提条件：水平スケールするためのデータ設計手法

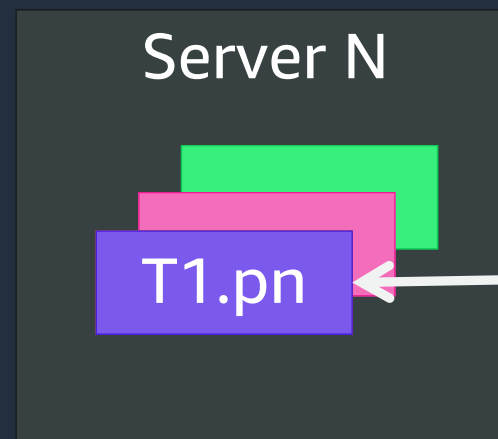
テーブルの操作をするとき...



DynamoDB が必要な処理を行います

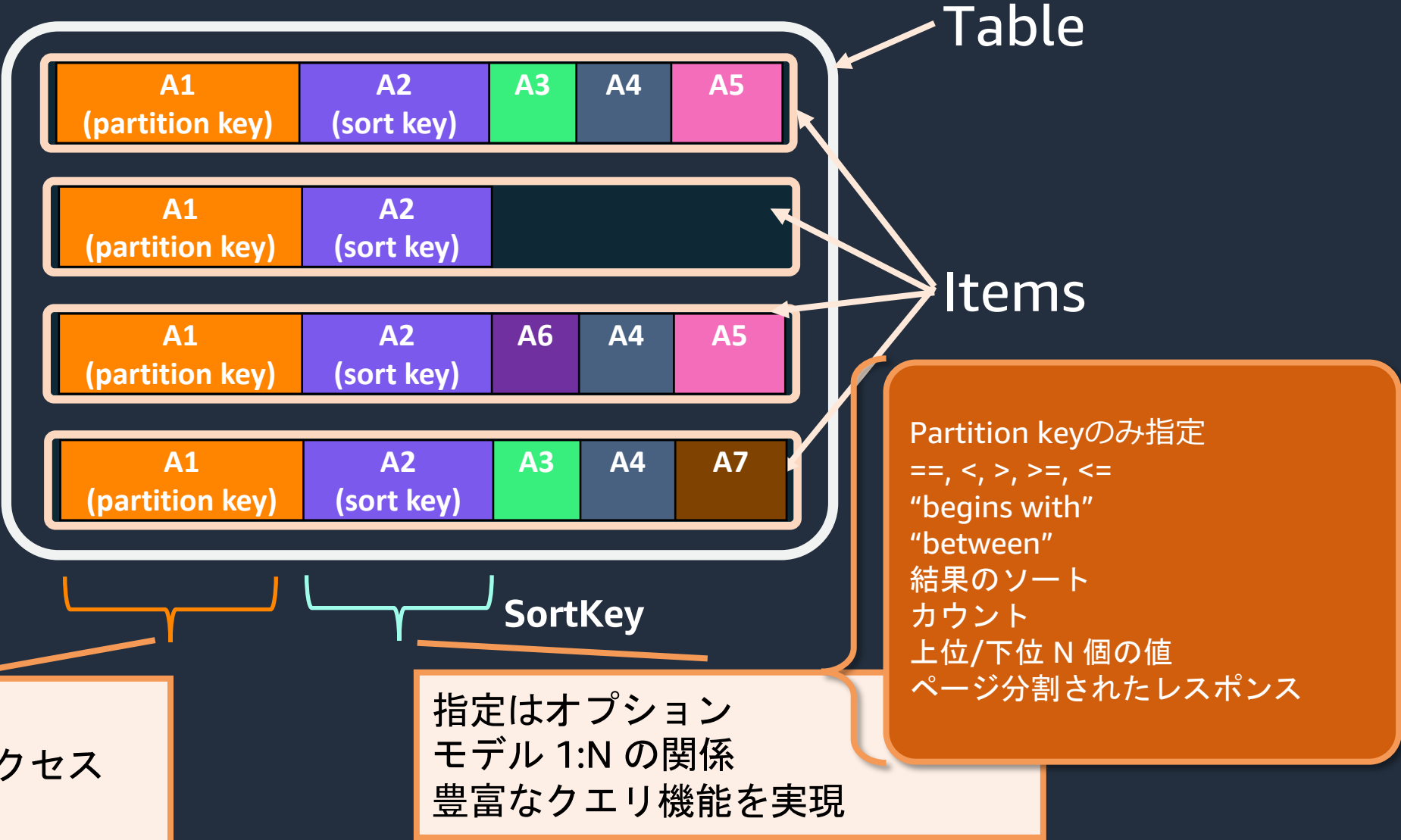


...



← 1K WCU と 3K RCU

DynamoDB テーブル



Lab:

Launch Cloud9 IDE - Read Sample Data

Scan, QueryとFilter

GetItem pk oranieかつ sk latest result

userId (Partition key) : String ⇅	sk (Sort key) : String ⇅	createdAt : String ⇅	updatedAt : String ⇅	data : String ⇅
oranie	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
oranie	bookmark_20210407_1600_0000	20210407_1600_0000	20210407_1600_0000	Title:testA,meta:iroiro
oranie	bookmark_20210407_1700_0000	20210407_1700_0000	20210407_1700_0000	Title:testB,meta:iroiro
Alice	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
Bob	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...

Scan

userId (Partition key) : String ↕	sk (Sort key) : String ↕	createdAt : String ↕	updatedAt : String ↕	data : String ↕
oranie	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
oranie	bookmark_20210407_1600_0000	20210407_1600_0000	20210407_1600_0000	Title:testA,meta:iroiro
oranie	bookmark_20210407_1700_0000	20210407_1700_0000	20210407_1700_0000	Title:testB,meta:iroiro
Alice	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
Bob	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...

ScanかつFilterでdataがTitleから始まるもの

userId (Partition key) : String	sk (Sort key) : String	createdAt : String	updatedAt : String	data : String
Filterによるcutをした部分				
oranie	bookmark_20210407_1600_0000	20210407_1600_0000	20210407_1600_0000	Title:testA,meta:iroiro
oranie	bookmark_20210407_1700_0000	20210407_1700_0000	20210407_1700_0000	Title:testB,meta:iroiro
Filterによるcutをした部分				

Query – pk oranieかつbookmarkから始まるsk

userId (Partition key) : String ↕	sk (Sort key) : String ↕	createdAt : String ↕	updatedAt : String ↕	data : String ↕
oranie	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
oranie	bookmark_20210407_1600_0000	20210407_1600_0000	20210407_1600_0000	Title:testA,meta:iroiro
oranie	bookmark_20210407_1700_0000	20210407_1700_0000	20210407_1700_0000	Title:testB,meta:iroiro
Alice	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...
Bob	latest_result	20210407_1508_01	20210407_1508_01	QueryResultData_ir...

Lab:

Reading Item Collections using Query

- Deleting Data

Transaction

DynamoDB トランザクション API

- TransactWriteItems

同期的な更新、挿入、削除、およびチェック

Atomicityの担保

トランザクション内で最大 100 個の Item

複数のテーブル、複数 Item をサポート

複雑な条件チェック

- ベストプラクティス

Item 間で変更をコミットする

条件付きバッチ挿入 / 更新

- アンチパターン

- 正規化されたデータの保守

ゲームの状態 - トランザクション API

ゲーマー「Hammer57's」ヘルスケアコインのATOMIC更新

```
{ "TransactItems" : [ {  
  "Update " : {  
    "TableName": "Gamers",  
    "Key" : {"GamerTag" : {"S": "Hammer57"},  
    "Type" : {"S" : "Status"}}.  
    "UpdateExpression" : "Set health = :nhealth",  
    "ExpressionAttributeValues":{":nhealth":{"N":"100"}}  
  },  
  {  
    "Update " : {  
      "TableName": "Gamers",  
      "Key" : {"GamerTag" : {"S": "Hammer57"},  
      "Type" : {"S" : "Assets"} }.  
      "ConditionExpression" : "coins > :cost",  
      "UpdateExpression" : "Set coins = coins - :cost",  
      "ExpressionAttributeValues" : {":cost":{"N":"400"}}  
    }  
  }  
}
```

Primary key		Attributes		
Partition key: GamerID	Sort key: Type			
Hammer57	Assets	Coins		
		1000		
	Rank	Level	Points	Tier
		87	4050	Elite
	Status	Health	Progress	
		90%	30	
	Weapon	Class	Damage	Range
		Taser	55-67	120

Lab:

Transaction

- Global Secondary Indexes

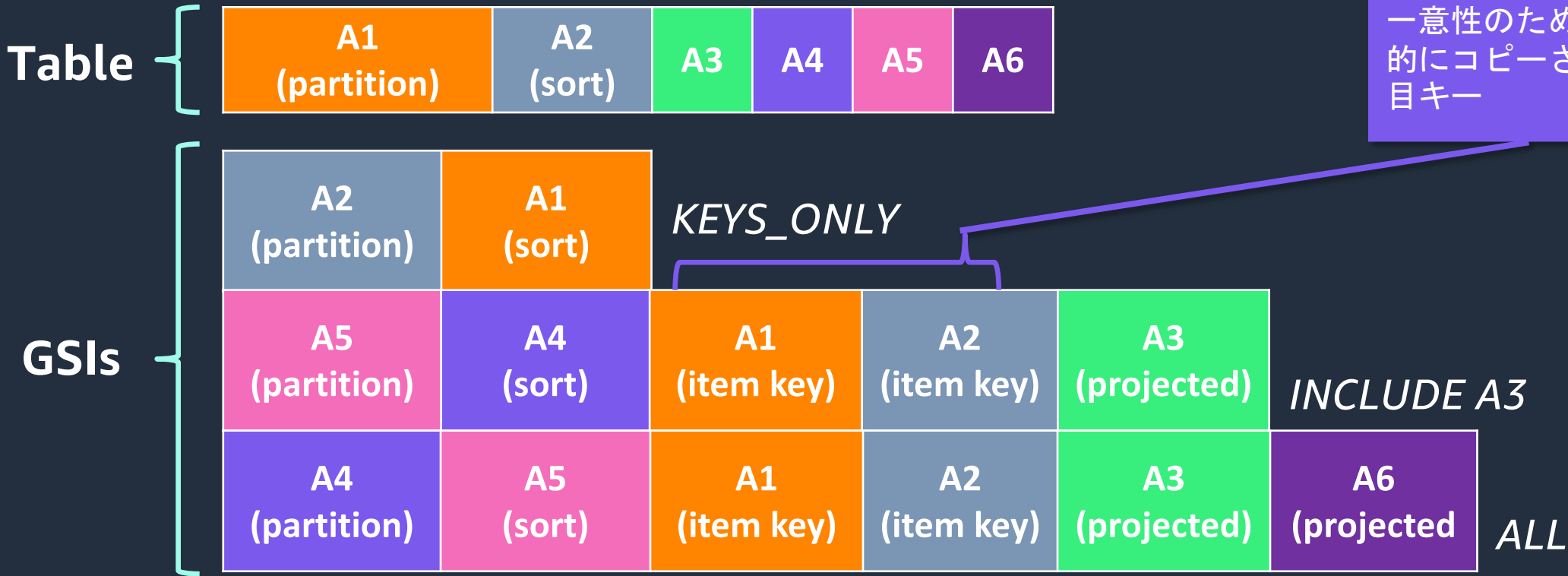
Global Secondary Indexes (GSI)

Global secondary index (GSI)

テーブルあたり **最大 20 GSI**

異なるパーティションまたはソートキーを利用できる
インデックスはすべてのパーティションキーにまたがる
複合インデックスに複合ソートキーを使用する

GSI 用に個別にプロビジョニングされた RCUs/WCUs



新しいアクセスパターンに対応する

ベーステーブル

Primary key		Attributes	
Partition key: DeviceID	Sort key: State#Date	Operator	Date
d#12345	NORMAL#2020-04-24T14:55:00	Liz	2020-04-24
	WARNING1#2020-04-24T14:45:00	Liz	2020-04-24
	WARNING1#2020-04-24T14:50:00	Liz	2020-04-24
d#54321	NORMAL#2020-04-11T06:00:00	Liz	2020-04-11
	NORMAL#2020-04-11T09:30:00	Sue	2020-04-11
	WARNING2#2020-04-11T09:25:00	Sue	2020-04-11
	WARNING3#2020-04-11T05:55:00	Liz	2020-04-11
d#11223	WARNING4#2020-04-27T16:10:00	Sue	2020-04-27
	WARNING4#2020-04-27T16:15:00	Sue	2020-04-27

Primary key		Attributes	
Partition key: Operator	Sort key: Date	State#Date	DeviceID
2020-04-11	2020-04-11	WARNING3#2020-04-11T05:55:00	d#54321
		NORMAL#2020-04-11T06:00:00	d#54321
2020-04-24	2020-04-24	WARNING1#2020-04-24T14:45:00	d#12345
		WARNING1#2020-04-24T14:50:00	d#12345
2020-04-24	2020-04-24	NORMAL#2020-04-24T14:55:00	d#12345
		WARNING2#2020-04-11T09:25:00	d#54321
2020-04-11	2020-04-11	NORMAL#2020-04-11T09:30:00	d#54321
		WARNING4#2020-04-27T16:10:00	d#11223
2020-04-27	2020-04-27	WARNING4#2020-04-27T16:15:00	d#11223

GSI-Operator



Backup

データ保持のためのバックアップの使用



Create backup

Backup settings [Info](#)

Source table

Messages

Backup name

This will be used to identify your backup.

my-backup

From 3 to 255 characters in length, only A-Z,a-z, 0-9, underscores, hyphens and periods allowed.

Cancel

Create backup

On-demand backupsは長期間、コンプライアンス要件を満たすような保持が可能

point-in-time recovery (PITR)過去 35 日間の任意の時点にテーブルを復元

パフォーマンスに影響を与えることなく、PB単位のデータをバックアップ

異なるAWSリージョンへのリストアも可能

Lab: Backup

まとめ

まとめ

- Amazon DynamoDB はスケーラビリティの高いNoSQLデータベース
 - オペレーションが楽
- テーブルにPartition KeyとSort Keyを持つ
 - Sort Keyの前方一致を活用してQuery
- Transaction や GSI 等、高度な機能も持つ



Thank you!