DynamoDB

amazon web services™

قواعد بيانات DynamoDb
على بيئة خدمات أمازون ويب AWS

النسخة 1.0 - 6/2018

تتضمن هذا الورقات مجموعة من المواضيع التي تتعلق بقواعد بيانات DynamoDb على بيئة خدمات أمازون ويب AWS التي تساعد على فهم هذه الخدمة ومعرفة ما يمكن عمله وتنفيذه بأفضل الطرق.

اسماعيل عنجريني - فادي عبدالوهاب

الفهرس

مقدمة 3
إسماعيل عنجريني 4
فادي عبد الوهاب 4
جداول DynamoDB 5
أنماط القراءة في DynamoDB 6
Read Capacity Unit 6
Write Capacity Units 7
مفتاح الجدول المركب 7
القراءة من DynamoDB 9
القراءة من DynamoDB باستخدام query 9
فلترة نتائج الاستعلام 9
تحديد عدد نتائج الاستعلام 10
تقسيم النتائج إلى صفحات وعدد النتائج 11
استخدام الفهارس الثانوية مع جداول DynamoDB 11
الفهرس الثانوي العام GSI 11
مزامنة البيانات بين الجدول والفهرس العام 13
الفهرس الثانوي المحلي LSI 13
استخدام لغات البرمجة مع DynamoDB 15
تجهيز بيئة العمل لاستخدام DynamoDB SDK 16
البرمجة باستخدام Low-Level Interface 21
البرمجة باستخدام Document 21
البرمجة باستخدام Object Persistence 21
مثال عن البرمجة باستخدام Object Persistence 21
قراءة سطر من الجدول 24
قراءة أكثر من سطر من الجدول 24
تغيير اسم العمود 25
حذف سطر من جدول 25
النسخ الاحتياطي 27
عمل نسخة احتياطية كاملة لقاعدة بيانات DynamoDb 29
استرجاع نسخ احتياطية كاملة لقاعدة بيانات DynamoDb 40
حذف النسخ الإحتياطية لقاعدة بيانات DynamoDb 41
عمل نسخة احتياطية بشكل تزايدي لقاعدة بيانات DynamoDb 43

إدراة أمن المعلومات في قاعدة بيانات DynamoDb 46
إدارة الصلاحيات والوصول للبيانات الموجودة في قاعدة بيانات DynamoDb 46
التحكم بالوصول على مستوى الجدول Table ومستوى واجهات التطبيق البرمجية API 46
التحكم بالوصول على حسب الحقول Attribute 50
التحكم بالوصول على حسب المفتاح الرئيسي Partition Key 51
تشفير البيانات المخزنة في قاعدة بيانات DynamoDb 54

مقدمة

مع التقدم السريع والكبير في الخدمات السحابية ولعل من أشهر هذه المنصات اعتمادا على تقارير Gartner هي خدمات أمازون AWS ونظرًا للحاجة إلى استخدام قوعد بيانات من النوع NoSQL ما تقدمه من ميزات جديدة وخاصة الحجم الكبير مع السرعة الفائقة, قررنا أن نوفر دليلًا سريعًا عن كيفية استخدام DynamoDb على خدمات أمازون وذلك بشكل مختصر, علمًا أن هذا هو الإصدار الأول وسنعمل جاهدين على تحسينه وإضافة ما يحتاجه المبرمج العربي.

شعارنا هو #بالعربي_نتقدم

إسماعيل عنجريني

مساعد مدير التطوير في شركة ثقة, مختص Big Data و AWS Cloud ومهتم بأمن المعلومات. أحب الكتابة ومشاركة المعرفة وخاصة في مجال تقنية المعلومات.

للمزيد عن المعلومات الخاصة بي وبعض أعمالي وكتاباتي المنشورة على الانترنت يمكن الاطلاع على الرابط التالي:

https://www.linkedin.com/in/ienjreny/

فادي عبد الوهاب

مهتم بالتقنية وبالأخص المواضيع التي تتعلق بأمن المعلومات للتطبيقات والخدمات السحابية وانشاء بيئة قابلة للتمدد والتعافي من المخاطر حيث حاصل على العديد من الشهادات المعتمدة في هذه المجالات.

AWS Solutions Architect – Professional

(ISC) 2 - CSSLP® Certified Secure Software Lifecycle Professional

للمزيد عن المعلومات الخاصة بي وبعض أعمالي وكتاباتي المنشورة على الانترنت يمكن الاطلاع على الروابط التالية:

مدونتي:

https://fabdulwahab.com

حسابي على تويتر:

https://twitter.com/fadi_abdulwahab

حسابي على لينكدإن:

https://www.linkedin.com/in/fadiabdulwahab

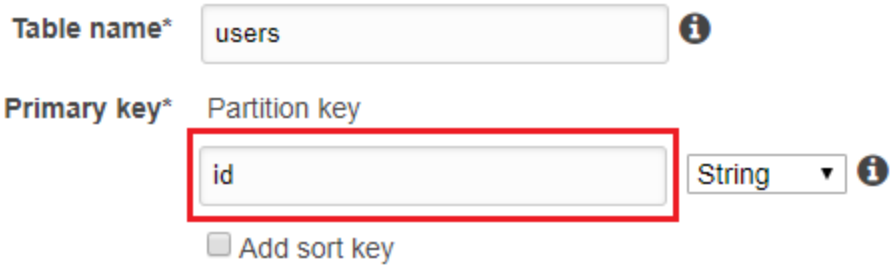
جداول DynamoDB

DynamoDB هي عبارة عن قاعدة بيانات من نوع NoSQL تم تصميمها من قبل شركة أمازون لكي توفر الأداء العالي وإمكانية التوسع بسهولة. تستخدم أمازون DynamoDB لتوفير خدمات مختلفة على موقع amazon.com

العنصر الرئيسي عند استخدام DynamoDB هو الجدول, حيث لا يتوفر خيار إنشاء أكثر من قاعدة بيانات كما يحدث عند استخدام MS SQL Server. فمثلًا في حال وجود أكثر من مشروع تستخدم فيها DynamoDB كقاعدة بيانات على aws فيجب عندها الاعتماد على الجداول فقط حيث لا يمكن إنشاء قاعدة بيانات لكل مشروع إلا في حال إنشاء AWS Account لكل مشروع وهذا الخيار لن يكون مناسبًا في كثير من الحالات.

لنفترض لدينا مشروعين, الأول للرسائل النصية القصيرة والثاني للمستشفيات, عندها قد يكون شكل الجداول كما يلي:

وباعتبار DynamoDB هي من قاعدة بيانات من النوع NoSQL فلا توجد أعمدة معرفة بشكل صريح لكل جدول حيث أنها إحدى أهم الميزات التي يجب توفرها في قواعد البيانات من نوع NoSQL وهي Schema less. لكن عند إنشاء جدول جديد يجب تحديد مفتاح رئيسي للجدول وهو العمود الوحيد الذي يتم تحديده للجدول عند إنشائه.



وباعتبار أن العمود هو مفتاح رئيسي فيجب ألا يتكرر ويفيد بالطبع فيما يخص سرعة الأداء بشكل رئيسي إضافة إلى استخدامه في ميزات أخرى مثل Global Secondary Index.

السؤال هنا هو: كيف أضيف أعمدة إلى الجدول؟ والجواب يكمن في أن الـ Schema less تعني بأنه لا يمكن تعريف أعمدة للجدول, حيث يمكن أن يحتوي كل سطر من أسطر الجدول على أعمدة مختلفة وذلك باستخدام صيغة JSON لإدخال البيانات ضمن جداول DynamoDB. لنفترض تم تعريف الجدول sms_users مع العمود id كمفتاح رئيسي, الآن سندخل سطر يحتوي على العمود username والعمود mobile وذلك باستخدام الصيغة التالية:

```
{
  "id": "1",
  "username": "ismail",
  "mobile": "055xxxxxxx"
}
```

عند إدخال السطر السابق ضمن الجدول sms_users سيظهر عمودان إضافيان هما username و mobile ضمن الجدول كما يلي:

id	mobile	username
1	055xxxxxxx	ismail

أما إذا أردنا إضافة البريد الإلكتروني للمستخدم إضافة إلى الأعمدة السابقة فتكون الصيغة كما يلي:

```
{
  "id": "1",
  "mobile": "055xxxxxxx",
  "username": "ismail",
  "email": "ismaill.xxxxx@gmail.com"
}
```

عند إدخال السطر السابق ضمن الجدول sms_users سيظهر عمودان إضافيان هما username و mobile ضمن الجدول كما يلي:

id	mobile	username	email
2	055xxxxxxx	ismail	ismaill.xxxxx@gmail.com
1	055xxxxxxx	ismail	

نلاحظ ظهور عمود جديد email للسطر الجديد دون أي تأثير على على السطر الذي تمت إضافة أولًا.

وبما أن الجداول هي العنصر الرئيسي عند استخدام DynamoDB لذلك هناك الكثير من الخصائص والعمليات المرتبطة والتي يمكن تنفيذها على الجداول, وفيما يلي سيتم شرح أهم الخصائص التي يجب الانتباه إليها قبل البدء باستخدام أي جدول.

أنماط القراءة في DynamoDB

Eventually Consistent Reads: والتي تعني بأن البيانات التي ستعود عند تنفيذ عملية القراءة من الجدول قد لا تكون النسخة الأخيرة من البيانات التي تم تحديثها على الجدول.

Strongly Consistent Reads: والتي تعني بأن البيانات التي ستعود عند تنفيذ عملية القراءة من الجدول هي آخر نسخة تم تحديثها على الجدول.

Read Capacity Unit

كل Read Capacity Unit تعني قراءة نتيجة واحدة بالنمط Strongly Consistent Read بالثانية, أو نتيجتين بالنمط Eventually Consistent Reads, علمًا أن حجم كل نتيجة يجب ألا يتجاوز 4KB.

وأي نتيجة يتجاوز حجمها فوق 4KB تستهلك 2 Read Capacity Units, مثلًا لو كان 10 = Read Capacity Units أي 10 نتائج حجم كل واحد منها هو 4KB في الثانية. فعند قراءة نتيجة حجمها 8KB عندها يتم احتساب 2 Read Capacity Units ويتبقى فقط 8 Read Capacity Units.

ملاحظة: المثال السابق يفترح أن نمط القراءة هو Strongly Consistent Read.

عندما يتجاوز الحد المسموح قراءته في الثانية عندها يتم إعادة استثناء ProvisionedThroughputExceededException دليل على فشل عملية القراءة عندها يجب المحاولة مرة أخرى لكي تنجح عملية القراءة, علمًا يوصى باستخدام AWS SDK لتنفيذ عمليات القراءة من DynamoDB لأنها مصممة لمعالجة مثل هذه الحالات في حال حدوثها.

Write Capacity Units

كل WCU تمثل كتابة واحدة في الثانية لعنصر لا يتجاوز حجمه 1KB, فمثلًا لو أن 10 = Write Capacity Units عندها يمكن كتابة 10 عناصر خلال ثانية واحدة في الجدول, وعندما يتجاوز حجم العنصر 1KB عندها يتم احتساب 2 WCU لتنفيذ عملية كتابة العنصر بدلًا من 1 WCU.

يجب ضبط إعدادات القيم RCU و WCU بشكل مدروس لكل جدول وذلك حسب حجم البيانات المتوقعة لكل عملية قراءة وكتابة إضافة إلى عدد العمليات المتوقعة خلال الثانية الواحدة, وهذا يتم تحديده حسب كل نظام وحسب متطلبات الأعمال.

مفتاح الجدول المركب

ذكرنا سابقًا بأنه عند إنشاء جدول يجب إضافة مفتاح رئيسي كمتطلب إجباري لاستكمال عملية إنشاء الجدول وهذا ما يطلق عليه Partition key وفي هذه الحالة لا يمكن تكرار قيمة هذا المفتاح ضمن عنصرين, ويوجد نوع آخر من المفاتيح الرئيسية والذي يتألف من مفتاح رئيسي ومفتاح للترتيب Sort key وفي هذه الحالة يمكن تكرار قيمة المفتاح الرئيسي لأكثر من سطر من أسطر الجدول لكن لا يمكن تكرار قيمة المفتاح الرئيسي ومفتاح الترتيب, كما في المثال التالي حيث جدول المؤلفين يحتوي على ما يلي:

- اسم الكاتب
- عنوان الكتاب
- اللغة

حيث نلاحظ تكرار اسم الكاتب وتكرار اسم الكتاب, لكن لا يوجد تكرار لاسم الكاتب واسم الكتاب لسطرين مختلفين.

	author	title	lang
	a2	t2	en
	a2	t2_1	ar
	a2	t2_2	en
	a2	t2_3	en
	a1	t1	ar
	a1	t2	ar
	a1	t3	ar
	a1	t4	en

ويطلق على المفتاح الرئيسي اسم آخر هو Hash key والسبب هو تقني بحت, حيث أن DynamoDB توزع بيانات الجدول على أقسام منفصلة وذلك لتحسين الأداء عند طلب معلومات من أي جدول, وتتم عملية التوزيع على الأقسام بناء على قيمة الـ Hash للمفتاح الرئيسي حيث يتم تخزين جميع أسطر الجدول التي لها نفس قيمة الـ Hash للمفتاح الرئيسي في نفس القسم.

أما مفتاح الترتيب فيستخدم لإعادة الأسطر التي لها نفس المفتاح الرئيسي مرتبة حسب مفتاح الترتيب. حيث تلاحظ أن الأسطر في الصورة السابقة مجمعة حسب قيمة author ومرتبة حسب قيمة title.

فمثلًا, حسب القيم في الصورة السابقة فإن جميع الأسطر التي قيمة author تساوي a1 تكون مخزنة في نفس القسم والأسطر التي لها القيمة a2 تكون مخزنة في قسم آخر.

ملاحظة: لا يمكن تعديل قيمة المفتاح الرئيسي أو مفتاح الترتيب

القراءة من DynamoDB

يمكن القراءة من جداول DynamoDB بطريقتين هما الاستعلامات Queries والمسح Scan ولكل منهما الاستخدامات الخاصة بها, وسيتم شرحها كما يلي:

القراءة من DynamoDB باستخدام query

عند القراءة من DynamoDB باستخدام الاستعلامات Queries فالهدف هو الحصول على نتائج باستخدام المفتاح الرئيسي Primary Key أو كما يسمى أيضًا Partition Key, علمًا يجب استخدام العملية "يساوي =" للبحث باستخدام المفتاح الرئيسي, ويمكن استخدام مفتاح الترتيب كقيمة اختيارية لتنفيذ عملية الاستعلام, علمًا أن توفير قيمة للمفتاح الرئيسي إجباري عند استخدام الاستعلام.

وعند استخدام هذه العملية للقراءة من DynamoDB فإن النتائج دائمًا تكون عبارة عن مجموعة وفي حال عدم وجود نتائج تكون المجموعة فارغة.

وكما في معظم خدمات أمازون السحابية aws هناك قيد على حجم البيانات التي يمكن إعادتها باستخدام عملية استعلام واحدة فقط, وهذا الحجم هو 1 MB ولا يمكن تجاوز هذا القيد, ومن الواضح أن هذا القيد هدفه تقديم خدمة بأفضل أداء حتى في حال أن المطور كان لديه مشكلة في الشيفرة البرمجية فإن ذلك سيخفف من أثر الشيفرة البرمجية الغير احترافية.

فلترة نتائج الاستعلام

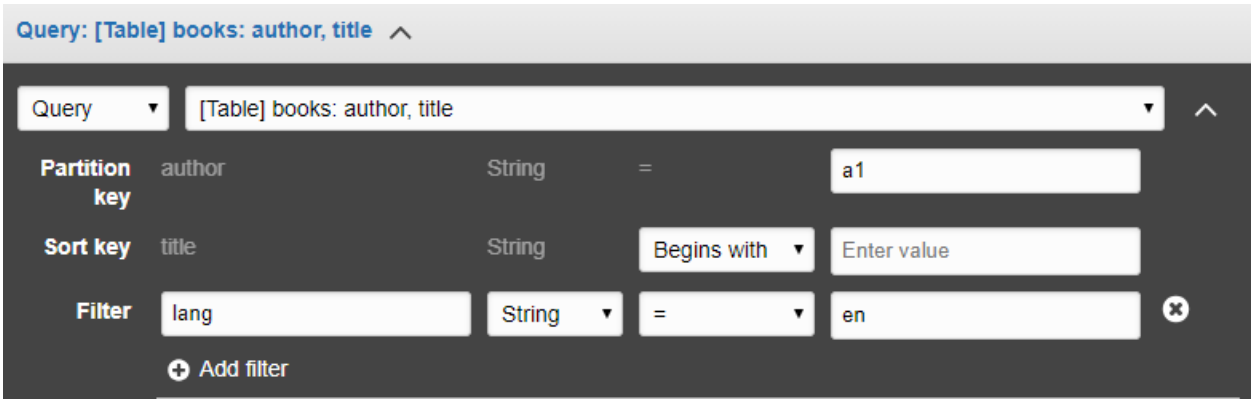
عند الحصول على نتائج من أحد جداول DynamoDB باستخدام الاستعلام يمكن تطبيق فلاتر إضافية بحيث يتم تضييق نتائج البحث التي يتم إرجاعها, **ويجب الانتباه** إلى أن الفلاتر تطبق على مجموعة نتائج الاستعلام وهذه نقطة مهمة جدًا يجب الانتباه إليها عند اعتماد الاستعلام كطريقة لاستخراج النتائج من أي جدول, فمثلًا لو أن الاستعلام بالمفتاح الرئيسي أعاد مجموعة نتائج تحتوي على 20 عنصر, فإن الفلاتر ستطبق على الـ 20 عنصر.

بنفس الطريقة فإن الفلاتر تطبق بعد تطبيق قيد حجم البيانات 1MB, أي يتم تقييد حجم البيانات بـ 1MB وبعدها تطبق الفلاتر على هذه النتائج.

والمقصود بالفلاتر الإضافية هي استخدام أحد الـ Attributes ضمن الجدول غير المفتاح الرئيسي ومفتاح الترتيب لكي يتم تضييق نتائج الاستعلام, وفيما يلي مثال عن تطبيق الفلاتر, حيث يتم الاستعلام باستخدام المفتاح الرئيسي author باستخدام القيمة a1 وعرض النتائج التي قيمة lang تساوي en فقط:

author	title	lang
a1	t4	en

ملاحظة: عدد وحدات CRU المستهلكة لعرض نتائج الاستعلام السابق مع استخدام الفلتر باستخدام القيمة lang = en تساوي 4 وحدات علمًا أن عدد الأسطر التي تمت إعادتها هي سطر واحد فقط, والسبب كما ذكرنا سابقًا هو أن الفلاتر تطبق على مجموعة النتائج فقط وليس على الجدول مباشرة, حيث أن نتيجة الاستعلام هي 4 أسطر باستخدام المفتاح الرئيسي author وبالتالي 4 قراءات وبعدها يطبق الفلتر lang=en على الأربعة أسطر.



نلاحظ من الشاشة السابقة كيف أن title وهو مفتاح ترتيب كان اختياري أثناء تنفيذ عملية الاستعلام, علمًا أن المحتوى الأصلي للجدول books هو:

author	title	lang
a2	t2	en
a2	t2_1	ar
a2	t2_2	en
a2	t2_3	en
a1	t1	ar
a1	t2	ar
a1	t3	ar
a1	t4	en

تحديد عدد نتائج الاستعلام

عند استخدام الاستعلام للحصول على نتائج يمكن تحديد عدد النتائج التي سيتم استرجاعها, فمثلًا لو تم تحديد عدد النتائج 15 عندها سيتم إرجاع 15 نتيجة حتى لو كان التطابق مع القيم المحدد للمفتاح الرئيسي أكثر من ذلك وعندها يمكن استخدام ميزة عرض أكثر من صفحة للنتائج بحيث في كل صفحة يتم عرض 15 نتيجة وسيتم التطرق لها في قسم آخر.

لكن في حال تحديد فلاتر مع الاستعلام مثل lang=en وكان عدد العناصر المحدد هو 15 عندها سيتم إعادة عنصر واحد فقط بناء على المثال السابق وذلك لأن الفلتر يطبق على النتيجة بعد تنفيذ الاستعلام أي على الـ 4 عناصر التي سيتم استرجاعها في حال حددنا author = a1.

لذلك وبناء على العلاقة بين نتائج الاستعلام وطريقة عمل الفلاتر يجب تحديد عدد العناصر المراد استرجاعها بطريقة مدروسة مع اختيار الفلاتر بشكل صحيح.

تقسيم النتائج إلى صفحات وعدد النتائج

توفر DynamoDB آلية لتقسيم نتائج الاستعلام إلى صفحات حيث تحتوي كل صفحة على عدد محدد من النتائج ولكن في جميع الأحوال يجب ألا يتجاوز حجم النتائج في كل صفحة 1MB, علمًا أن عدد النتائج أيضًا يتأثر بعدد النتائج التي حددها التطبيق لاسترجاعها وقد شرحنا هذا في قسم سابق.

والآلية التي توفرها DynamoDB لدعم تعدد الصفحات تعتمد على العنصر LastEvaluatedKey, ففي حال وجود قيمة لهذا العنصر في النتائج فهذا يعني وجود نتائج أخرى في الصفحة التالية وإلا فلا توجد نتائج إضافية يمكن عرضها.

وللحصول على نتائج الصفحة التالية يجب أخذ قيمة العنصر LastEvaluatedKey وبناء صيغة استعلام تعتمد جميع القيم المحددة مسبقًا مع تحديد قيمة للعنصر ExclusiveStartKey تساوي القيمة LastEvaluatedKey.

ملاحظة: لا توجد طريقة لمعرفة هل توجد نتائج إضافية في الصفحة التالية أم لا إلا بالاعتماد على وجود أم عدم وجود قيمة للعنصر LastEvaluatedKey.

ويجب الانتباه أنه عند استخدام AWS SDK للاستعلام من جداول DynamoDB فإن هذه المكتبة تعالج موضوع وجود صفحات إضافية أم لا بطريقة جدًا سهلة وعملية.

ومن المعلومات الإضافية التي تعود من نتائج الاستعلام هي العناصر التالية:

- ScannedCount وتعبر عن عدد العناصر التي تتطابق مع قيمة المفتاح الرئيسي قبل تطبيق الفلاتر
- Count وتعبر عن عدد النتائج المتبقية بعد تطبيق الفلاتر

وفي حال عدم وجود فلاتر فإن القيمتين ScannedCount و Count متساويتان, ويجب الانتباه إلى أنه في حال أن حجم النتائج المسترجعة أكبر من 1MB فإن ScannedCount و Count لن تعبر عن قيم دقيقة لعدد النتائج.

استخدام الفهارس الثانوية مع جداول DynamoDB

تعتبر الفهارس الثانوية إحدى أهم الطرق المستخدمة في DynamoDB لتسريع عملية الوصول إلى البيانات المخزنة في الجداول وخاصة عندما تكون الحاجة للوصول إلى هذه البيانات تعتمد على مفاتيح رئيسية مختلفة عن تلك المحددة في الجدول.

والفهرس الثانوي يرتبط بجدول واحد فقط ويحتوي على جميع بيانات الجدول أو جزء منها حسب ما يتم تحديده أثناء إنشاء الفهرس الثانوي. ويتم تحديث بيانات الفهارس الثانوية بناء على ما يتم إضافته, أو تعديله أو حذفه من الجدول الرئيسي المرتبط به الفهرس, وتتم هذه العملية بشكل تلقائي من قبل DynamoDB.

وتوفر DynamoDB نوعين من الفهارس الثانوية:

- فهرس ثانوي عام Global Secondary Index: يحتوي على مفتاح رئيسي ومفتاح ترتيب قد يكونان مختلفان عن مفاتيح الجدول الرئيسي, ويطلق عليه الاسم GSI للاختصار.
- فهرس ثانوي محلي Local Secondary Index: يحتوي على نفس المفتاح الرئيسي للجدول مع إمكانية اختيار مفتاح ترتيب مختلف, ويطلق عليه الاسم LSI للاختصار.

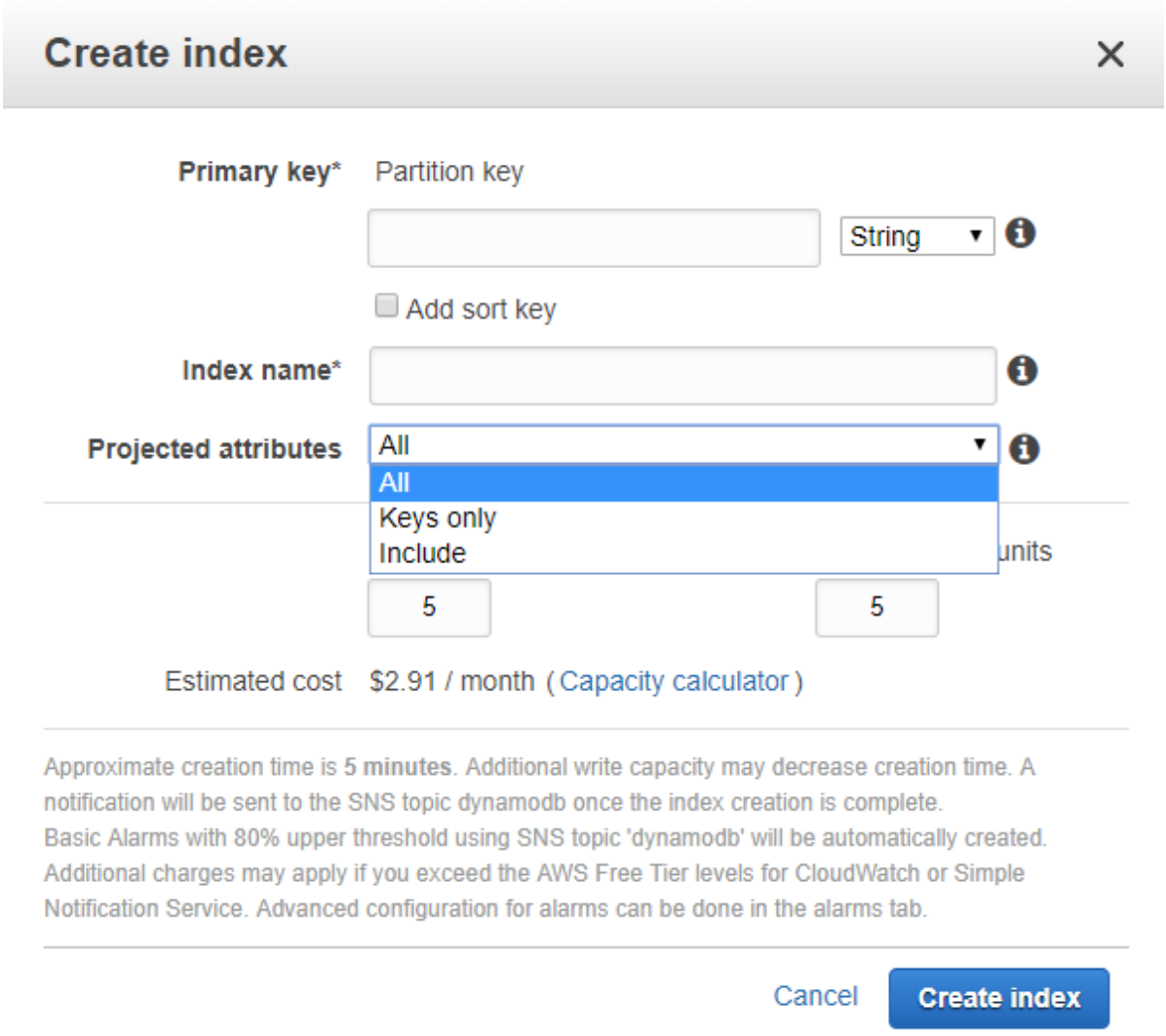
الفهرس الثانوي العام GSI

يعتبر الفهرس الثانوي العام من الأدوات الهامة التي يجب على مبرمجي DynamoDB استخدامها لتحسين آلية الوصول إلى البيانات المخزنة في الجداول مع الأخذ بعين الاعتبار النقاط التالية:

- يمكن إنشاء فهرس أثناء إنشاء الجدول أو إضافة فهرس بعد إنشاء الجدول

- يمكن إنشاء 5 فهارس عامة كحد أعلى لكل جدول
- يمكن اختيار أي عمود من أعمدة الجدول ليكون المفتاح الرئيسي للفهرس على أن يكون نوع العمود string, number, binary فقط
- يمكن استخدام أي عمود من أعمدة الجدول لكي يكون مفتاح الترتيب للفهرس بشرط أن يكون من أحد الأنواع الثلاثة السابقة, علمًا أن مفتاح الترتيب اختياري
- الاستعلام من الفهرس العام يدعم فقط Eventual Consistency
- توجد إعدادات خاصة بالفهرس العام مرتبطة بـ Read Capacity Units و Write Capacity Units
- يمكن فقط الحصول على البيانات المخزنة في الفهرس والتي يتم اختيارها عند إنشاء الفهرس
- المفاتيح المعرفة للفهرس العام ليس بالضرورة يجب أن تكون فريدة أي يمكن أن تتكرر بعكس المفاتيح المطبقة على الجدول

والشاشة التالية تبين كيفية إنشاء فهرس عام:



فيما يخص Partition key و Sort key ينطبق عليهما نفس الشروط التي تنطبق عليها في حال إضافتها إلى الجدول, وفيما يخص Projected attributes:

- All: سيتم إضافة جميع الأعمدة الموجودة في الجدول إلى الفهرس العام
- Keys only: سيتم إضافة مفتاح الجدول الرئيسي ومفتاح الترتيب إلى الفهرس فقط
- Include: يمكن تحديد أسماء الأعمدة التي نريد إضافتها من الجدول إلى الفهرس إضافة إلى أن DynamoDB تضيف مفاتيح الجدول بشكل تلقائي إلى الفهرس

مزامنة البيانات بين الجدول والفهرس العام

عند إنشاء فهرس عام مرتبط بجدول ما فإن DynamoDB تنسخ البيانات من الجدول وتخزنها في الفهرس والبيانات التي يتم نسخها تعتمد على الأعمدة التي تم تحديدها أثناء إنشاء الفهرس. ولكي تبقى البيانات المخزنة في الفهرس العام متوافقة مع نظيرتها في الجدول فإن DynamoDB عند تنفيذ أي عملية إضافة أو تعديل أو حذف على الجدول فإنها تعكس هذه التعديلات بطريقة غير متزامنة على جميع الفهارس العامة المرتبطة بالجدول.

وبما أن عملية المزامنة بين الجدول والفهارس تتم بطريقة غير متزامنة فإن بعض القراءات من الفهرس العام قد ترجع بيانات لا تمثل البيانات الحقيقية وذلك لأن عملية الزامنة قد لا تكون قد انتهت لذلك فإن عملية قراءة البيانات من الفهرس العام تعتمد على النمط Eventual Consistence.

وبما أن أي عملية كتابة تتم على الجدول يتم عكسها على الفهرس العام المرتبط بها فإن إعدادات Write Capacity Units المرتبطة بالفهرس العام تؤثر على عدد العناصر التي يمكن كتابتها خلال ثانية واحدة في الجدول وسيؤدي ذلك إلى حدوث خطأ مرتبط بعدد العناصر التي يمكن كتابتها خلال ثانية واحدة في الجدول المرتبط به الفهرس العام.

أيضًا يجب الانتباه إلى DynamoDB قد لا تزامن جميع العناصر التي تتم كتابتها في الجدول إلى الفهرس العام وذلك عندما لا يتم توفير أحد الأعمدة التي يجب أن تنسخ إلى الفهرس أثناء الكتابة في الجدول, وكمثال على ذلك جدول الكتب books الذي يحتوي على العمود lang. لو أنشأنا فهرس يحتوي على العمود lang كعمود مفتاح رئيسي أو مفتاح ترتيب وكتبنا عنصر جديد في الجدول books بدون توفير قيمة العمود lang كما يلي:

```
{
  "author": "a3",
  "title": "t3"
}
```

عندها لن يتم نسخ هذا السطر إلى الفهرس العام وذلك لأن العمود lang غير متوفر في السطر الجديد, ويجب التفريق بين تمرير قيمة فارغة أو NULL إلى العمود lang أو عدم ذكره تمامًا في الـ JSON الذي يتم إرساله.

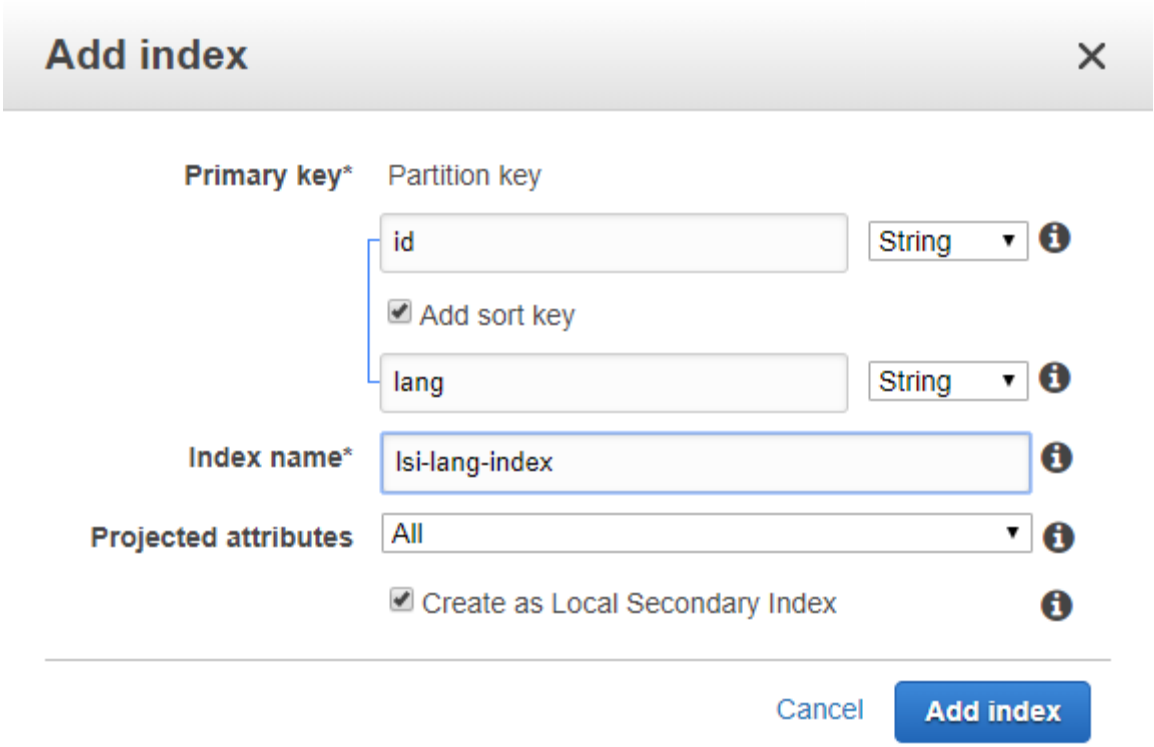
الفهرس الثانوي المحلي LSI

يُستخدم الفهرس الثانوي المحلي بشكل رئيسي للاستعلام عن البيانات باستخدام نفس المفتاح الرئيسي للجدول ولكن بمفتاح ترتيب مختلف, وبالتالي الحصول على نفس البيانات ولكن بترتيب مختلف. ويمكن تحديد أي عمود من الجدول على أنه مفتاح الترتيب بشرط أن نوعه String, Number, Binary. وعند إنشاء فهرس محلي يجب الانتباه إلى النقاط التالية:

- يمكن إنشاء فهرس أثناء إنشاء الجدول فقط
- يمكن إنشاء 5 فهارس محلية كحد أعلى لكل جدول
- المفتاح الرئيسي للفهرس هو نفسه المفتاح الرئيسي للجدول
- يمكن استخدام أي عمود من أعمدة الجدول لكي يكون مفتاح الترتيب للفهرس بشرط أن يكون من أحد الأنواع الثلاثة السابقة, علمًا أن مفتاح الترتيب إجباري
- الاستعلام من الفهرس العام يدعم فقط Eventual Consistency و Strong Consistency

- يمكن الحصول على البيانات المخزنة في الفهرس والتي يتم اختيارها عند إنشاء الفهرس إضافة إلى البيانات المخزنة في الجدول والتي لم تتم إضافتها إلى الفهرس

ويتم إنشاء فهرس ثانوي محلي أثناء إنشاء الجدول ولا يمكن إنشاؤه بعد إنشاء الجدول علمًا أن المفتاح الرئيسي للفهرس المحلي هو نفس المفتاح الرئيسي للجدول, والشاشة التالية تبين كيفية إنشاء هذا الفهرس:



يجب أن نلاحظ بأنه أثناء إنشاء الجدول لا يمكن تحديد أعمدة إضافية غير المفتاح الرئيسي ومفتاح الترتيب للجدول, ومع ذلك تم اختيار العمود lang كمفتاح ترتيب للفهرس المحلي علمًا أنه لم تتم إضافته إلى الجدول, ولكن DynamoDB تسمح بهذه العملية لأنه وكما ذكرنا سابقًا لا يوجد أعمدة محددة لجداول الـ DynamoDB.

وبما أنه تمت إضافة lang على أنه مفتاح ترتيب للفهرس المحلي فإن قيمة هذا العمود ستكون إجبارية عند إنشاء عنصر جديد للجدول, وفيما يخص Projected attributes:

- All: سيتم إضافة جميع الأعمدة الموجودة في الجدول إلى الفهرس العام
- Keys only: سيتم إضافة مفتاح الجدول الرئيسي ومفتاح الترتيب إلى الفهرس فقط
- Include: يمكن تحديد أسماء الأعمدة التي نريد إضافتها من الجدول إلى الفهرس إضافة إلى أن DynamoDB تضيف مفاتيح الجدول بشكل تلقائي إلى الفهرس

ملاحظة: لا يمكن أن يتجاوز حجم البيانات المخزنة في الفهرس 10G.

ويجب الانتباه إلى أن إعدادات عدد العناصر التي يمكن قراءتها في الثانية الواحدة هي نفسها إعدادات الجدول, ما عدا أن عدد الوحدات تعتمد على حجم العنصر في الفهرس وليس حجم العنصر في الجدول.

استخدام لغات البرمجة مع DynamoDB

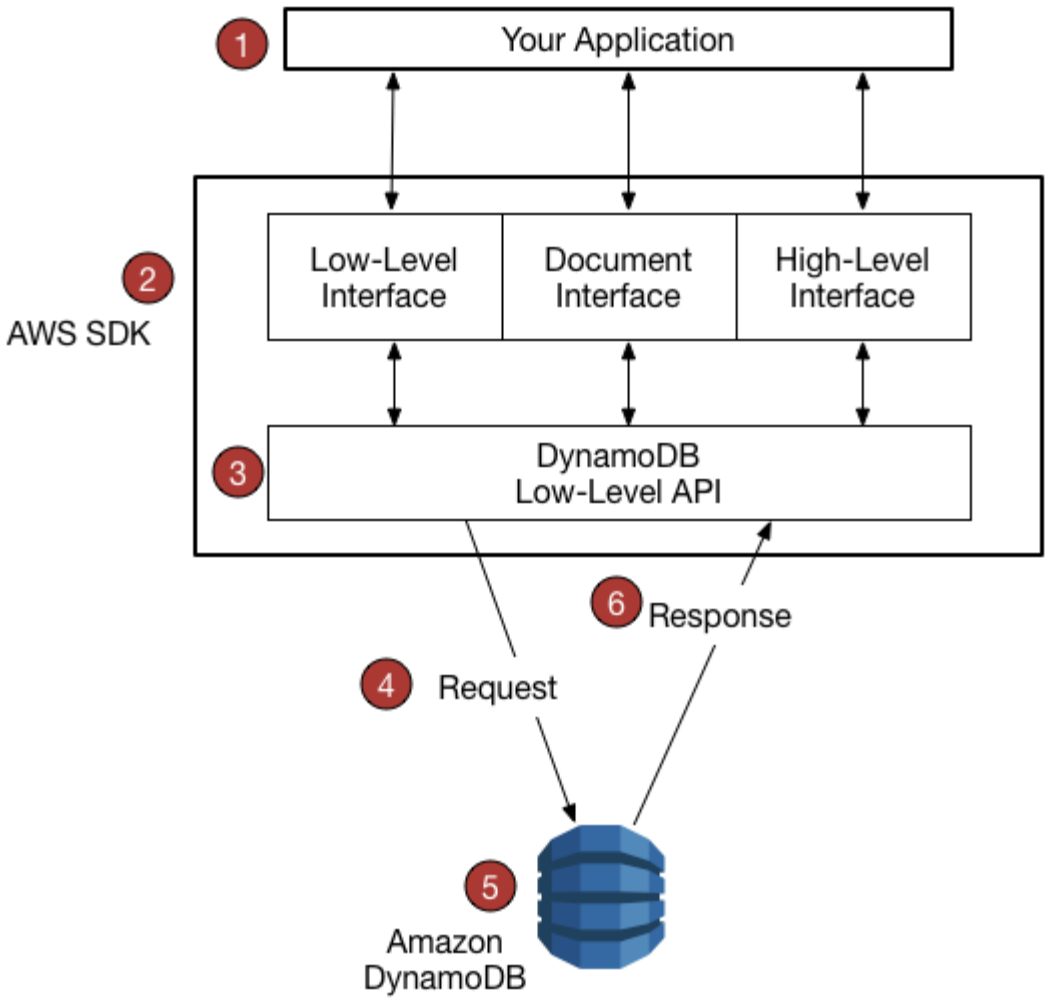
يمكن استخدام لغات برمجة مختلفة للتعامل برمجيًا مع قاعدة بيانات DynamoDB ومع جداولها (إنشاء, كتابة, استعلام, ...) وتقريبًا جميع المكتبات املتوفرة توفر نفس الميزات مع بعض الاختلافات. والمميز في الموضوع أن فريق DynamoDB في أمازون هو المسؤول عن صيانة هذه المكتبات وتعديلها للتوافق مع ميزات DynamoDB. علمًا أن المكتبات تتوفر باللغات البرمجية التالية:

- Java
- JavaScript
- Node.js
- .NET
- PHP
- Python
- Ruby

ويمكن الوصول إلى هذه القائمة من خلال الرابط التالي:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/GettingStarted.html

والمخطط التالي يبين آلية عمل المكتبات البرمجية مع DynamoDB



علمًا أن توفر واجهة برمجة التطبيقات المحددة في النقطة رقم 2 يعتمد على اللغة البرمجية لكن جميعها توفر Low-Level Interface.

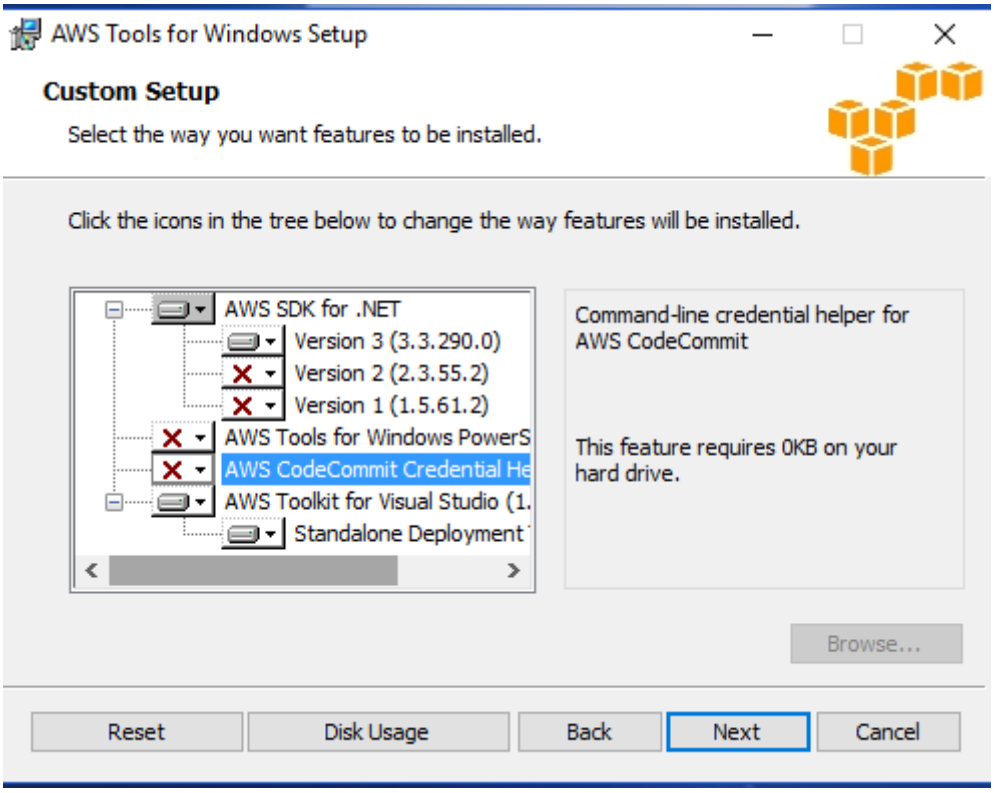
تجهيز بيئة العمل لاستخدام DynamoDB SDK

في الرابط التالي يمكنك إيجاد جميع الأدوات التي تساعدك على البدء بتنفيذ مهام برمجية باستخدام لغتك المفضلة

https://aws.amazon.com/tools/

الخطوات التالية هي لتثبيت أدوات البرمجة باستخدام منصة عمل .NET.

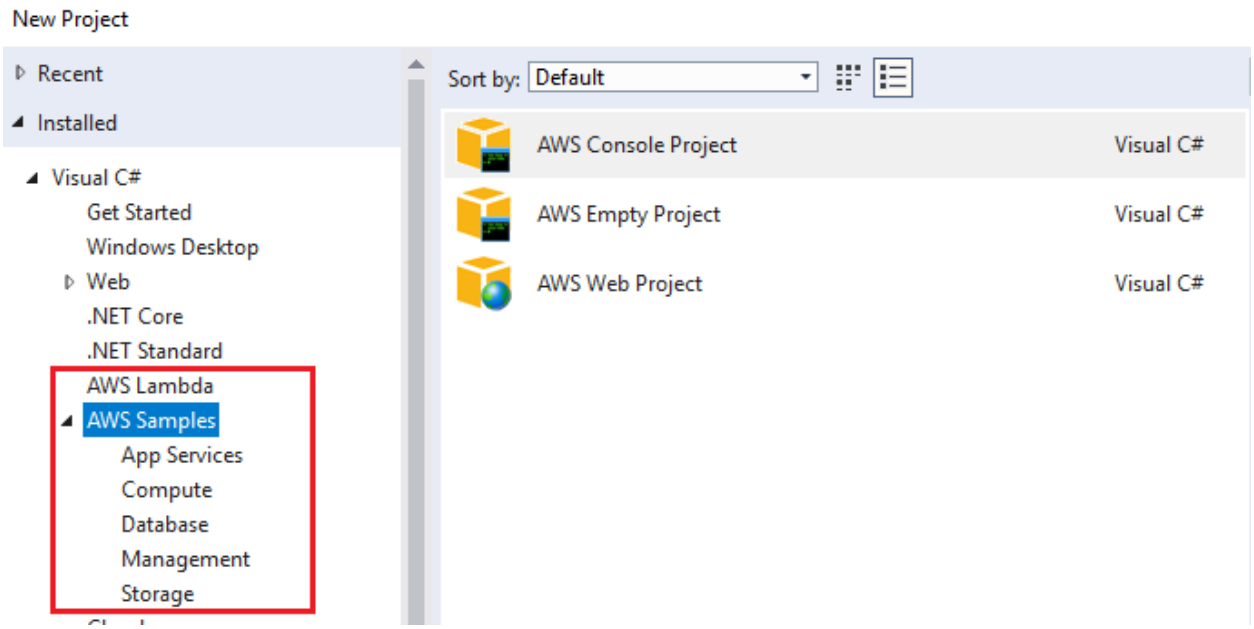
1. تثبيت مكتبة .NET SDK من موقع أمازون وذلك من خلال الرابط السابق, ومن ثم الانتقال إلى القسم SDKs والضغط على Install تحت .NET.



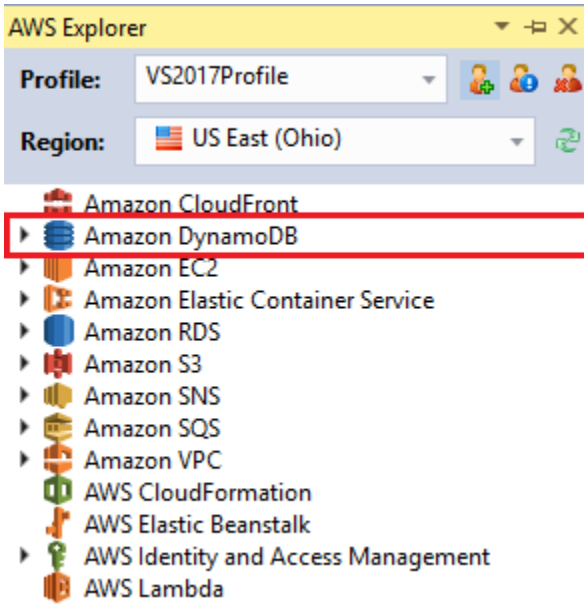
2. بعد تنفيذ الخطوة الأولى, يفضل تثبيت أدوات مساعدة تعمل مع Visual Studio مباشرة وذلك لتسهيل عمل المطور, وباعتبار أننا نستخدم Visual Studio 2017 فيجب تثبيت هذه الأدوات من سوق مايكروسوفت مباشرة على الرابط التالي:

https://marketplace.visualstudio.com/items?itemName=AmazonWebServices.AWSToolkitforVisualStudio2017

هذه الأدوات توفر قوالب جاهزة لإنشاء مشاريع لمهام متعددة تستخدم AWS SDK كما هو مبين بالشكل التالي:

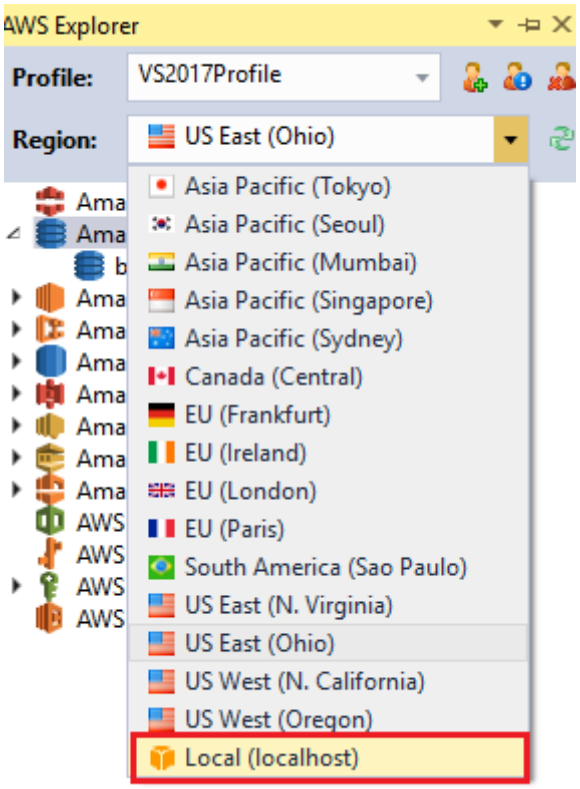


كما توفر AWS Explorer والذي من خلاله يمكن تنفيذ العديد من العمليات على خدمات AWS ومنها DynamoDB. يومكن الوصول إلى AWS Explorer من VIEW ومن ثم الضغط على AWS Explorer لتظهر شاشة تشبه الشاشة التالية:

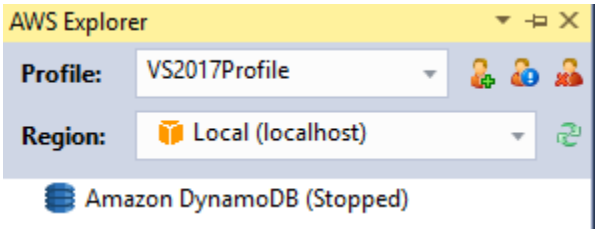


ومن الميزات التي يوفرها AWS Explorer هي إمكانية استخدام DynamoDB بدون الاعتماد على Cloud service حيث يتم تثبيت DynamoDB على الجهاز والاتصال بقاعدة البيانات باستخدام localhost. ويمكن تثبيتها باتباع الخطوات التالية:

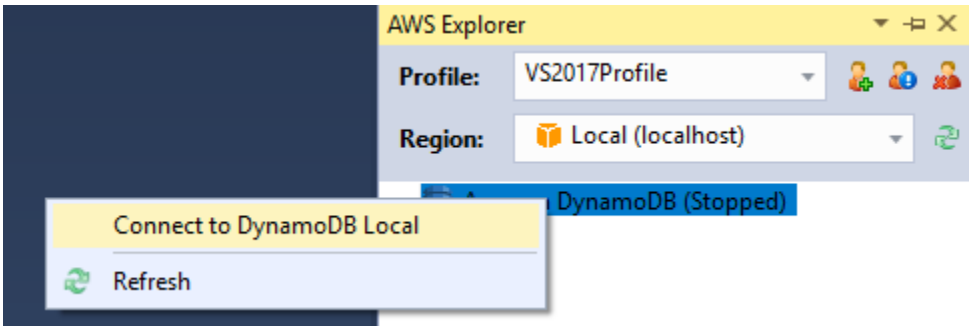
1. افتح AWS Explorer
2. من القائمة Region اختر Local



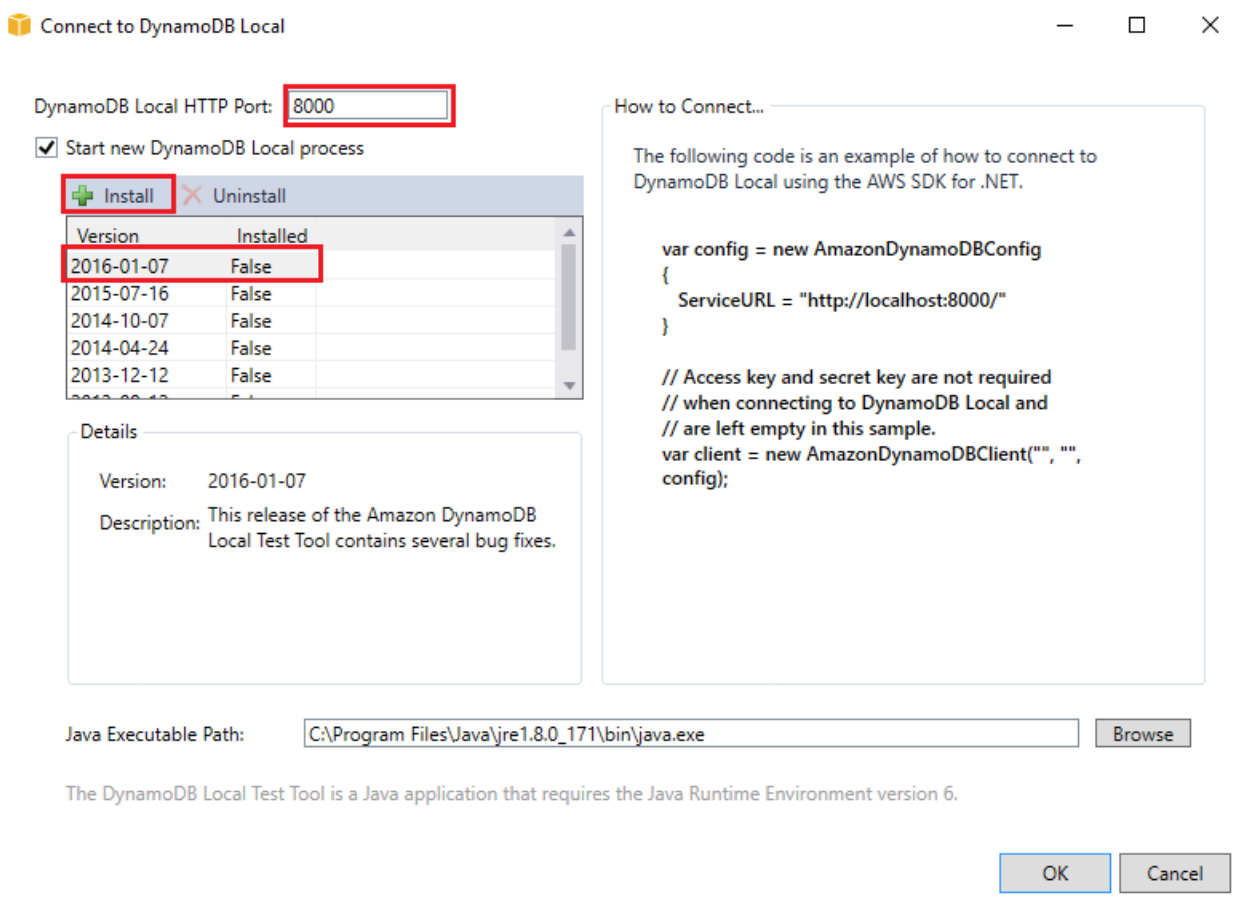
عندها يتم فتح الشاشة التالية:



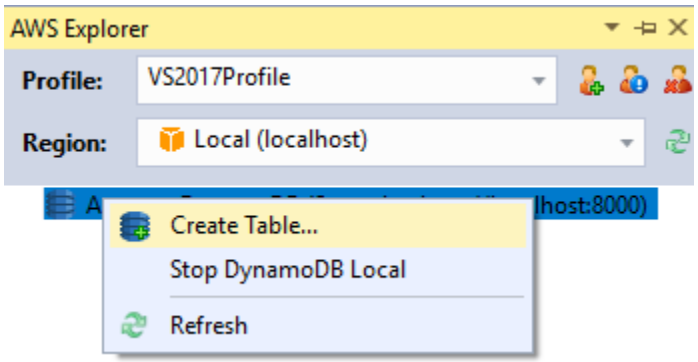
3. اضغط Right click على Amazon DynamoDB واختر الأمر Connect to DynamoDB Local



4. اختر آخر إصدار واضغط الأمر Install



5. بعد تثبيت الإصدار المطلوب يمكن الآن إنشاء الجداول على قاعدة البيانات والتعامل معها بدون الحاجة للاتصال بالإنترنت كما هو مبين في الشكل التالي



ملاحظة: العمليات التي يمكن تطبيقها على الجداول من خلال Local DynamoDB محدودة ولا تتوفر جميع ميزات DynamoDB لكنها تساعد بشكل كبير لبناء تطبيقات تعتمد على الجداول دون الحاجة لخدمة DynamoDB Cloud.

البرمجة باستخدام Low-Level Interface

جميع المكتبات التي توفرها أمازون للبرمجة مع DynamoDB تدعم هذا المستوى والذي يسمح لك بتحديد نوع العمود الذي تريد التعامل معه.

البرمجة باستخدام Document

باستخدام هذه الطريقة لا حاجة لتحديد نوع البيانات التي تمررها حيث يتم ذلك بشكل ضمني من قبل AWS SDK, ولا تتوفر هذه الطريقة لجميع لغات البرمجة المدعومة من قبل أمازون ولكن بشكل أساسي فهي متوفرة للغات التالية Java, .NET, .Node.js, and JavaScript

البرمجة باستخدام Object Persistence

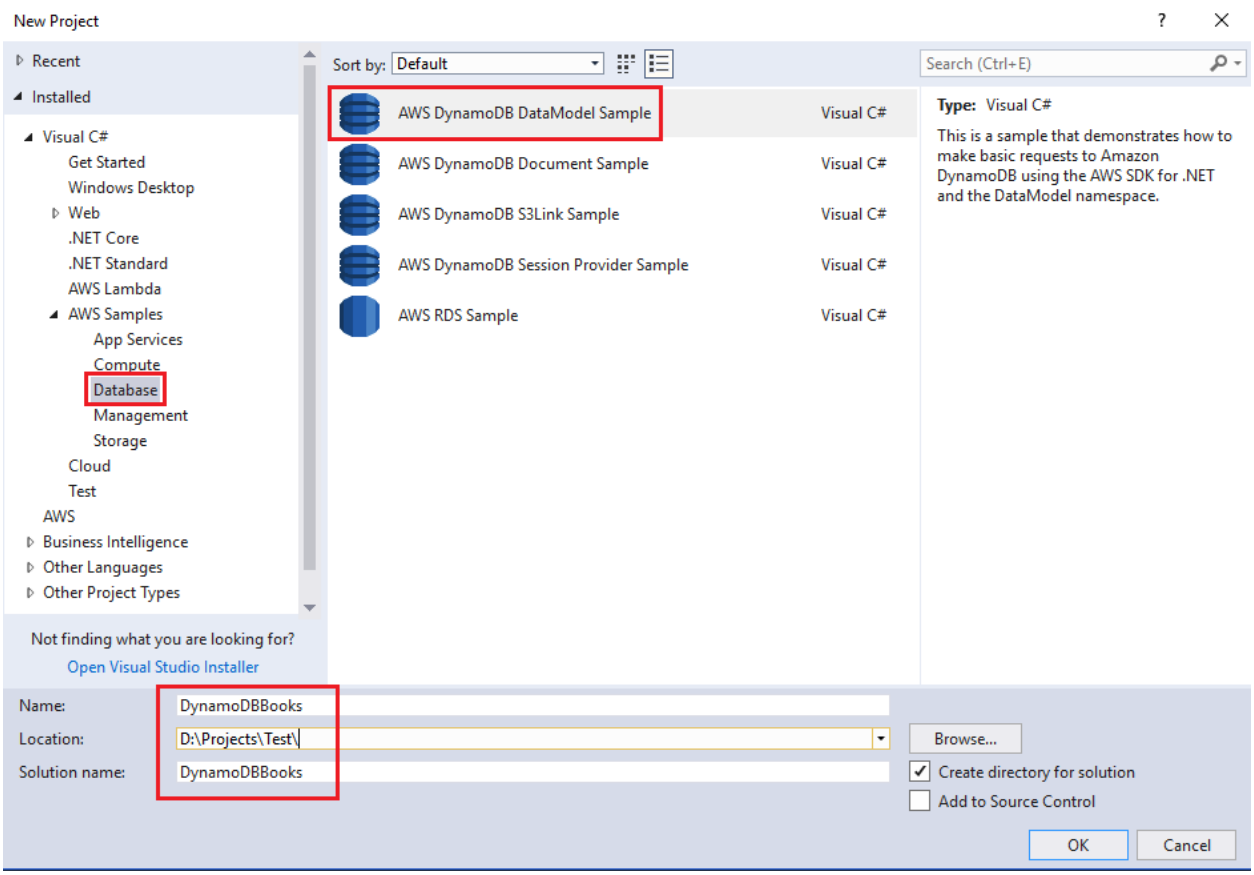
باستخدام هذه الطريقة يتم إنشاء class يمثل جدول أو فهرس ويتم التعامل مع مثيل عن هذا الصف بشكل مباشر, وتشبه هذه الطريقة استخدام تقنية الـ ORM ومنها استخدام الـ Entity Framework لمن يستخدمون لغة #C. وتعتبر هذه الطريقة الأسهل للبرمجة مع DynamoDB.

مثال عن البرمجة باستخدام Object Persistence

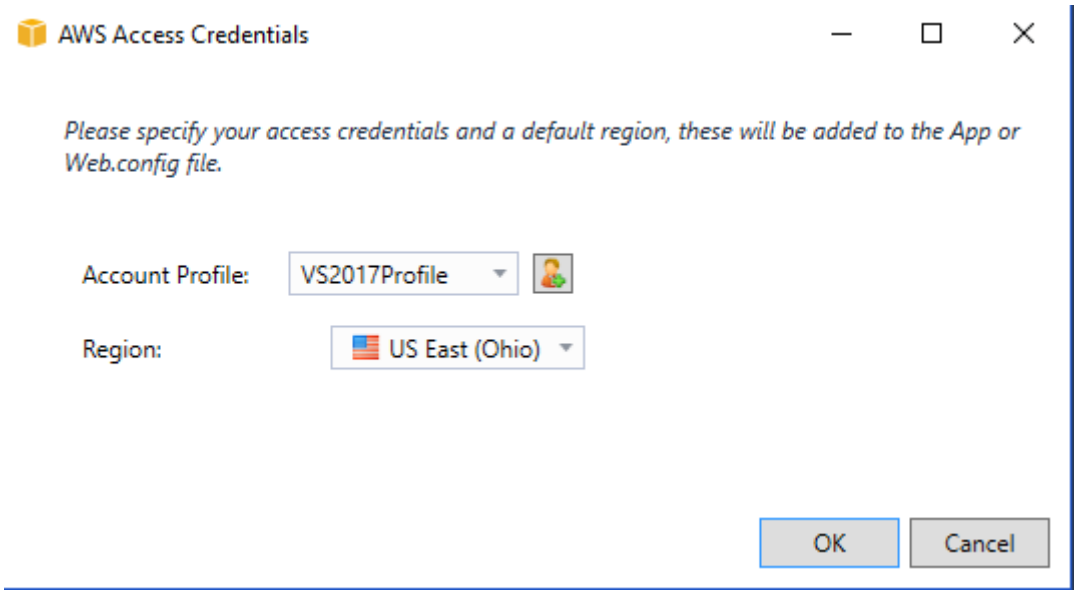
في هذا المثال سنعتمد البنية التالية لجدول الكتب books:

- author
- title
- lang

أولًا سيتم إنشاء مشروع باستخدام القالبAWS DynamoDB DataModel Sample كما يلي:



عند اختيار OK سيظهر النموذج التالي الذي يهدف إلى تحديد معلومات المستخدم التي سيتم استخدامها للاتصال بخدمة DynamoDB على AWS:



ملاحظة: آلية إنشاء Profile وربطه مع Visual Studio هي خارج نطاق هذه الوثيقة, ولمزيد من المعلومات يمكن زيارة الرابط التالي:

https://docs.aws.amazon.com/toolkit-for-visual-studio/latest/user-guide/credentials.html

احذف الملفات التالية بعد إنشاء المشروع:

- DataModelSample
- DataModelSchema
- TableOperations

أنشئ الصف BookModel كما يلي:

```
[DynamoDBTable("books", LowerCamelCaseProperties = true)]
public class BookModel
{
    public string Author { get; set; }

    public string Title { get; set; }

    public string Lang { get; set; }
}
```

عند استخدام هذه الطريقة للبرمجة مع DynamoDB يجب استخدام الـ Attributes للربط بين معلومات الصف الذي تم إنشاؤه وبين معلومات الجدول في DynamoDB, وفيما يلي شرح لما تم استخدامه في الصف BookModel:

- DynamoDBTable: الهدف من استخدامها هو ربط الصف BookModel مع الجدول books, حيث تم تمرير اسم الجدول كأول وسيط.
- أما استخدام LowerCamelCaseProperties مع القيمة true تعني تحويل أول حرف لأي property في الصف BookModel إلى حرف صغير قبل حفظه في الجدول books, أي أن Author ستكون author, وسبب هذه النقطة هو أن Author تختلف عن author حيث أن DynamoDB حساسة لحالة الأحرف.

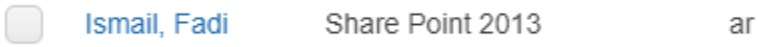
ملاحظة: تم استخدام نفس أسماء أعمدة الجدول books ضمن الصف BookModel.

الصف BookModel بشكله السابق هو أبسط شكل لربط صف مع جدول في DynamoDB ويمكن الآن متابعة كتابة البرنامج لتخزين عنصر جديد في الجدول books.

```
public static void Main(string[] args)
{
        CreateBookUsingObject();

        Console.WriteLine("Done!");
        Console.Read();
}

private static void CreateBookUsingObject()
{
        BookModel book = new BookModel()
        {
                Author = "Ismail, Fadi",
                Title = "Share Point 2013",
                Lang = "ar"
        };

        using (IAmazonDynamoDB db = new AmazonDynamoDBClient())
```

```
        {
                DynamoDBContext cntxt = new DynamoDBContext(db);
                cntxt.Save(book);
          }
}
```

عند تنفيذ الشيفرة السابقة بنجاح سيتم كتابة سطر جديد في الجدول books كما يلي:



من المثال السابق نلاحظ مدى سهولة استخدام هذه الطريقة مع DynamoDB.

قراءة سطر من الجدول

من الوظائف الشائعة للتعامل مع الجداول هي قراءة سطر واحد من خلال المفتاح الرئيسي, ويمكن تنفيذ ذلك من خلال الشيفرة التالية:

```
private static void ReadBookUsingObjectModel()
{
        BookModel book = null;

        using (IAmazonDynamoDB db = new AmazonDynamoDBClient())
        {
                DynamoDBContext cntxt = new DynamoDBContext(db);
                book = cntxt.Load<BookModel>("a2", "t2");
                if (book != null)
                {
                        Console.WriteLine(book.Title);
                }
         }
}
```

في السطر cntxt.Load تم تمرير قيمتين, والسبب أننا عند إنشاء الجدول books ذكرنا بأن author هو مفتاح رئيسي و title هو مفتاح ترتيب, لذلك عند الاستعلام عن سطر محدد باستخدام Load من الجدول يجب تمرير قيمة المفتاح الرئيسي وقيمة مفتاح الترتيب.

قراءة أكثر من سطر من الجدول

في المثال التالي سيتم الاستعلام عن جميع الأسطر التي تشترك بقيمة المفتاح الرئيسي ذاتها:

```
private static void GetBooks()
{
        using (IAmazonDynamoDB db = new AmazonDynamoDBClient())
        {
                DynamoDBContext cntxt = new DynamoDBContext(db);
                IEnumerable<BookModel> books = cntxt.Query<BookModel>("a2");

                foreach (BookModel b in books)
                        Console.WriteLine("{0}\t{1}\t{2}", b.Author, b.Title, b.Lang);
        }
}
```

في المثال الأخير الاختلاف عن شيفرة قراءة سطر واحد من الجدول هو استخدام Query بدلًا من Load.

تغيير اسم العمود

في الصف BookModel استخدمنا نفس أسماء الأعمدة في الجدول books, لكن ماذا لو أردنا استخدام الاسم BookTitle لتمثيل العمود title من نفس الجدول؟ يمكن تنفيذ ذلك من خلال استخدام DynamoDBProperty كما يلي:

```
[DynamoDBTable("books", LowerCamelCaseProperties = true)]
public class BookModel
{
    public string Author { get; set; }

    [DynamoDBProperty("title")]
    public string BookTitle { get; set; }

    public string Lang { get; set; }
}
```

وهذا يسمح لنا باستخدام أسماء خصائص الصف بحيث تكون مختلفة عن أسماء الأعمدة في الجدول.

وأيضًا يمكن استخدام DynamoDBHashKey لتحديد المفتاح الرئيسي للجدول في حال عدم استخدام نفس اسم العمود في الجدول كما يلي:

```
[DynamoDBHashKey("author")]
public string BookAuthor { get; set; }
```

ولتحديد اسم مفتاح الترتيب يجب استخدام DynamoDBRangeKey كما يلي:

```
[DynamoDBRangeKey("title")]
public string BookTitle { get; set; }
```

حذف سطر من جدول

لحذف سطر من جدول يحتوي على مفتاح رئيسي بدون مفتاح ترتيب يتم استخدام الشيفرة التالية:

```
cntxt.Delete<BookModel>("a2");
```

في المثال السابق تم تمرير القيمة a2 على أنها قيمة نصية لأن عمود المفتاح الرئيسي في الجدول books نوعه قيمة نصية, علمًا أن التابع Delete يقبل قيمة المفتاح الرئيسي من نوع object وذلك لكي يتم تمرير قيمة نصية أو رقم أو قيمة ثنائية.

أما في حال أن في حال وجود مفتاح ترتيب أيضًا, عندها يجب تمرير قيمة لمفتاح الترتيب إلى جانب المفتاح الرئيسي كما يلي:

```
cntxt.Delete<BookModel>("a2", "t2");
```

ويمكن حذف سطر من خلال مثيل الكائن كما يلي:

```
BookModel book = cntxt.Load<BookModel>("a2", "t2_1");
cntxt.Delete(book);
```

حيث يتم الحصول على معلومات الكائن من خلال التابع Load ومن ثم تمريره إلى التابع Delete.

ولمعرفة كيفية استخدام Java أو .NET. مع بعض الأمثلة للبرمجة مع DynamoDB يمكن الاطلاع على الرابط التالي:

https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/HigherLevelInterfaces.html

النسخ الاحتياطي

كيف يمكن إدارة النسخ الإحتياطية في قاعدة بيانات DynamoDb؟

للإجابة على هذا السؤال يجب أولا معرفة أي نموذج من نماذج الحوسبة السحابية تندرج قاعدة بيانات DynamoDb وما هي النقاط التي تندرج تحت مسؤولية أماوزن ومسؤولية العميل أو مسؤولياتنا.

يوجد ثلاث أنواع من نماذج الحوسبة السحابية وهي كتالي:

- خدمات Infrastructure

وهي تتضمن توفير خدمات السيرفرات ومساحات التخزين الخاصة بالسيرفرات حيث في هذا النوع تكون مسؤولية أماوزن توفير العتاد وحمايته أما مسؤولياتنا تبدأ من نظام التشغيل وما فوق ذلك. فعلى سبيل المثال تحديث نظام التشغيل وحمايته من الفيروسات هو أحد مسؤولياتنا أما حماية الجزء الفيزيائي أو العتاد من التعطل أو تزويد الكهرباء فهي مسؤولية أمازون.

مثال على هذه الخدمات على بيئة أمازون هي خدمة EC2 أو خدمة EBS.

- خدمات Container

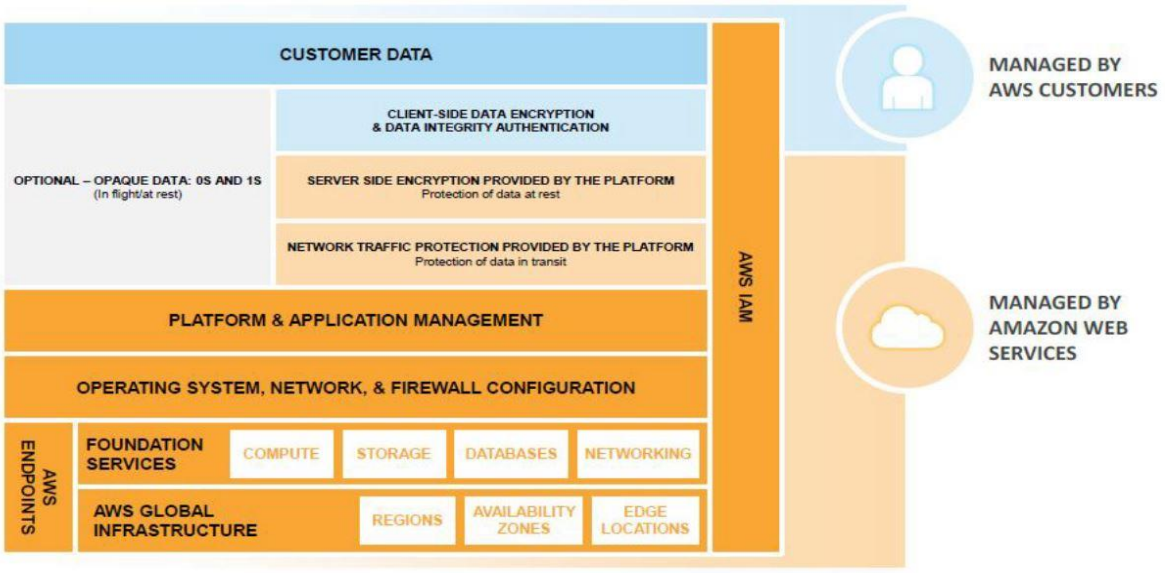
وهي تتضمن توفير خدمات على شكل منصات أو ما يأتي بعد نظام التشغيل مثل قاعدة بيانات SQL Server أو سيرفر الويب IIS حيث تكون مسؤولية أمازون نظام التشغيل وما دون ذلك من العتاد والكهرباء أما مسؤولياتنا تبدأ من المنصة وما فوق ذلك فمثلا تحديث نظام التشغيل وحمايته من الفيروسات هو من مسؤوليات أمازون أما حماية المنصة وأخذ نسخ احتياطية من البيانات فهي تندرج تحت مسؤولياتنا.

مثال على هذه الخدمات على بيئة أمازون هي خدمة RDS أو خدمة Beanstalk.

- خدمات Abstracted

وهي تتضمن توفير خدمات بحيث تكون إدارة المنصة وتحديثها وحماية العتاد والبرامج التي عليها من مسؤولية أمازون أما المستخدم فتكون مسؤوليته هو فقط الوصول للمعلومات وتخزينها دون الولوج في تفاصيل المنصة وإدارتها.

مثال على هذه الخدمات على بيئة أمازون هي خدمة S3 و خدمة قاعدة بيانات DynamoDb حيث كما في الشكل الموضع في الأسفل تكون مسؤولية المستخدم أو مدير النظام هو فقط الجزء الأزرق (الوصول للبيانات أو بيانات العميل).



من المهام التي أزيلت من قائمة العميل أو المستخدم وأصبحت تحت إدارة أمازون بما يخص قاعدة بيانات DynamoDb هي كتالي:

- إدارة التنصيب و الإعدادات
- حماية قاعدة البيانات بتخزين نسخ احتياطية في ثلاث أماكن مختلفة من المناطق Availability Zone
- إدارة توفير العتاد وتزويده في حالة التوسع
- إدارة تحديث البرامج
- توفير واجهات التطبيق البرمجية مثل API أو أوامر Cli وغيرها من الأدوات

بما أن قاعدة بيانات DynamoDb تندرج تحت الخدمات التي تسمى Fully managed أو Database as service فهل هذا يعني إني لا احتاج إلى عمل نسخ احتياطية حيث تقوم أمازون بإدارة العتاد وحماية قاعدة بيانات DynamoDb من الفشل أو حدوث توقف للخدمة أوحماية البيانات من الضياع ؟

الجواب لا, لأنه ما زال هناك احتمال لحدوث الخطأ أو ضياع البيانات والتي سببها غالبا خطأ برمجي أو حذف البيانات بشكل خاطئ أو حتى عملية اختراق لتطبيق العميل أو المستخدم حيث يؤدي ذلك إلى ضياع البيانات.

من هنا يتبين أن عميلة أخذ نسخ احتياطية من النقاط المهمة التي تندرج تحت مسؤولياتنا لحماية البيانات من الضياع.

يوجد العديد من الطرق التي تساعد على عمل نسخ احتياطية لقاعدة بيانات DynamoDb لكن لا يوجد حتى هذه اللحظة حل كامل مدعوم بشكل مباشر داخل قاعدة بيانات DynamoDb يدعم جميع الإحتياجات المتعلقة بعمليات النسخ الإحتياطي لذلك نجد بعض هذه الطرق مدعومة بشكل مباشر في خدمة قاعدة بيانات DynamoDb وبعضها يمكن عمله عن طريق الإستفادة من خدمات أمازون الأخرى مثل خدمة S3 وخدمة Lambda.

يوجد طريقتين لأخذ نسخ احتياطية في قواعد البيانات, إما عمل نسخة احتياطية كاملة لقاعدة البيانات Full Backup أو عمل نسخة احتياطية تزايدية Incremental Backup.

لنبدأ بالطريقة الأولى:

عمل نسخة احتياطية كاملة لقاعدة بيانات DynamoDb

توفر قاعدة بيانات DynamoDb خاصية اخذ نسخ احتياطية عند الطلب On Demand وذلك من خلال صفحة إدارة قاعدة بيانات DynamoDb في منصة أمازون حيث تعتبر هذه الخاصية من الإضافات الحديثة والتي تسهل على المستخدم أخذ نسخ احتياطية دون التأثيرعلى الخدمة من ناحية السرعة والأداء حيث تتم هذه العميلة بشكل Async بأخذ نسخ احتياطية من الأماكن الأخرى التي يتم عكس بيانات DynamoDb عليها وهي كما ذكرنا سابقا ثلاث نسخ في ثلاث اماكن مختلقة Availability Zone.

مما يميز هذه الطريقة:

- أن عملية النسخ الاحتياطي تأخذ وقت قصير خلال دقائق تقريبا
- جميع ملفات النسخ الاحتياطية مشفرة
- لا يوجد عدد محصور أو معين لأخذ نسخ احتياطية
- يمكن عمل نسخ احتياطية أيضا عن طريق واجهات التطبيق البرمجية مثل API أو أوامر Cli
- عملية النسخ الاحتاطي تتضمن ايضا
  - فهارس الجدول Indexes
  - التدفق Streams
  - إعدادات RCUs و WCUs
- لكن لا تتضمن:
  - إعدادات Auto scaling
  - إعدادات IAM
  - مقاييس و المنبهات لخدمة Cloudwatch
  - علامات Tags
  - إعدادات Stream
  - إعدادات Time to live

أيضا من التحديات, لا يوجد آلية مباشرة لنقل نسخ الاحتياط من حساب إلى حساب آخر أو من إقليم إلى إقليم آخرى Region في أمازون لحماية البيانات في حالة تم اختراق الحساب الأول لذلك نحتاج إلى استخدام الخدمات الأخرى التي توفرها أمازون لنقل بيانات الجدول إلى حساب آخر.

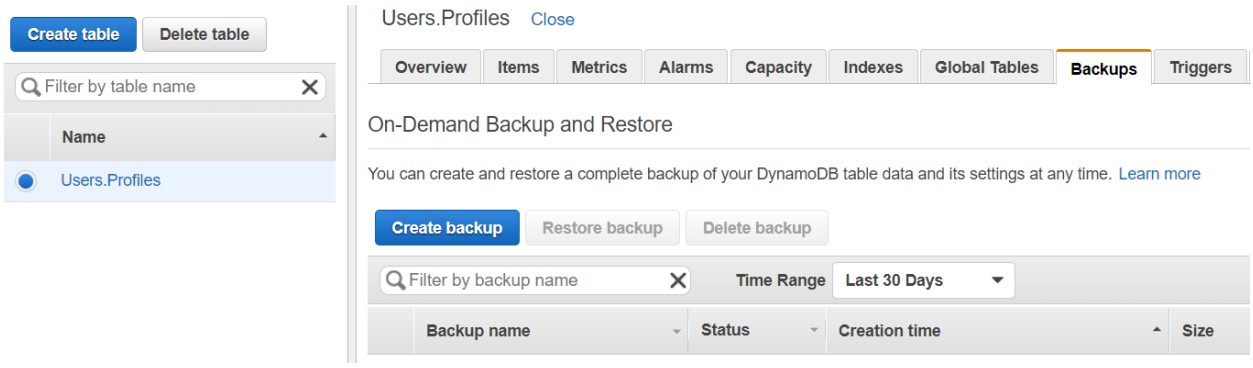
هذا النوع من النسخ الاحتياطي مناسب للحالات التي تحتاج إلى نسخ احتياطي عند الطلب أو لعمل بيئة التعافي من الكوارث.

أما بالنسبة لطريقة حساب التكاليف تكون فقط على مساحات التخزين لملفات النسخ الاحتياطي.

وأما بالنسبة لطريقة حساب نقطة استعادة البيانات Recovery Point Objective - RPO والوقت الازم للإستعادة Recovery Time Objective - RTO حيث RPO هو الفجوة الزمنية بين نسخ الاحتياط و RTO الوقت اللازم لاستعادة النظام.

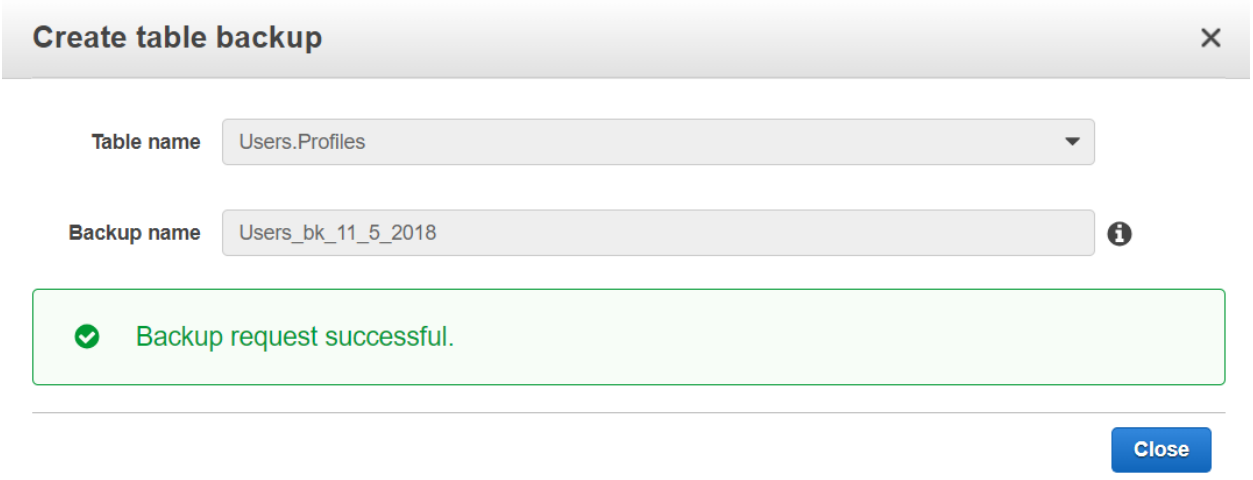
فمثلا إذا كان النسخ الاحتياطي يأخذ كل ساعة ففي هذه الحالة يكون RPO عالي حيث امكانية ضياع البيانات تصل إلى ساعة تقريبا أما عملية الاستعادة RTP فهي تأخذ وقت قصير فالاستعادة تكون خلال دقائق.

لأخذ نسخة احتياطية كاملة لجدول في قاعدة بيانات DynamoDb, اذهب إلى صفحة معلومات الجدول ثم في الجزء الأيمن ثم اضغط على جزء النسخ الاحتياطية Backups كما في الصورة

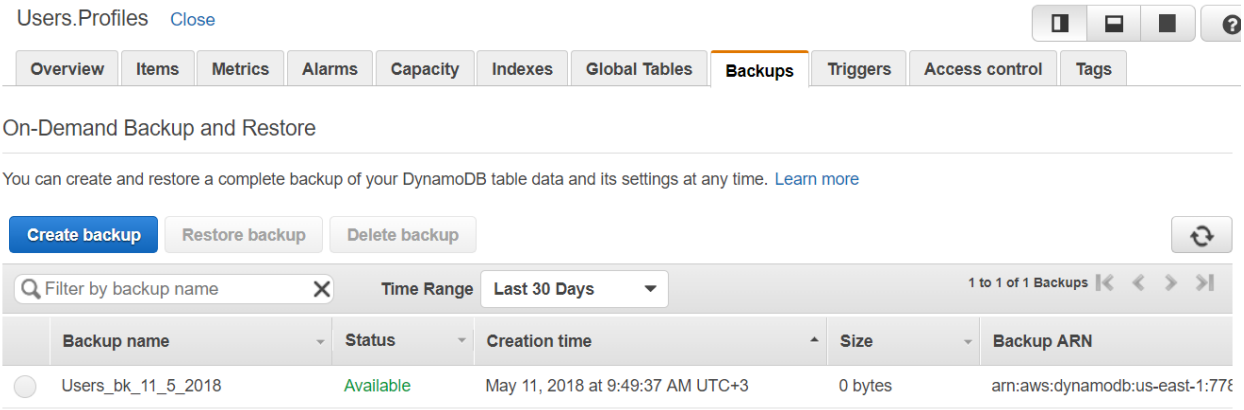


لعمل نسخة احتياطية عند الطلب لهذا الجدول , اضغط على زر انشاء نسخة احتياطية Create Backup ثم قم بإدخال اسم لنسخة الاحتياط التي سيتم اخذها

اضغط على زر انشاء Create



في هذه الحالة ستقوم هذا الخدمة بعمل نسخة احتياطية بشكل async دون التأثير على أداء قاعدة البيانات كما ذكرنا سابقا.



نلاحظ وجود Backup ARN حيث يساعد هذا المعرف للوصول إلى هذا الملف عن طريق واجهات التطبيق البرمجية مثل API أو أوامر Cli.

يمكن عمل عدد لا محدود من النسخ الاحتياطية لكن يجب الأخذ بالاعتبار أن أمازون ستقوم بإضافة تكلفة على مساحة تخزين هذه الملفات.

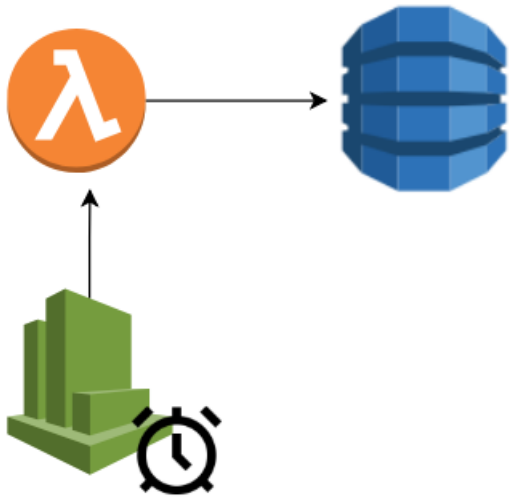
كيف يمكن توقيت عملية النسخ الاحتياطي الكامل بحيث تكون بشكل تلقائي ضمن وقت محدد؟

لجدولة عملية النسخ الاحتياطي الكامل, نحتاج إلى استخدام خدمات أمازون الأخرى مثل خدمة Lambda و خدمة Cloudwatch لأنه لا يوجد طريقة مباشرة من داخل لوحة تحكم قاعدة بيانات DynamoDb.

يوجد حلول كثيرة جاهزة على الإنترنت تعتمد على خدمة Lambda أو مفهوم Serverless وكلها تؤدي نفس الغرض, من هذه الحلول المعتمد استخدامها بين محترفي قاعدة بيانات DynamoDb هو الحل الموجود في Github على الرابط التالي:

https://github.com/abhayachauhan/dynamodb-ondemand-backups-scheduler

حيث يعتمد هذا الحل على إستخدام خدمة Cloudwatch event مع توقيت زمني من أجل استدعاء شيفرة المصدر أو الإجراء الموجودة في خدمة lambda التي تقوم بإنشاء نسخة احتياطية لجدول موجود في قاعدة بيانات DynamoDb في كل مرة يتم استدعائها.



لتنفيذ الحل قم بعمل الخطوات التالية:

أولا قم بتحميل الملفين كما الصورة

الملف الأول ondemand-backups-scheduler-policy.json يحدد الصلاحيات التي سوف يستخدمها الإجراء lambda مع خدمات أماوزن الأخرى والسماح له بعمل هذه العمليات وهي كتالي:

امكانية استدعاء اجراء lambda

```
{
  "Effect": "Allow",
  "Action": [
    "lambda:InvokeFunction"
  ],
  "Resource": [
    "*"
  ]
},
```

امكانية كتابة سجلات التدقيق في Cloudwatch

```
{
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:*"
},
```

امكانية انشاء نسخ احتياطي وهذه هي النقطة المهمة

```
{
  "Effect": "Allow",
  "Action": [
    "dynamodb:CreateBackup"
  ],
  "Resource": [
      "arn:aws:dynamodb:*:*:*"
  ]
}
```

الملف الثاني index.js يحتوي على الإجراء بلغة Node.js حيث يقوم بعمل العميات التالية:

قراءة قائمة باسماء الجداول التي تحتاج إلى نسخ احتياطي كامل

```
exports.handler = (event, context, callback) => {
    console.log('Received event', event);
    var tablesToBackup = event.tablesToBackup.split(",");
```

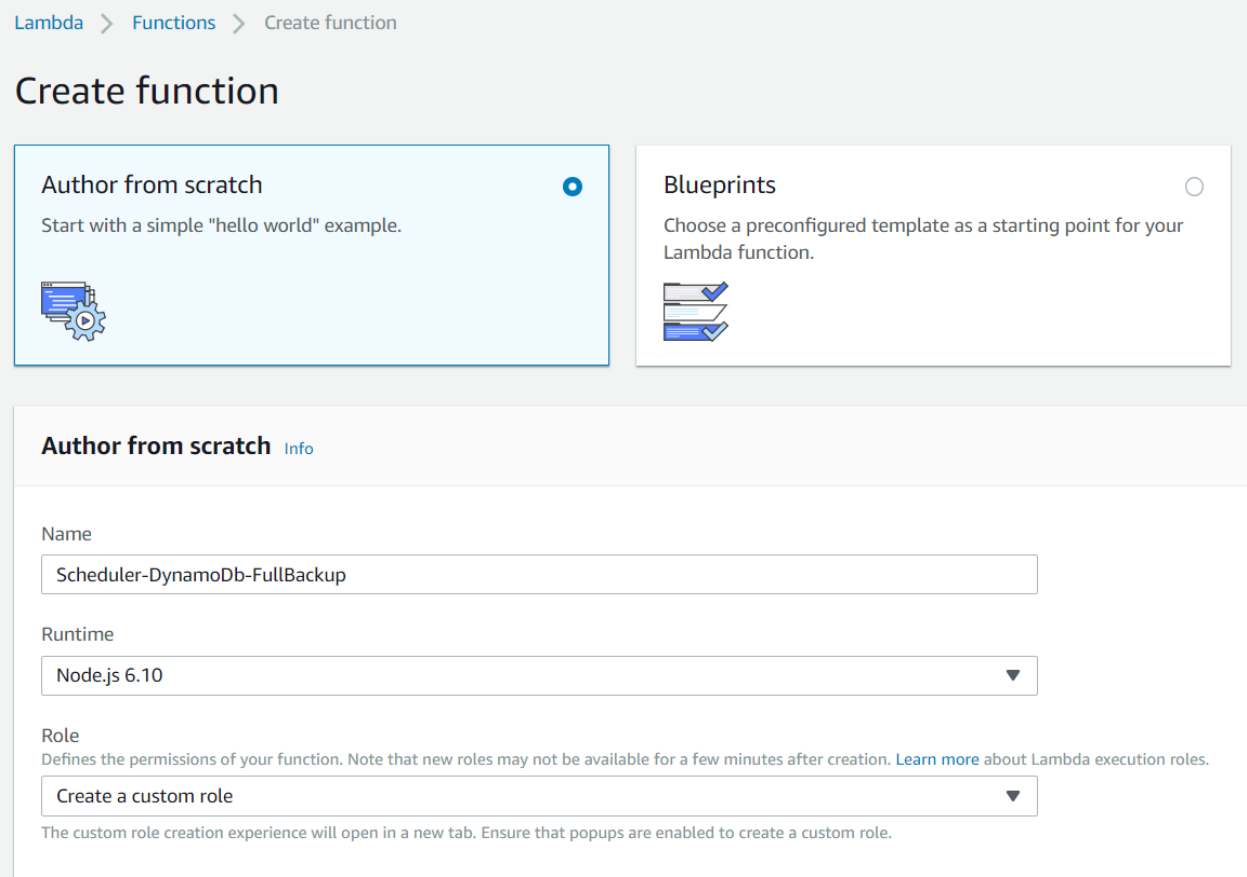
ثم استداعاء شيفرة المصدر الخاصة بعملية نسخ الاحتياطي

```
 var promises = tablesToBackup.map(backupTable);
```

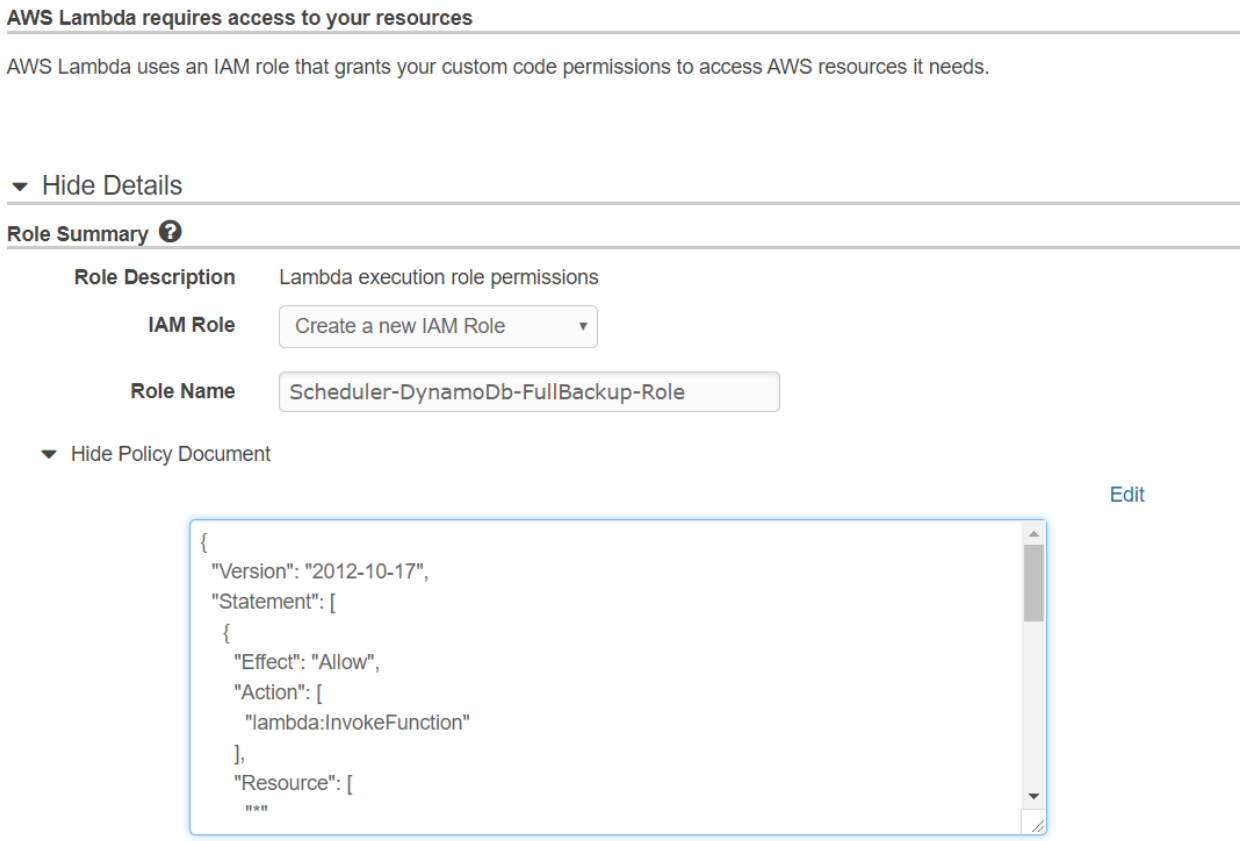
حيث يقوم هذا الجزء بانشاء اسم للنسخة الاحتياطية تحتوي على اسم الجدول + التاريخ التي تمت بها العملية ثم يقوم بعمل النسخ الاحتياطي

```
function backupTable(tablename) {
  var timestamp = new Date().toISOString()
    .replace(/\..+/, '')
    .replace(/:/g, '')
    .replace(/-/g, '');

  var params = {
    TableName: tablename,
    BackupName: tablename + timestamp
  };
  console.log(params);
  return dynamodb.createBackup(params).promise();
}
```

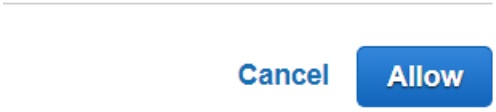
ثانيا قم بفتح لوحة تحكم خدمات أمازون والانتقال إلى انشاء إجراء lambda كما في الصورة



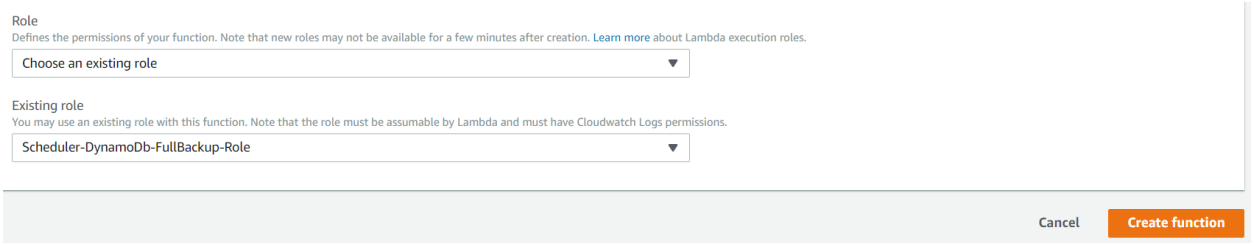
قم بنقل المحتوى الموجود في الملف ondemand-backups-scheduler-policy.json كما في الصورة



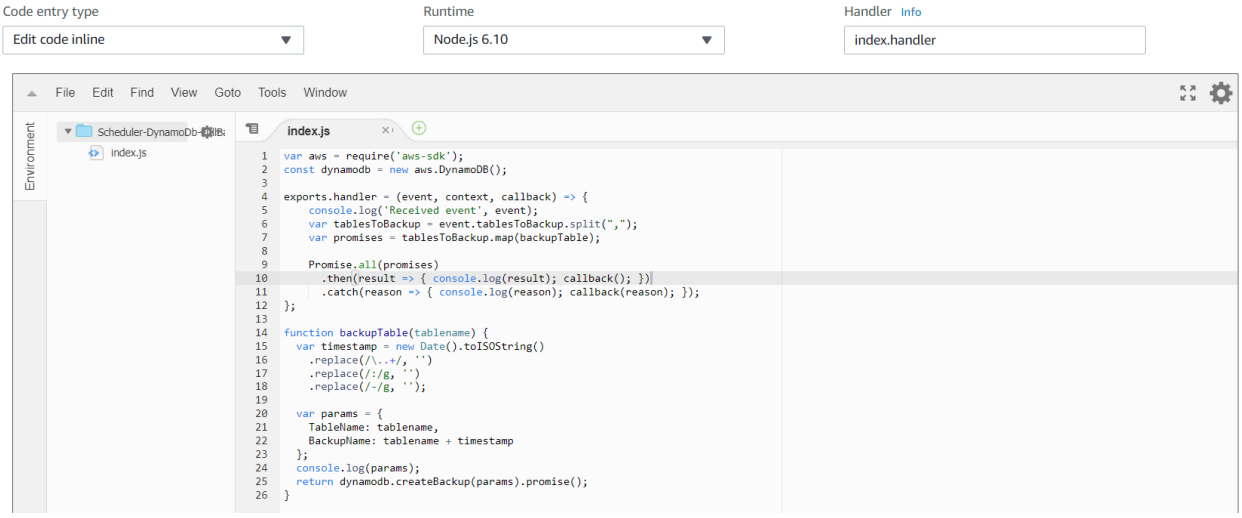
ثم اضغط على زر Allow



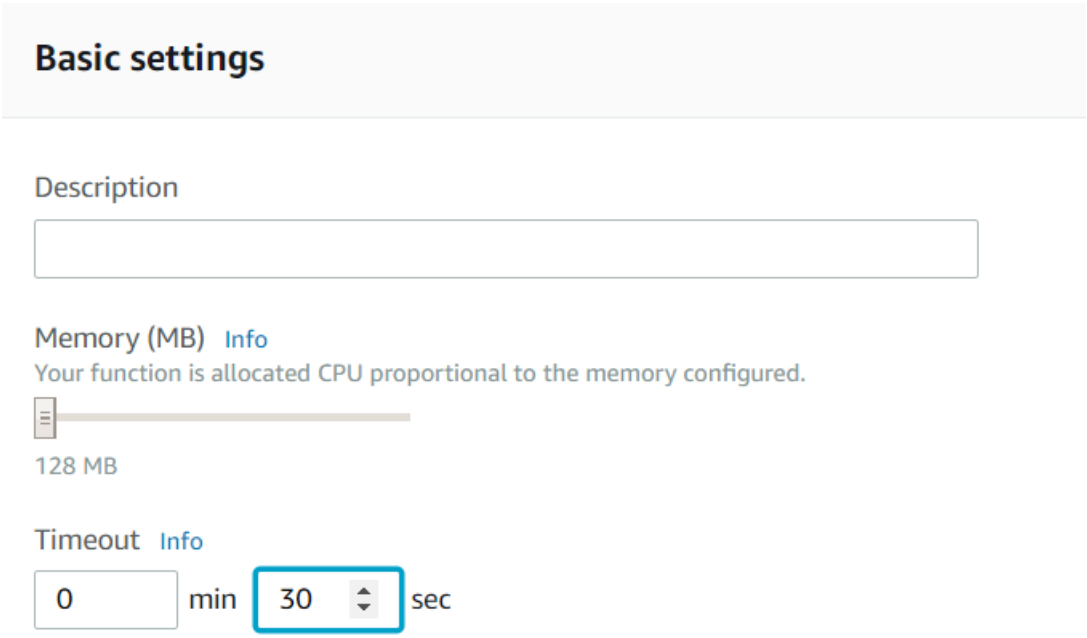
ثم اضغط على زر انشاء إجراء Create function



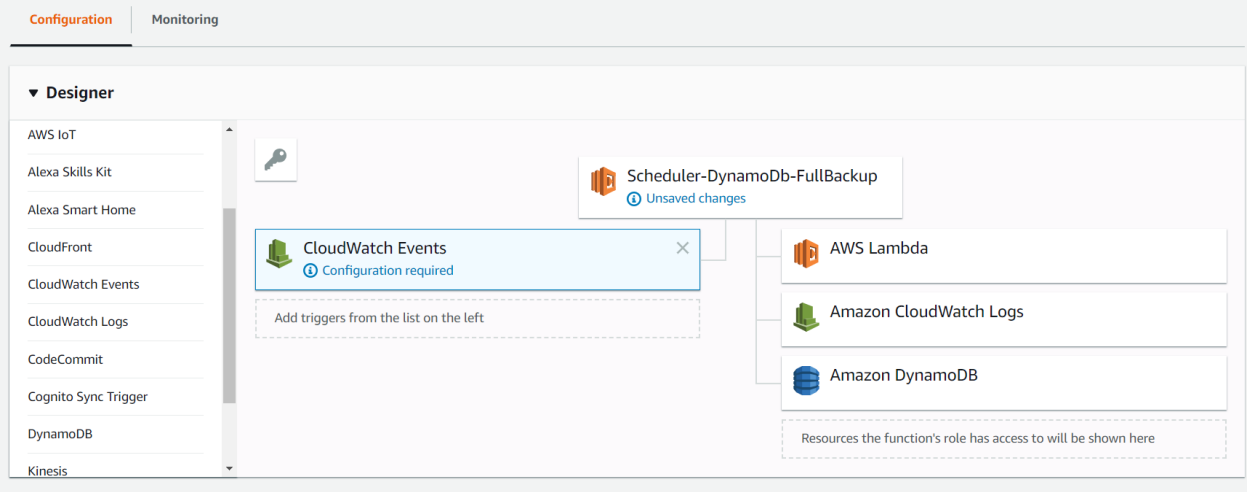
انسخ شيفرة المصدر الموجودة في الملف index.js وضعه كما في الصورة



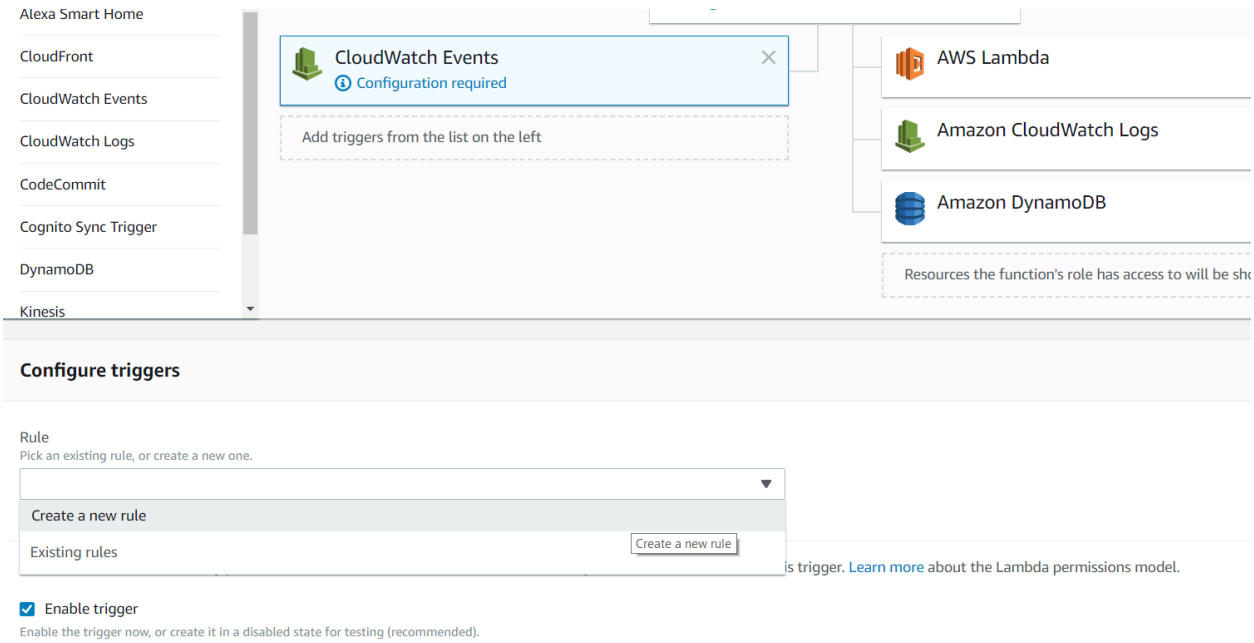
قم بتعديل مدة تنفيذ الإجراء بزيادته إلى 30 sec



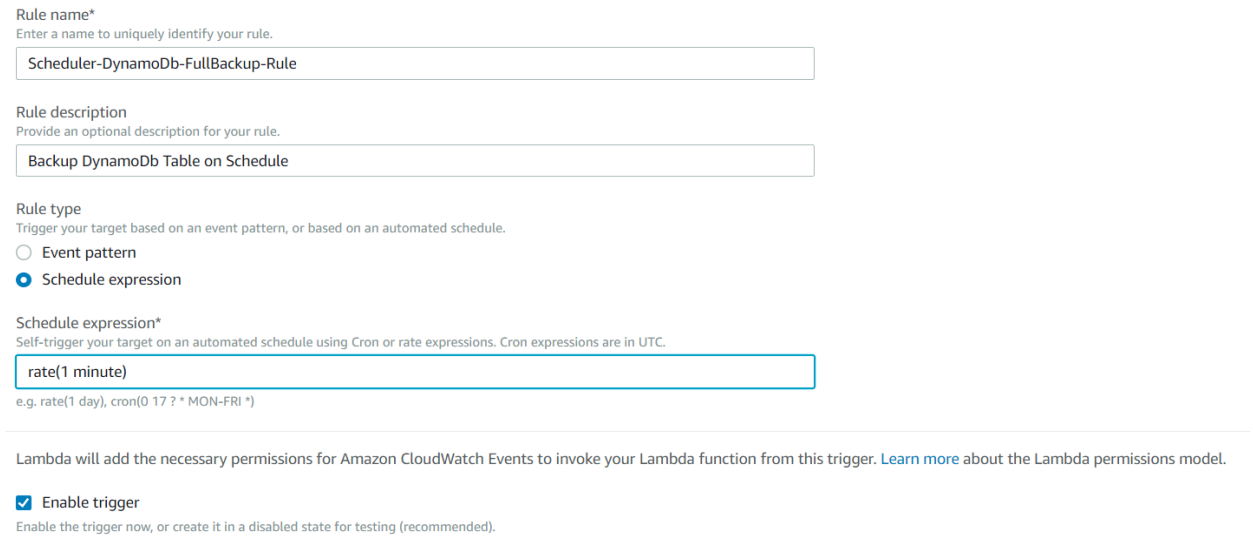
ثم قم بإضافة عنصر Cloudwatch event كما في الصورة



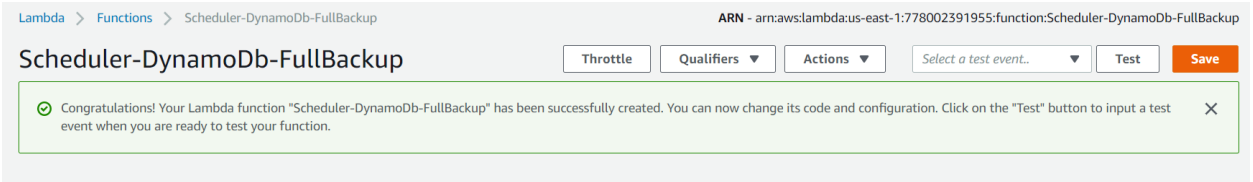
انشئ قاعدة جديدة Rule كما في الصورة لتفعيل التوقيت الزمني لاستدعاء الإجراء



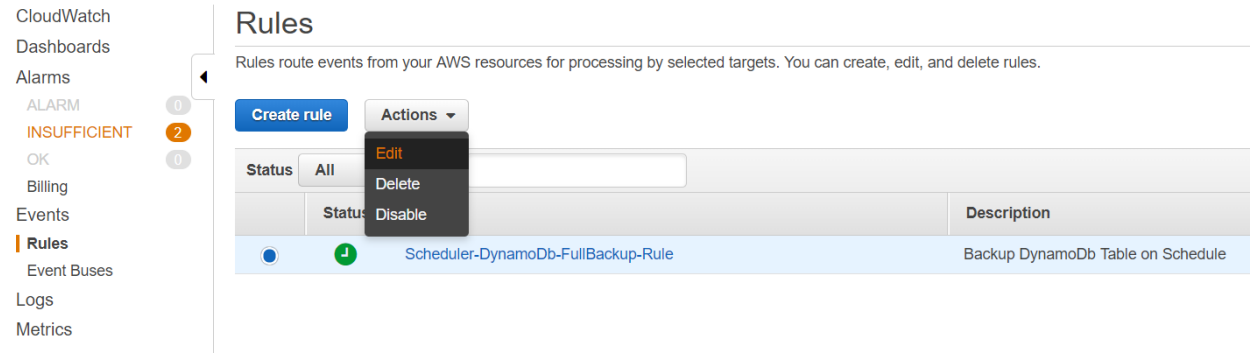
قم بتعبئة الحقول كما في الصورة



ثم اضغط على زر حفظ



قم بالإنتقال إلى لوحة تحكم Cloudwatch وتعديل القاعدة Rule التي تم انشاؤها مسبقا من أجل ادخال اسماء الجداول كما في الصورة



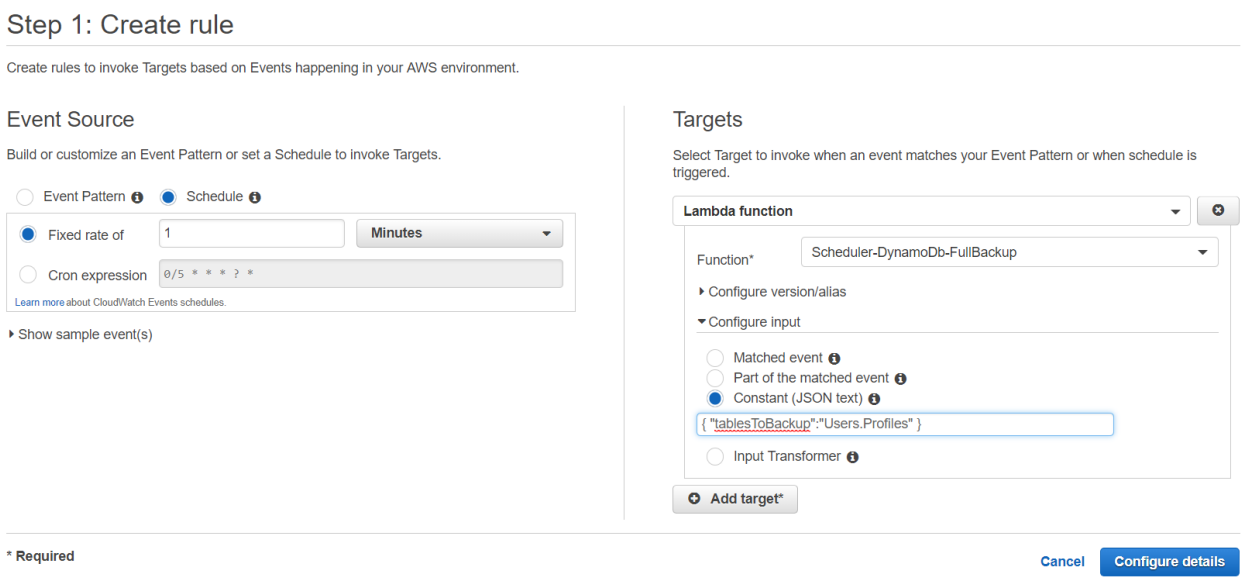
قم بإدخال اسماء الجداول كما في الصورة مثلا

{ "tablesToBackup":"Users.Profiles" }

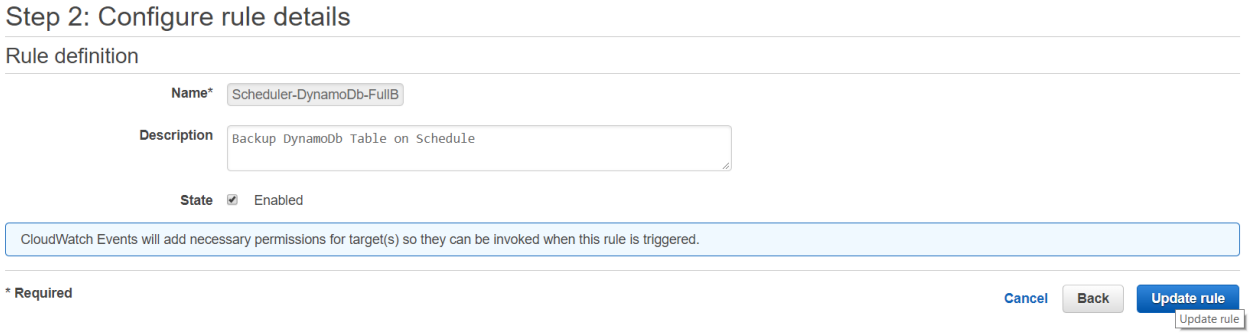
أو في حالة أكثر من جدول

{ "tablesToBackup":"Users.Profiles, Users.Profiles2,... " }

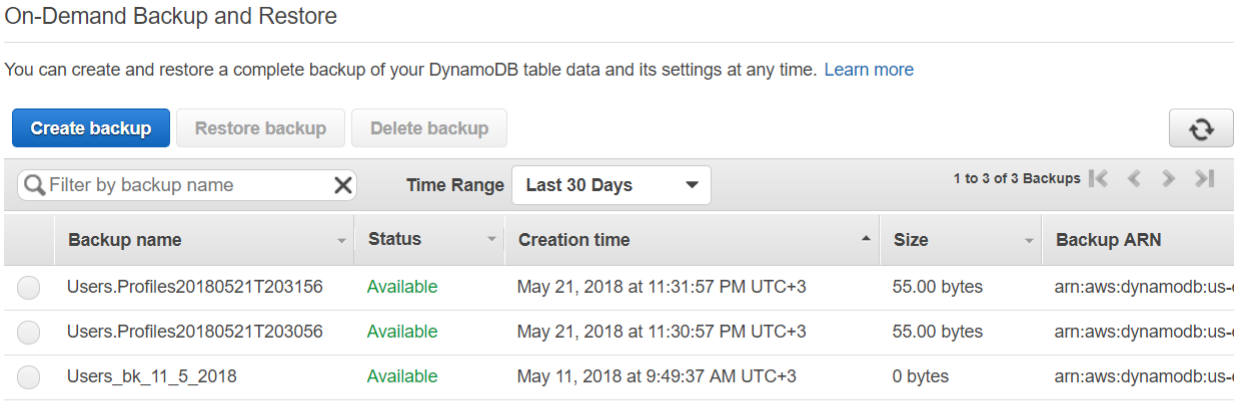
وتحديد التوقيت الزمني لتشغيل الإجراء



ثم قم بالضغط على زر تحديث القاعدة Update rule



لو انتقلنا إلى لوحة التحكم الخاصة بقاعدة بيانات DynamoDb في جزء النسخ الاحتياطي, نلاحظ انشاء نسخة احتياطية اسم الجدول + تاريخ العميلة كما في الصورة



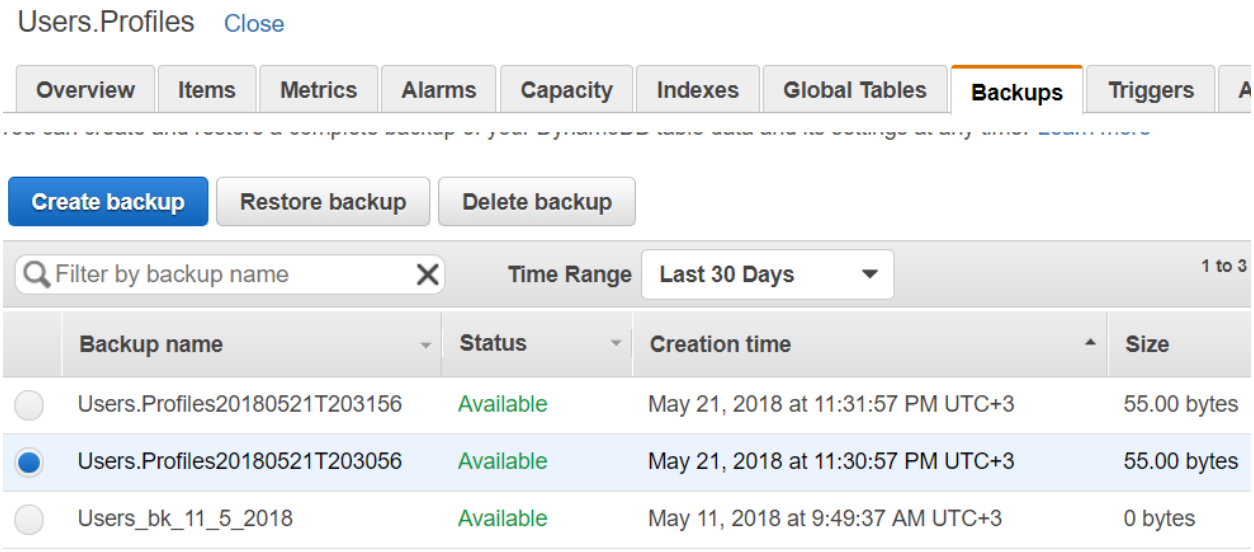
ملاحظة

قبل تفعيل خاصية النسخ الاحتياطي ضمن خدمات قاعدة بيانات DynamoDb, كانت عملية اخذ نسخ احتياطي كامل يتم عن طريق استخدام خدمة EMR مع خدمة S3 وهذا يتطلب العديد من الإعدادات والعمليات مع إشكالية زيادة التكلفة بسب إستخدام خدمات أخرى و عمل قراءة لجميع البيانات في الجدول Scan في كل مرة يتم أخذ نسخ احتياطية كاملة.

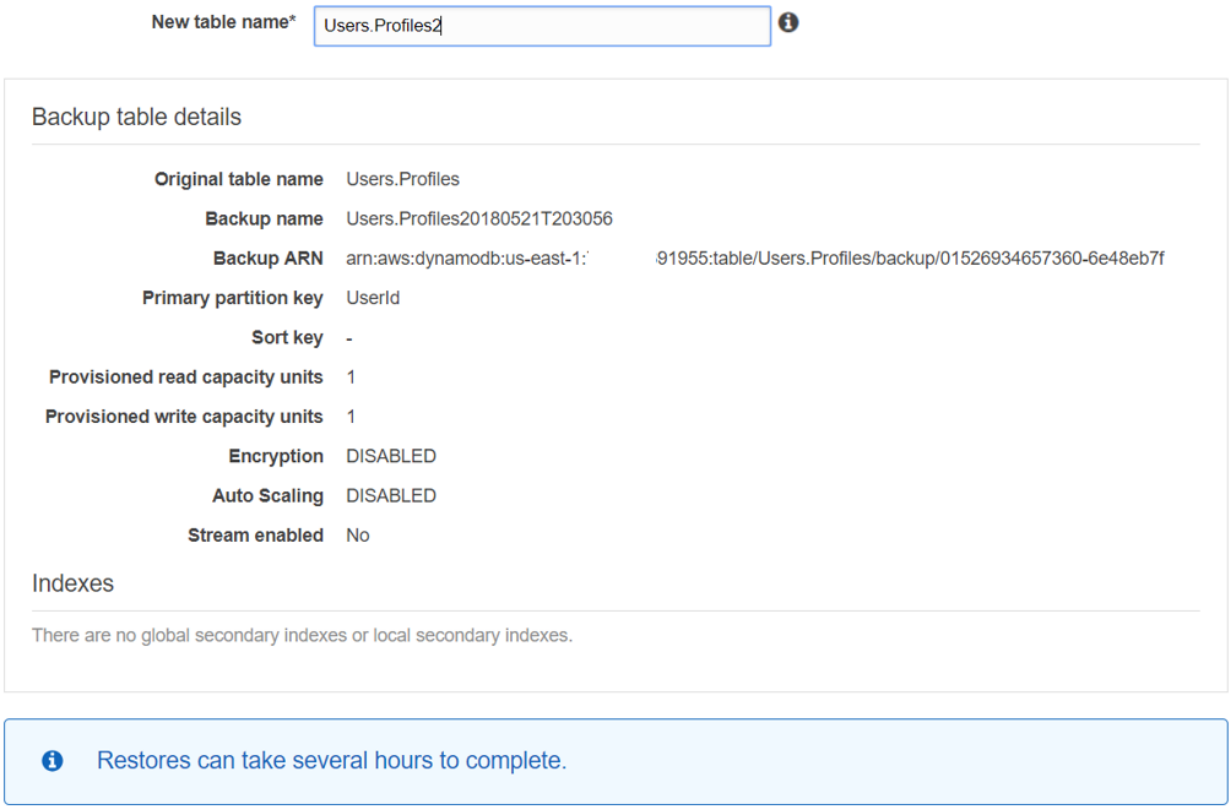
استرجاع نسخ احتياطية كاملة لقاعدة بيانات DynamoDb

لننتقل إلى كيفية استرجاع نسخة احتياطية كاملة لقاعدة بيانات DynamoDb, لعمل ذلك يجب مراعاة أن عملية الاسترجاع تقوم بانشاء جدول جديد من النسخة الاحتياطية مع الحفاظ على بعض الخصائص التي تم حفظها عند اخذ نسخة احيتاطية كما ذكرنا سابقا, ويكمن عمل ذلك باتباع الخطوات التالية:

قم بإختيار الجدول المراد استرجاع النسخة الاحتياطية الخاصة به ثم اضغط على زر استرجاع



قم بإدخال اسم الجدول الجديد المراد انشاؤه

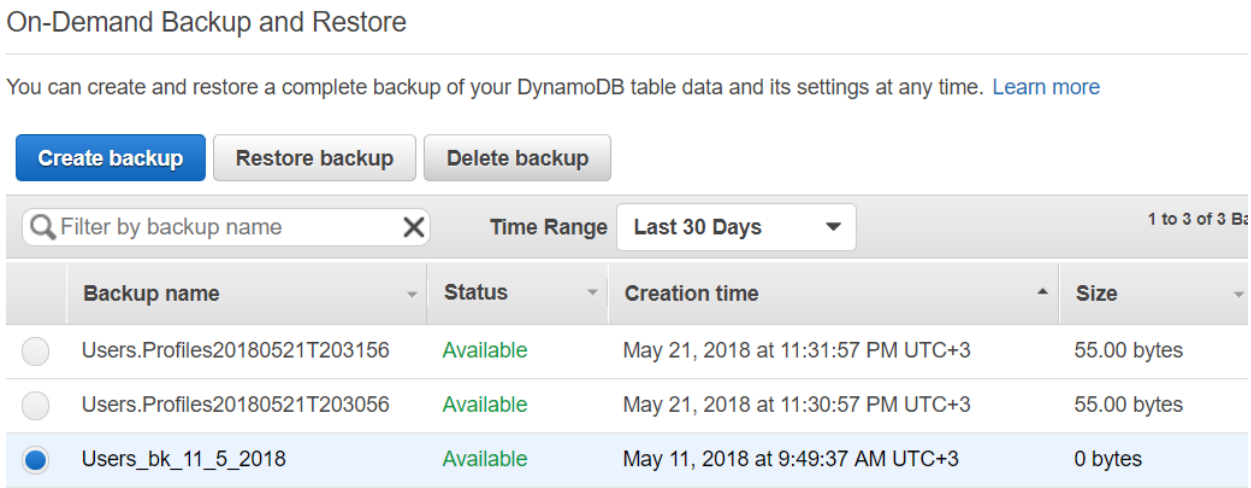


ثم اضغط على زر استرجاع, لاحظ أن هذه العملية قد تأخذ وقت لإتمامها, انتظر وراقب.

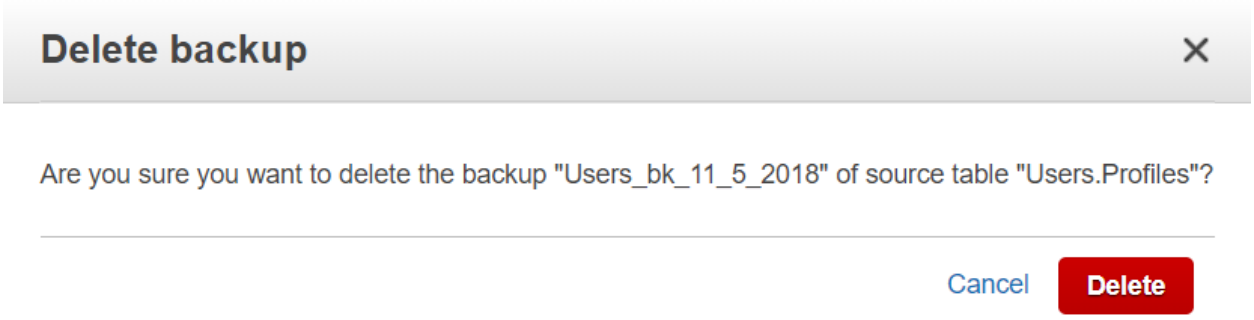
يمكن جدولة هذه العملية بنفس الطريقة التي تمت في جدولة عملية انشاء نسخ احيتاطية كاملة باستخدام الإجراء lambda مع الحدث في خدمة Cloudwatch event.

حذف النسخ الإحتياطية لقاعدة بيانات DynamoDb

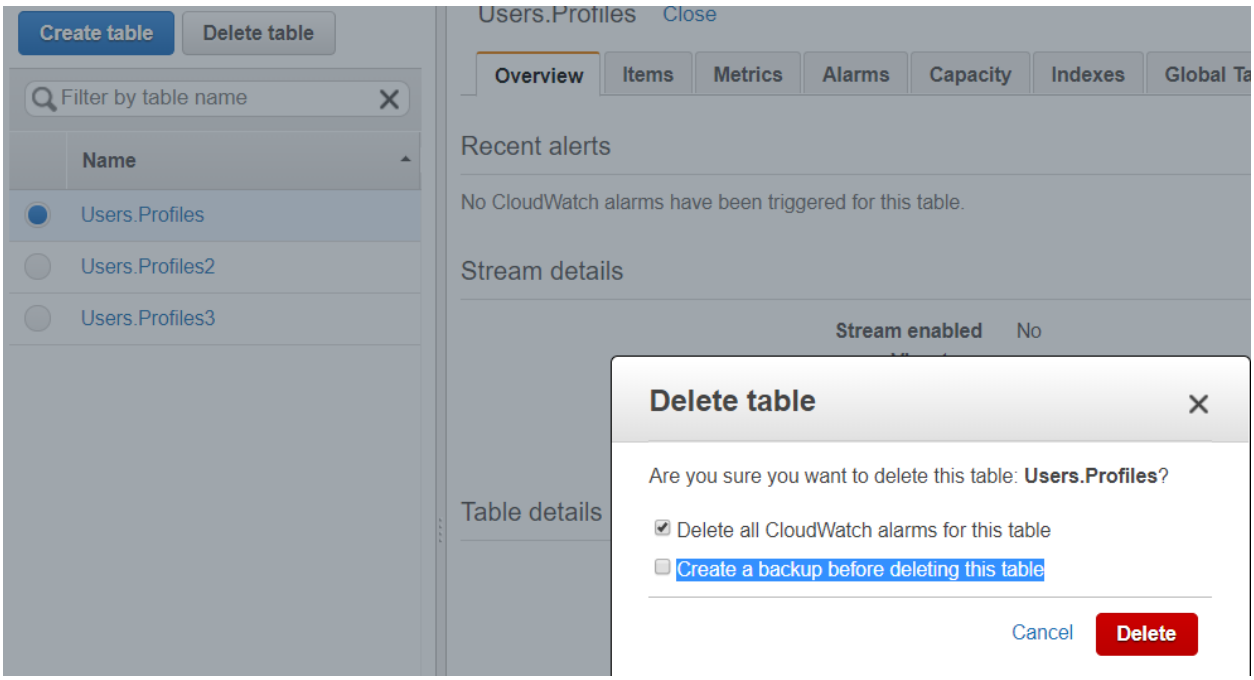
أيضا من العمليات التي يكمن عملها في لوحة تحكم قاعدة بيانات DynamoDb هي حذف النسخ الاحتياطية من لوحة تحكم قاعدة بيانات DynamoDb لتقليل مساحة التخزين المحجوزة والتي يتم احتسابها في فاتورة الشهر, لعمل ذلك قم باختيار النسخة المراد حذفها ثم اضغط على زر حذف.



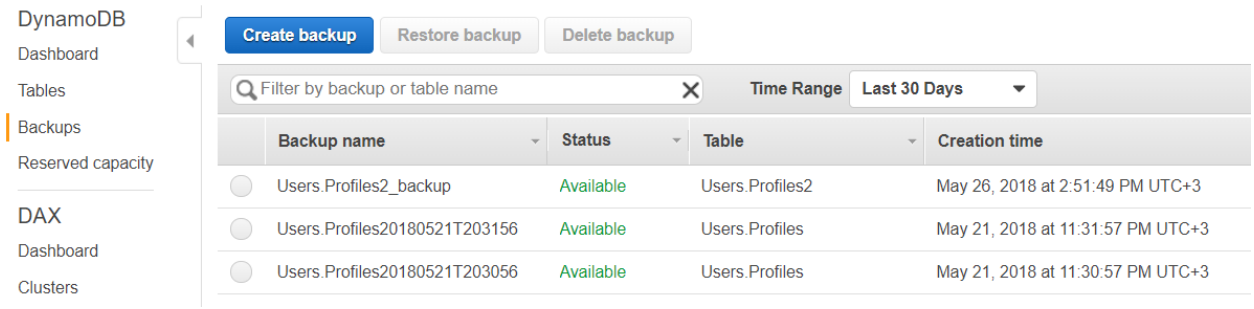
اضغط على زر حذف



أيضا عند حذف جدول في قاعدة بيانات DynamoDb, تقوم أمازون بسؤالك إذا كنت ترغب بأخذ نسخة احتياطية من الجدول المراد حذفه كما في الصورة



يمكن الوصول لهذه النسخة الاحتياطية عند الضغط على زر النسخ الاحتياطية الموجود في الجهة اليسرى في لوحة تحكم قاعدة بيانات DynamoDb كما في الصورة



عمل نسخة احتياطية بشكل تزايدي لقاعدة بيانات DynamoDb

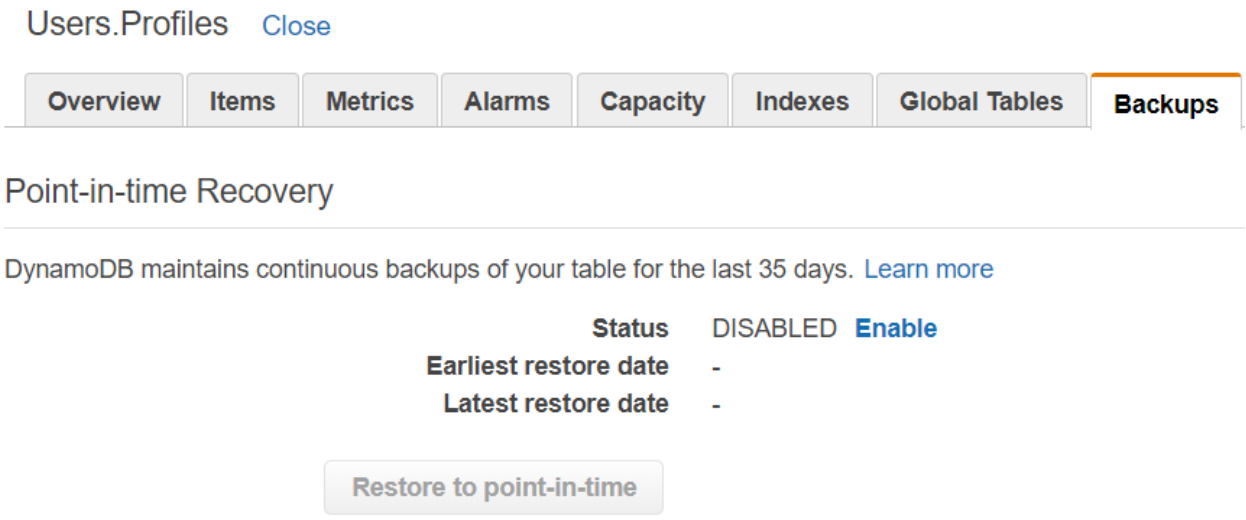
ماذا عن اخذ نسخ احتياطية بشكل تزايدي Incremental backup, لحسن الحظ تم عمل هذه الميزة من وقت قريب حيث كان في السابق لعمل ذلك نحتاج إلى عمل الكثير من الإعدادات الخاصة بقاعدة بيانات DynamoDb مثل Triggers و خدمة S3 لتخزين النسخ مع استخدام اجراء lamda مع الحدث Cloudwatch event ومع كل هذه الإعدادات كان الحل غير مكتمل وفيه العديد من المشاكل خاصة في طريقة استرجاع البيانات من خدمة S3.

في لوحة تحكم قاعدة بيانات DynamoDb نجد الجزء العلوي في قسم النسخ الاحتياطي حيث تقوم شركة أمازون بأخذ نسخ احتياطية بشكل تلقائي مع إمكانية استرجاع البيانات حتى مدة 35 يوم.

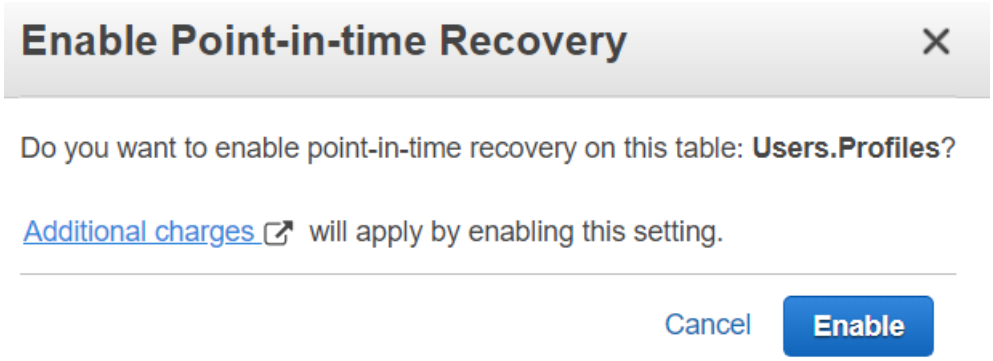
أيضا في هذا الحالة لن تحتاج إلى إدارة إنشاء النسخ الاحتياطية واسترجاعها ولن تحتاج أيضا إلى علمية الجدولة كما فعلنا مع النسخ الاحتياطي الكامل.

وكما عودتنا أمازون في خدماتها لن تؤثر هذا العمليات على إداء قاعدة بيانات DynamoDb حيث تعتمد على النسخ الأخرى كما تم شرحه مسبقا.

لاستخدام هذه الخدمة, يجب أولا تفعيلها من لوحة تحكم قاعدة بيانات DynamoDb, قم بالضغط على زر تفعيل كما في الصورة



عند تفعيل هذه الخدمة ستقوم أمازون بحساب مساحات التخزين الخاصة بالجدول و local index ضمن فاتورة الشهر لهذه الخدمة

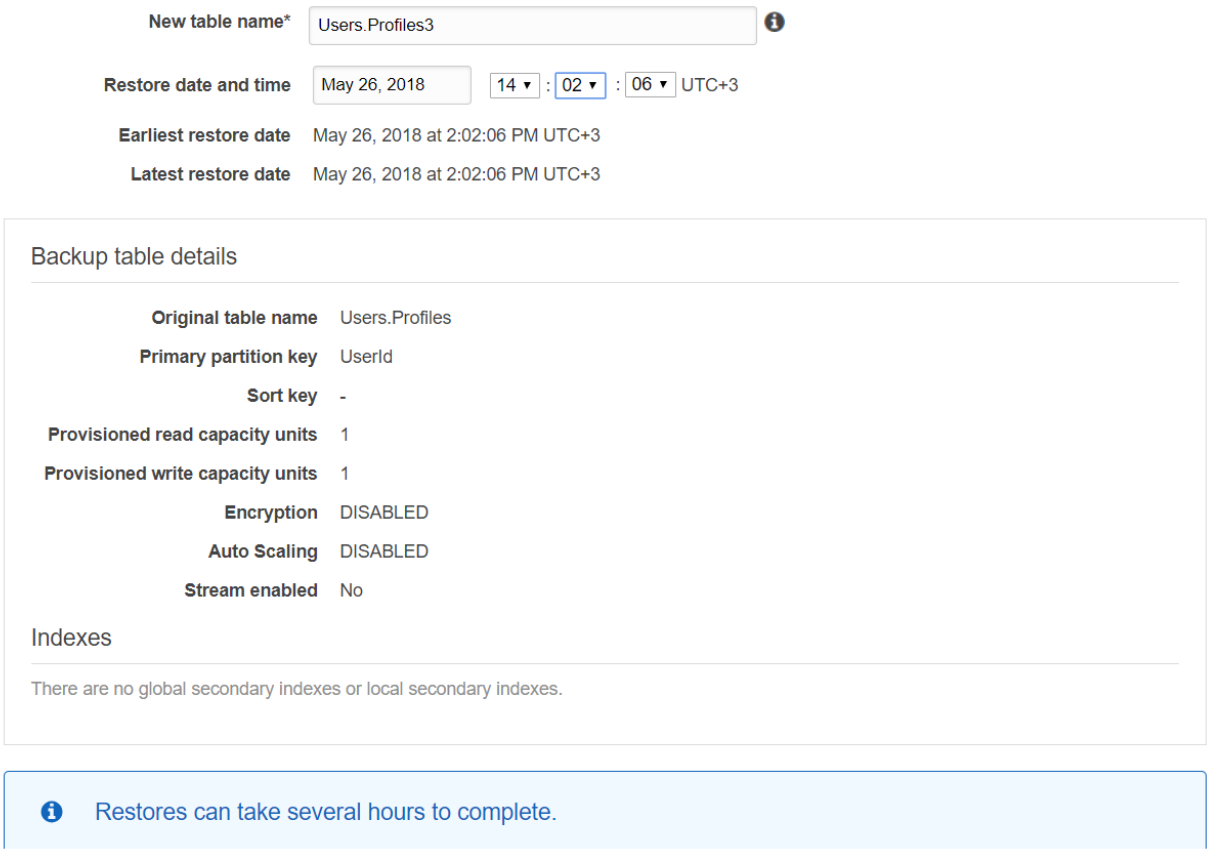


يمكنك استرجاع البيانات في أي وقت خلال 35 يوم من بداية تفعيل النسخ الإحتياطية, لاسترجاع بيانات في وقت معين قم بالضغط على زر استرجاع

ملاحظة

عند ايقاف تفعيل هذه الخاصية ستخسر النسخ الاحتياطية التي تم اخذها عن طريق هذه الخاصية ولن تستطيع استعادتهم مرة اخرى.

قم بادخال اسم الجدول الجديد المراد انشاؤه واختيار التاريخ المراد استرجاع البيانات المخزنة فيه في تلك اللحظة



ثم اضغط على زر استرجاع, لاحظ أن هذه العملية قد تأخذ وقت لاتمامها, انتظر وراقب.

ففي هذه الحالة نقطة استعادة البيانات Recovery Point Objective - RPO سيكون منخفض و الوقت الازم للإستعادة Recovery Time Objective – RTO سيكون على حسب الوقت المطلوب لانشاء الجدول حيث غالبا تكون عملية استرجاع البيانات و تشغيلها بشكل سريع.

إدراة أمن المعلومات في قاعدة بيانات DynamoDb

كما ذكرنا سابقا أن قاعدة بيانات DynamoDb تندرج تحت الخدمات التي تسمى Fully managed أي أن الكثير من المهام المتعلقة بإدارة العتاد وقاعدة البيانات تقع تحت مسؤولية أمازون ومن هذه النقاط أمن المعلومات حيث تتحمل أمازون مسؤولية حماية توقف أو ضياع البيانات المخزنة في قاعدة بيانات DynamoDb والتأكد من إدارتها بشكل يحافظ على السرية وأفضل معايير الأمان.

لكن يوجد بعض المهام المتعلقة بأمن المعلومات تندرج تحت مسؤولية العميل أو مستخدم قاعدة بيانات DynamoDb ويمكن حصر هذه المهام ضمن النقاط التالية:

- إدارة الصلاحيات والوصول للبيانات الموجودة في قاعدة بيانات DynamoDb
- تشفير البيانات المخزنة في قاعدة بيانات DynamoDb

إدارة الصلاحيات والوصول للبيانات الموجودة في قاعدة بيانات DynamoDb

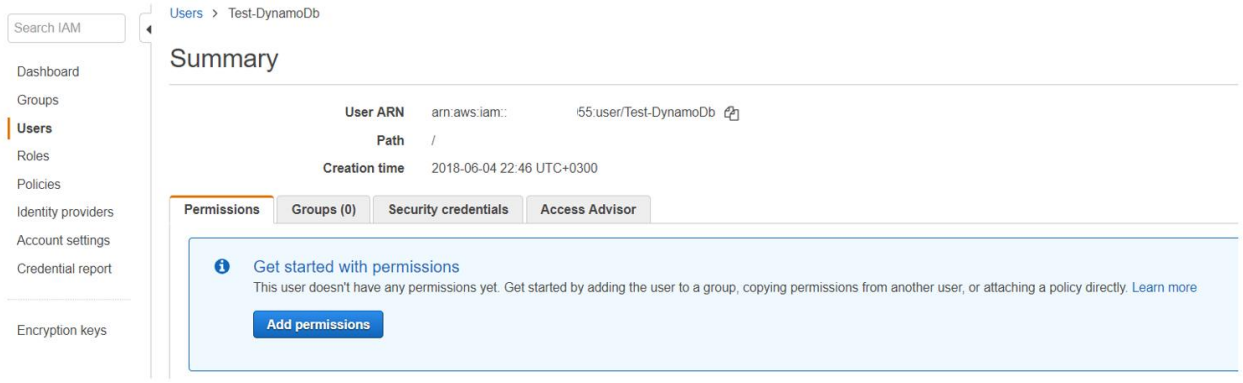
يتم إدارة الصلاحيات والوصول للبيانات الموجودة في قاعدة بيانات DynamoDb عن طريقة خدمة إدارة الهوية والوصول IAM والتي تعتبر خدمة موحدة تدير جميع الأمور التي تتعلق بأمن المعلومات لخدمات أمازون وبالأخص الملف المسؤول عن تعريف سياسة الوصول Policy حيث يمكن تحديد الصلاحيات ضمن مستويات مختلفة في هذا الملف وهي كتالي:

- التحكم بالوصول على مستوى الجدول Table
- التحكم بالوصول على حسب المفتاح الفهرسي Partition Key
- التحكم بالوصول على مستوى واجهات التطبيق البرمجية API
- التحكم بالوصول على حسب الحقول Attribute

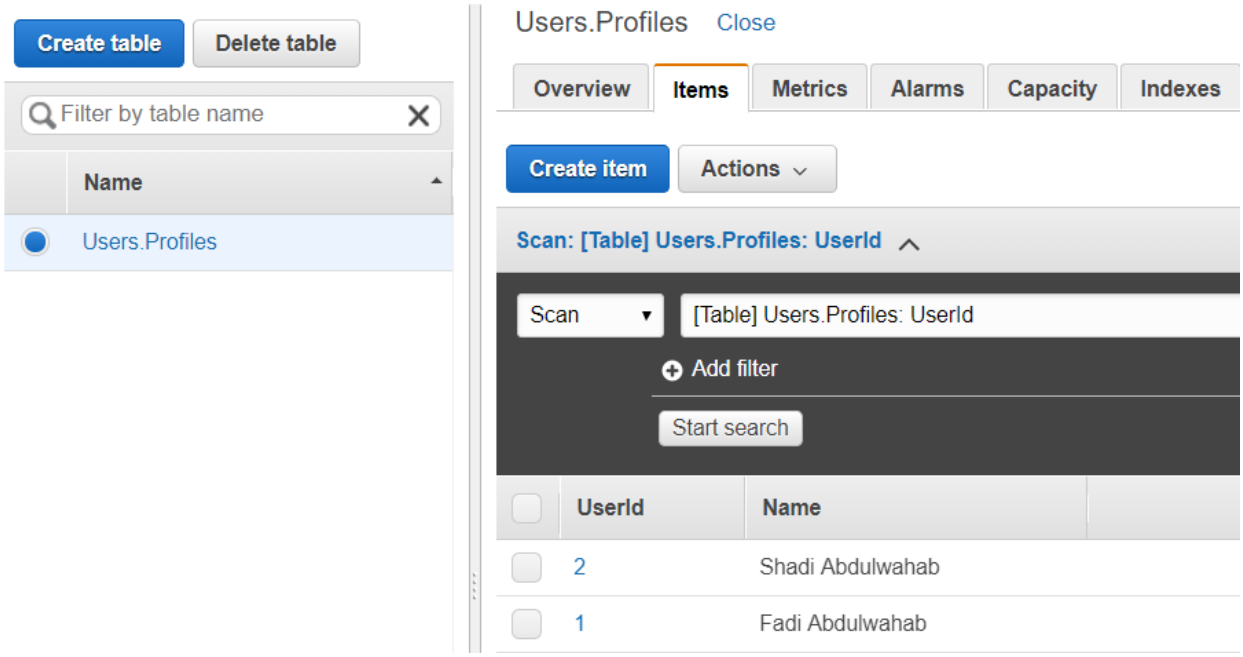
التحكم بالوصول على مستوى الجدول Table ومستوى واجهات التطبيق البرمجية API

لنبدأ مع التحكم بالوصول على مستوى الجدول حيث يمكن التحكم بمنح أو منع المستخدمين من الوصول إلى جدول أو مجموعة الجداول الموجودة في قاعدة بيانات DynamoDb.

لتوضيح هذه الطريقة لنفترض أنه لدينا مستخدم في خدمة إدارة الهوية والوصول IAM بدون أي سماحيات

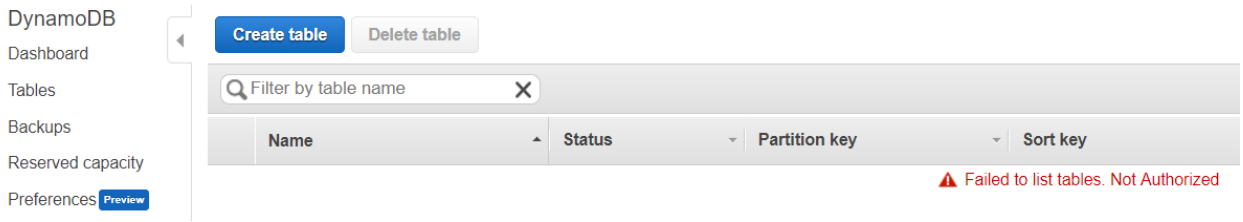


ويريد الوصول إلى الجدول التالي



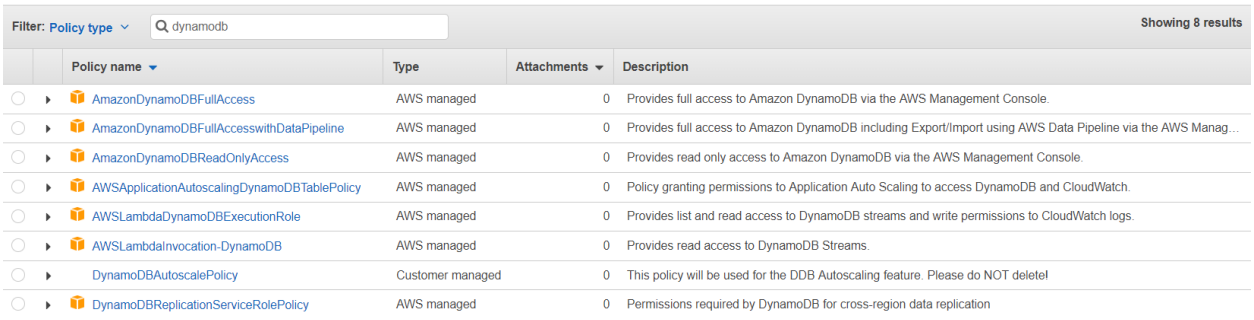
حيث UserId هو المفتاح الرئيسي

بهذه الحالة لو حاول المستخدم الوصول إلى قاعدة بيانات DynamoDb لن يتمكن من الوصول للبيانات لأنه ليس لديه أي صلاحيات تمكنه من قراءة البيانات وهذا جيد وهو يعتبر من أساسيات أمن المعلومات حيث الوضع الإفتراضي هو عدم تمكين أي أحد من الوصول لهذه الخدمات أو البيانات إلا بعد السماح له بذلك.

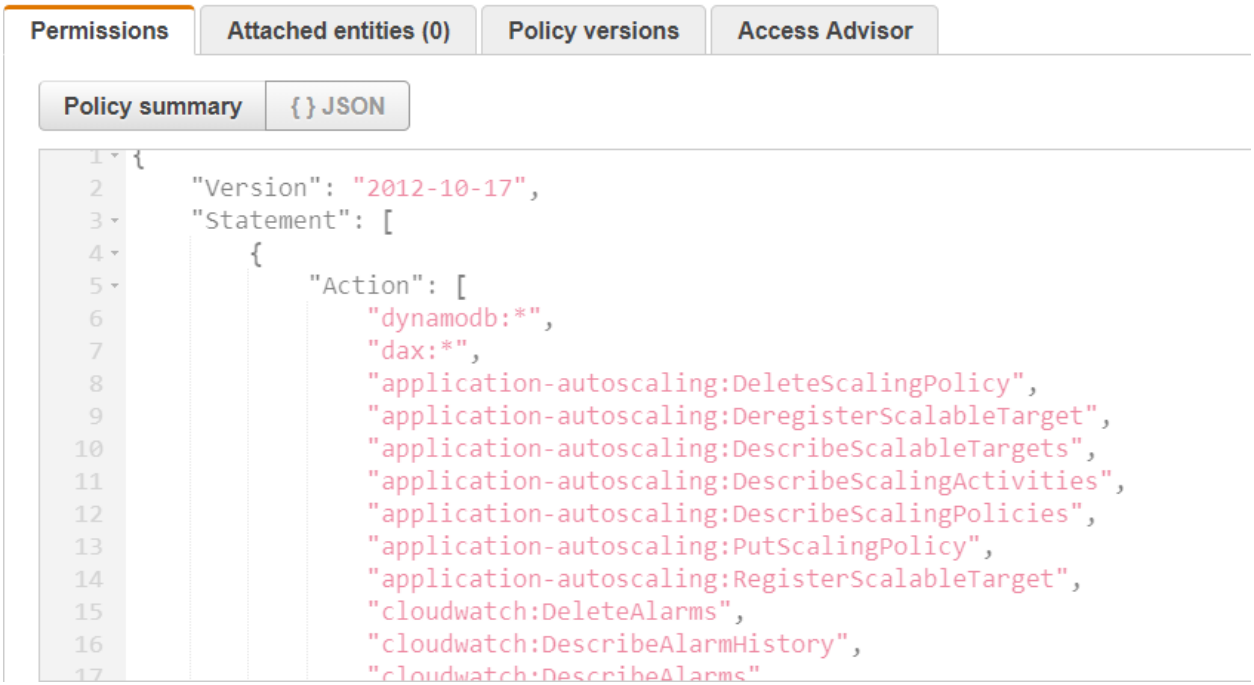


بشكل افتراضي تحتوي خدمة إدارة الهوية والوصول IAM على مجموعة من الصلاحيات المعرفة مسبقا والتي تساعد على تمكين المستخدمين من استخدام قاعدة بيانات DynamoDb ومن هذه المجموعات مثلا

AmazonDynamoDBFullAccess	حيث يمكن التحكم الكامل على قاعدة بيانات DynamoDb وعمل جميع العمليات الممكنة على جميع الجداول
AmazonDynamoDBReadOnlyAccess	حيث يمكن عمل جميع العمليات التي تقوم فقط بالقراءة ولا يسمح بعمل أي عملية تتعلق بالكتابة أو الحذف على جميع الجداول

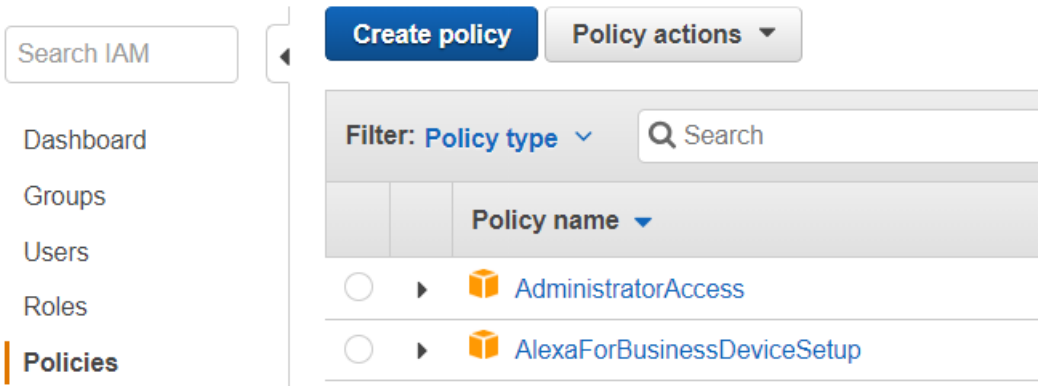


ننصح بالاطلاع على ملف سياسة الوصول Policy الخاص بكل مجموعة للتعرف على العمليات المسموح بها كما في الصورة



لنفترض مثلا لو أردنا إنشاء ملف سياسة وصول محدد Policy بحيث نسمح للمستخدم بالوصول إلى جداول معينة بدلا من جميع الجدول كما هوموجود في ملف سياسة الوصول AmazonDynamoDBFullAccess.

لعمل ذلك قم بانشاء ملف سياسة وصول Policy من شاشة انشاء ملف سياسات الوصول الموجود في خدمة إدارة الهوية والوصول IAM



نلاحظ في ملف سياسة الوصول تم منح الوصول لجدول واحد فقط تحت جزء Resource وبعض العمليات التي يمكن إجراؤها عن طريق واجهات التطبيق البرمجية API أو أوامر CLI مثل القراءة والكتابة على الجدول كما في الصورة

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:DescribeTable",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem",
                "dynamodb:GetRecords",
                "dynamodb:ListTables"
            ],
            "Resource": "arn:aws:dynamodb:us-east-1:*:table/Users.Profiles"
        }
    ]
}
```

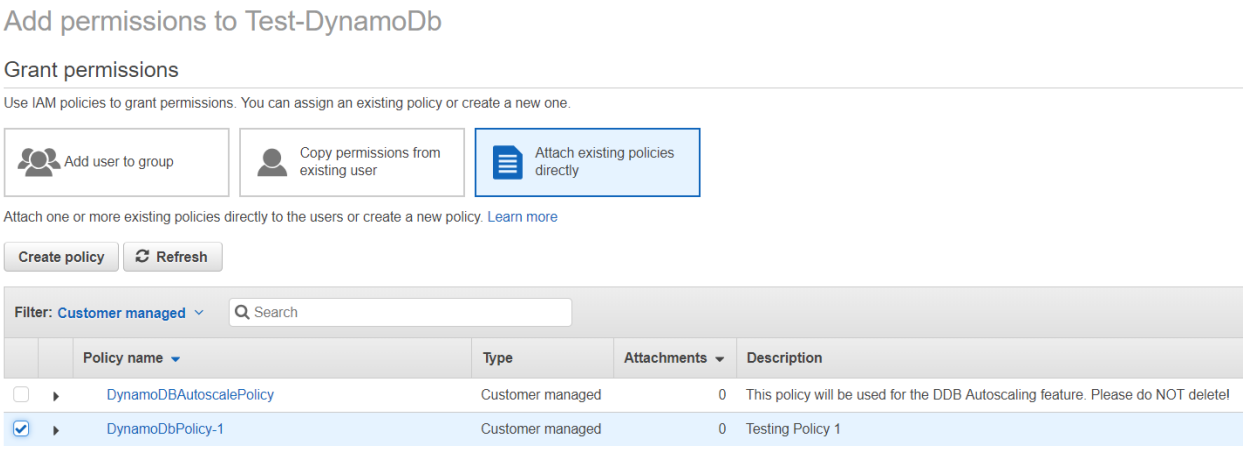
ملاحظة

عند التعامل مع جدول قاعدة بيانات DynamoDb عن طريق الشيفرة أو أوامر التنفيذ لن تحتاج إلى منح المستخدم جميع هذه الصلاحيات مثل DescribeTable و ListTables والتي تساعد على عرض الجداول في لوحة تحكم قاعدة بيانات DynamoDb.

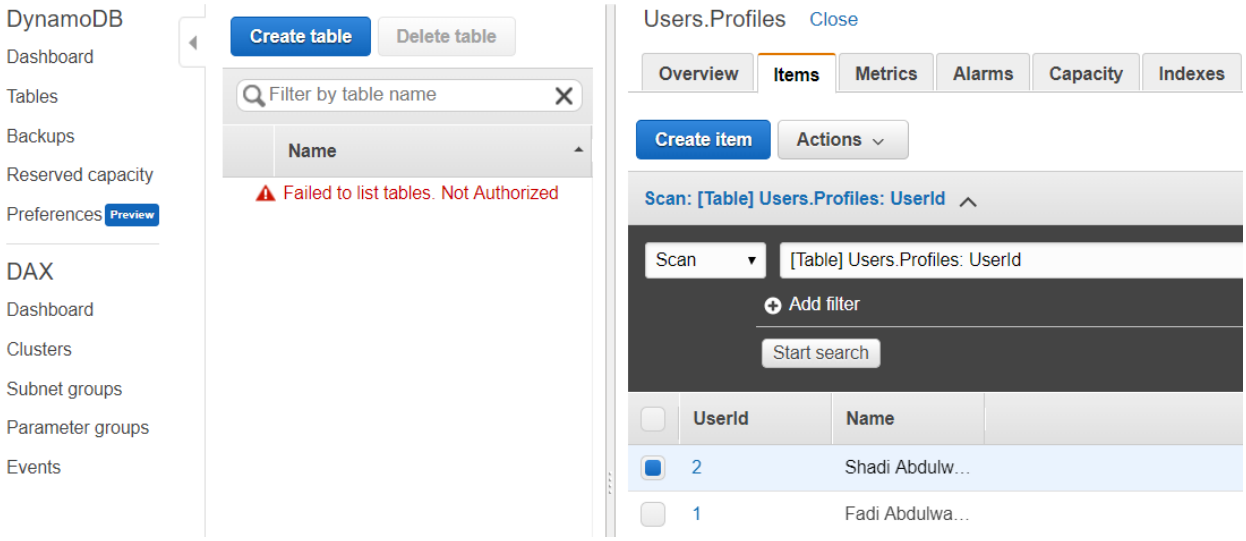
ملاحظة: لمنح المستخدم إمكانية الوصول إلى جميع الجداول يمكن استبدال السطر الذي في الأعلى ب * : Resource حيث يمكن تعريف هذا السطر بالشكل التالي:

```
"Resource": ["arn:aws:dynamodb:<REGION>:<ACCOUNTNUMBER>:table/<TABLE-NAME>"
```

فلو تم منح المستخدم هذه الصلاحيات كما في الصورة



ففي هذه الحالة سوف يتمكن من عرض الجدول والقيام ببعض العمليات مثل القراءة والكتابة



التحكم بالوصول على حسب الحقول Attribute

أيضا يمكن تقليل هذه السماحيات بالتحكم بالوصول لبيانات بجدول معين بمنح أو منع المستخدمين من الوصول إلى بعض الحقول وليس جميعها بحيث يمكن إخفاء بعض الحقول الحساسة عن بعض المستخدمين.

لتحديد الحقول التي يجب فقط الوصول إليها يمكن تعريف سياسات الوصول Policy كما في الصورة

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:GetItem",
                "dynamodb:BatchGetItem",
                "dynamodb:Query",
                "dynamodb:PutItem",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteItem",
                "dynamodb:BatchWriteItem"
            ],
            "Resource": [
                "arn:aws:dynamodb:<REGION>:<ACCOUNTNUMBER>:table/<TABLE-NAME>"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "dynamodb:Attributes": [
                        "<COLUMN-NAME-1>",
                        "<COLUMN-NAME-2>",
                        "<COLUMN-NAME-3>"
                    ]
                },
                "StringEqualsIfExists": {
                    "dynamodb:Select": "SPECIFIC_ATTRIBUTES"
                }
            }
        }
    ]
}
```

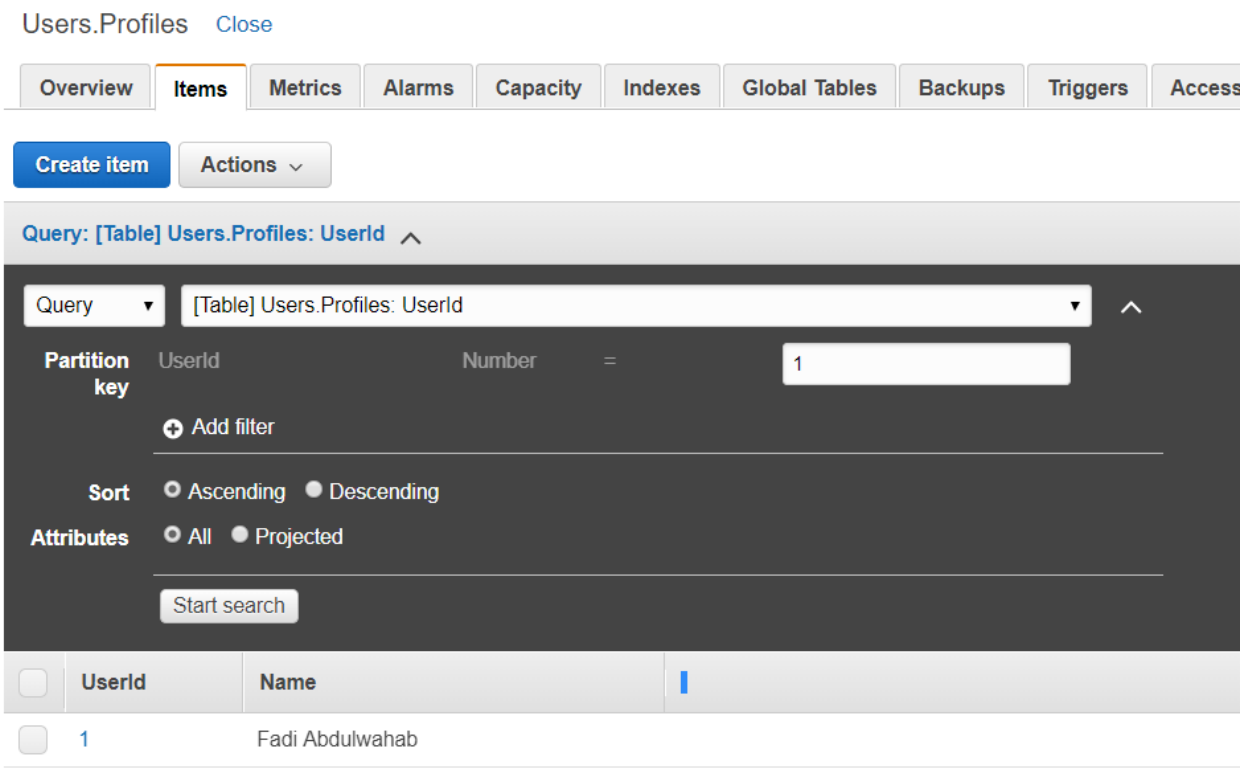
التحكم بالوصول على حسب المفتاح الرئيسي Partition Key

من الطرق أيضا التحكم بالوصول لبيانات ادول معين بمنح أو منع المستخدمين من الوصول إلى البيانات باستخدام المفتاح الرئيسي في جدول معين. فمثلا نقوم بتخزين اسم المستخدم أو البريد الإلكتروني كمفتاح رئيسي والسماح للمستخدم بالوصول إلى البيانات التابعة لهذا المفتاح الرئيسي فقط.

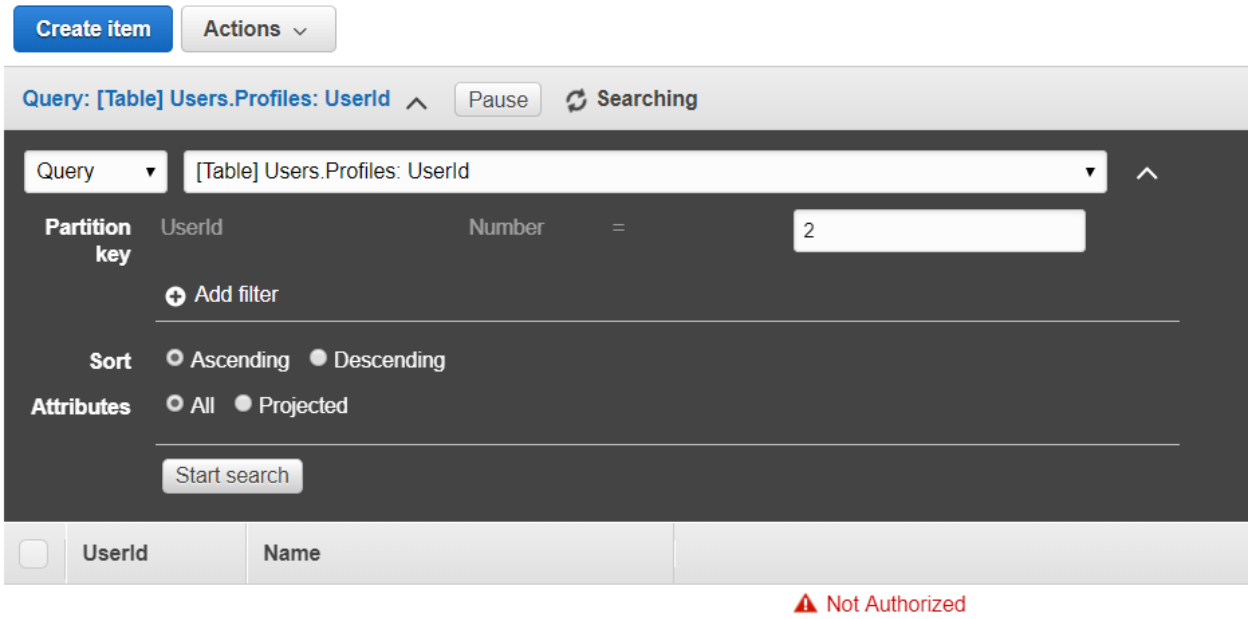
يمكن تحديد هذه السياسات كما في الصورة

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "dynamodb:PutItem",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem",
                "dynamodb:DescribeTable",
                "dynamodb:GetRecords",
                "dynamodb:ListTables",
                "dynamodb:GetRecords"
            ],
            "Resource": "arn:aws:dynamodb:us-east-1:*:table/Users.Profiles",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "dynamodb:LeadingKeys": "1"
                }
            }
        }
```

حيث يمكن للمستخدم فقط قراءة المفتاح الفهرسي (رقم 1) فقط كما في الصورة



لكن لو بحثنا بالمفتاح الفهرسي (رقم 2) نلاحظ عدم السماح بذلك



ملاحظة

عند الإتصال بقاعدة بيانات DynamoDb بإستخدام شيفرة برمجية فيجب معرفة أنه إذا كان الاتصال داخل شبكة أمازون ففي هذه الحالة يمكن الاعتماد والوثوق بإعدادات شبكة أمازون أما إذا كان الاتصال من الانترنت أي خارج شبكة أمازون ففي هذه الحالة يجب التأكد من أن عملية الاتصال تستخدم البروتوكول HTTPs وهي الاعدادات الافتراضية ضمن الإعدادات التي يتم تجهيزها في شيفرة المصدر ضمن مكتبة SDK.

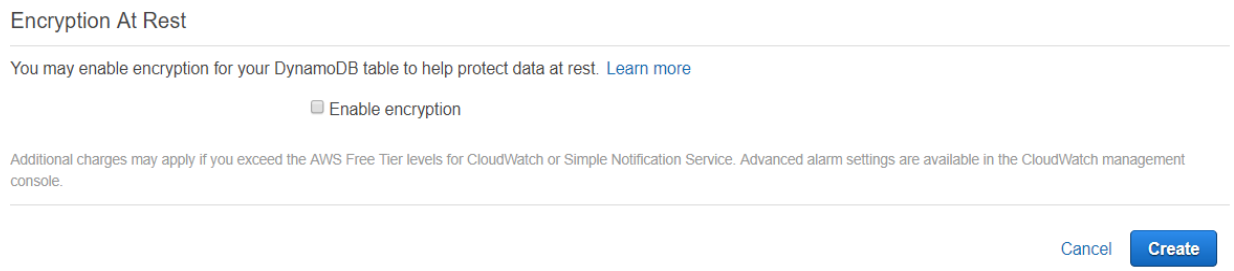
تشفير البيانات المخزنة في قاعدة بيانات DynamoDb

من الميزات التي تمت إضافتها مؤخرًا وأعلن عنها في مؤتمر AWS re:Invent 2017 هي إمكانية تخزين البيانات بشكل كامل في قاعدة البيانات DynamoDb بشكل مشفر لحماية البيانات الحساسة و التوافق مع السياسات العالمية التي تخص أمن المعلومات.

يمكن تفعيل خاصية التشفير عند إنشاء جدول جديد لتشفير جميع البيانات والفهارس للجدول بإستخدام الخوازمية AES-256 وخدمة إدارة مفاتيح التشفير KMS الموجودة في أمازون.

مع الأخذ بعين الاعتبار أن طريقة تعاملنا مع الجدول ستبقى كما هي حيث يمكن إضافة وحذف عناصر والاستعلام عنها دون الشعور بأي تأثير أو تغير على سرعة التنفيذ.

لعمل ذلك من لوحة تحكم قاعدة البيانات DynamoDb نلاحظ وجود الخيار عند إنشاء جدول جديد, قم بتفعيل هذا الخيار



ملاحظة

حتى هذه اللحظة لا يمكن تمكين هذا الخيار إلا عند إنشاء جدول جديد ولا يمكن إلغاء تفعيل هذا الخيار بعد تفعيله علمًا أن تفعيل خاصية التشفير لا تضيف أية مصاريف إضافية.

أخيرًا إذا رغبت بتشفير بعض العناصر في الجدول عن طريق تشفيرها في جهة العميل ففي هذه الحالة تأكد من استخدام نوع المحتوى Binary لتخزين القيمة المشفرة في الحقل الخاص بها.

شكرا لكم

اشكر كل من قرأ هذه الورقات ووجد فيها المفيد من المعلومات التي تساعد على نشر المعرفة وتعليم الأخرين. ويسعدنا مشاركة ملاحظاتكم على عناوين البريد الإلكتروني التالية:

Ismaeel.enjreny@gmail.com

fabdulwahab@outlook.com