

The background features a vibrant, multi-colored gradient. It starts with a dark blue on the left, transitions through purple and magenta, and then into bright orange and yellow towards the right. A diagonal line separates the darker blue/purple area from the lighter orange/yellow area.

AWS
re:Invent

DAT303

Data security best practices on Amazon DynamoDB

Padma Malligarjunan

Senior Product Manager
Amazon DynamoDB
Amazon Web Services

Somu Perianayagam

Principal Engineer
Amazon DynamoDB
Amazon Web Services

Agenda

Protecting your data at rest

Monitoring access to your data

Controlling access to your data

Limiting network access while using Amazon DynamoDB

Protecting your data at rest

All Data Encrypted @ Rest with AWS KMS

- **AWS-owned CMK – default encryption type**
- **AWS-managed CMK**
- **Customer Managed CMK**
- **Table Level Encryption Settings**
- **GSI, Streams, and backups**

Encryption Enabled By Default

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Auto Scaling capacity set to 70% target utilization, at minimum capacity of 5 reads and 5 writes.
- Encryption at Rest with DEFAULT encryption type.

Deselect
default settings

Encryption state of the table

Table details

Table name	ProductCatalogTable
Primary partition key	asin (String)
Primary sort key	type (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	DEFAULT Manage Encryption
KMS Master Key ARN	Not Applicable

Managing Encryption Type

Manage Encryption



DEFAULT

The key is owned by Amazon DynamoDB. You are not charged any fee for using these CMKs.

KMS - Customer managed CMK

The key is stored in your account that you create, own, and manage. AWS Key Management Service (KMS) charges apply. [Learn more](#)

KMS - AWS managed CMK

The key is stored in your account and is managed by AWS Key Management Service (KMS). AWS KMS charges apply.

aws/dynamodb



Cancel

Save

Encryption state of the table

Table name	ProductCatalogTable
Primary partition key	asin (String)
Primary sort key	type (String)
Point-in-time recovery	DISABLED Enable
Encryption Type	KMS Manage Encryption
KMS Master Key ARN	arn:aws:kms:us-east-2:784361292956:key/b65e7cfb-3106-4ff4-bb9f-d6323c8bbdf8

```
% aws dynamodb describe-table --table-name ProductCatalogTable
```

```
...
```

```
"SSEDescription": {  
  "Status": "ENABLED",  
  "SSEType": "KMS",  
  "KMSMasterKeyArn": "arn:aws:kms:us-east-1:111122223333:key/[KMS_KEY_ID]"  
}
```

AWS-managed CMK for DynamoDB

KMS > AWS managed keys

AWS managed keys (1)

Filter keys by alias or key ID

< 1 > ⚙️

Alias	Key ID	Status
aws/dynamodb	b65e7cfb-3106-4ff4-bb9f-d6323c8bbdf8	Enabled

AWS KMS key grants for DynamoDB table

```
% aws kms list-grants --key-id [KMS_KEY_ID]
```

```
...
```

```
  "KeyId": "arn:aws:kms:us-east-1:111122223333:key/[KMS_KEY_ID]",
```

```
  ...,
```

```
  "GranteePrincipal": "dynamodb.us-east-1.amazonaws.com",
```

```
  ...,
```

```
  "Constraints": {
```

```
    "EncryptionContextSubset": {
```

```
      "aws:dynamodb:subscriberId": "111122223333",
```

```
      "aws:dynamodb:tableName": "ProductCatalog"
```

```
    }
```

```
  }
```

AWS KMS key AWS CloudTrail event: who

```
{  
  "eventVersion": "1.05",  
  "userIdentity": {  
    ...  
    "arn": "arn:aws:sts::111122223333:assumed-role/ApplicationRole/worker70",  
    ...  
  },  
}
```

AWS KMS key AWS CloudTrail event: what

```
"resources": [  
  {  
    "ARN": "arn:aws:kms:us-east-1:111122223333:key/[KMS_KEY_ID]",  
    "accountId": "111122223333",  
    "type": "AWS::KMS::Key"  
  }  
],
```

AWS KMS key AWS CloudTrail event: from where

...

"eventSource": "kms.amazonaws.com",

"eventName": "Decrypt",

...

"sourceIPAddress": "dynamodb.amazonaws.com",

...

AWS KMS key AWS CloudTrail event: from where

```
"requestParameters": {  
  "encryptionContext": {  
    "aws:dynamodb:tableName": "ProductCatalog",  
    "aws:dynamodb:subscriberId": "111122223333"  
  }  
},
```

AWS KMS & DynamoDB permissions

User must have permissions to

- Create grants and encrypt data when creating a table with CMK encryption

- Use the CMK to decrypt when accessing data in the table (including via DynamoDB streams)

AWS-managed CMK has these permissions by default

- But a user/role policy can deny these permissions

Jane's IAM user permissions

```
{  
  "Effect": "Allow",  
  "Action": "dynamodb:*",  
  "Resource": "*" ,  
},  
{  
  "Effect": "Deny",  
  "NotAction": "dynamodb:*",  
  "Resource": "*" ,  
}
```

Allows all DynamoDB actions
on all DynamoDB resources

Denies everything other than
DynamoDB

Jane's table creation

```
% aws dynamodb create-table --table-name JanesTable \  
                               --provisioned-throughput ReadCapacityUnits=5,...\  
                               ...
```

```
...  
|| CreationDateTime | ItemCount | TableArn | TableId | TableName | ...  
|+-----+-----+-----+-----+-----+...  
|| 1541875075.012 | 0 | arn:aws:dynamodb:us-east-1:676190956703:table/JanesTable | 9f3370a2-4126-40fa-9e6b-9c759a76f477 | JanesTable | ...
```

```
% aws dynamodb create-table --table-name JanesTableEncrypted \  
                               --sse-specification Enabled=true \  
                               --provisioned-throughput ReadCapacityUnits=5,...\  
                               ...
```

An error occurred (AccessDeniedException) when calling the CreateTable operation: KMS key access denied error: ... User: ...:user/Jane is not authorized to perform: kms:DescribeKey on resource: ...:key/[KMS_KEY_ID] with an explicit deny ...

Jane's access to the table

```
% aws dynamodb get-item --table-name ProductCatalogNoEncrypt \  
                        --key '{"Id" : { "N" : "101" } }'
```

```
...  
"Title": {  
    "S": "Book 101 Title"  
},  
...
```

```
% aws dynamodb get-item --table-name ProductCatalog \  
                        --key '{"Id" : { "N" : "101" } }'
```

An error occurred (AccessDeniedException) when calling the GetItem operation: KMS key access denied error: com.amazonaws.services.kms.model.AWSKMSEException: The ciphertext refers to a customer master key that does not exist, does not exist in this region, or you are not allowed to access. (Service: AWSKMS; ...)

Fixing Jane's IAM user permissions

```
{  
  "Effect": "Allow",  
  "Action": "dynamodb:*",  
  "Resource": "*" ,  
},  
{  
  "Effect": "Deny",  
  "NotAction": [ "dynamodb:*", "kms:*" ],  
  "Resource": "*" ,  
}
```

Monitoring access to your data

DynamoDB and CloudTrail

Control plane operations

CreateTable, UpdateTable, etc.

Always enabled

Identifies who did what from where

Data plane operations

PutItem, GetItem, QueryItem, etc.

Not currently logged

Modifications (PutItem, UpdateItem, etc.) can be monitored via DynamoDB streams

Amazon CloudWatch Events

Near real-time tracking of service actions

Customer-defined rules associated with events

Example use case

Notification message when a table deletion is attempted

Troublesome Bob...

```
% date -u; aws --profile bob dynamodb delete-table --table-name MyTestTable
```

Sat Nov 10 20:15:43 UTC 2018

An error occurred (AccessDeniedException) when calling the DeleteTable operation:
User: ...:user/Bob is not authorized to perform: dynamodb:DeleteTable on resource:
...:table/MyTestTable

From	Subject	Date Received	Size
AWS Notifications	Attempted delete of DynamoDB table in account 676190956703 by user/Bob	Sat 11/10/18, 3:16 PM	24.3 KB

Event processed at 2018-11-10 20:15:45.867169

It appears that user/Bob attempted to delete a table which failed with the error message:

User: ...:user/Bob is not authorized to perform: dynamodb:DeleteTable on resource: ...:table/MyTestTable

Configuring DynamoDB delete event notifications

1. Configure CloudWatch Events rule to invoke AWS Lambda function
2. Configure Lambda function for processing event notification
3. Set up Amazon Simple Notification Service (Amazon SNS) topic for email notification

Configuring DynamoDB delete event notifications

1. Configure CloudWatch Events rule to invoke AWS Lambda function
2. Configure Lambda function for processing event notification
- 3. Set up Amazon SNS topic for email notification**

Configure Amazon SNS topic for email delivery

Create new topic

A topic name will be used to create a permanent unique identifier called an Amazon Resource Name (ARN).

Topic name ⓘ

Display name ⓘ

[Cancel](#) [Create topic](#)

Configure Amazon SNS topic for email delivery

Create subscription

Topic ARN	<input type="text" value="arn:aws:sns:us-east-1:676190956703:EmailDelivery"/>
Protocol	<input type="text" value="Email"/>
Endpoint	<input type="text" value="c...@amazon.com"/>

[Cancel](#) [Create subscription](#)

Configure Amazon SNS topic for email delivery

Topic details: EmailDelivery

Publish to topic

Other topic actions ▾

Topic ARN arn:aws:sns:us-east-1:676190956703:EmailDelivery
Topic owner 676190956703
Region us-east-1
Display name

Subscriptions

Create subscription

Request confirmations

Confirm subscription

Other subscription actions ▾



Filter

<input type="checkbox"/>	Subscription ID	Protocol	Endpoint	Subscriber
<input type="checkbox"/>	PendingConfirmation	email	c...@amazo...	

Configure Amazon SNS topic for email delivery

You have chosen to subscribe to the topic:

arn:aws:sns:us-east-1:676190956703:EmailDelivery

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)

Subscription confirmed!

You have subscribed c [redacted] @amazon.com to the topic:
EmailDelivery.

Your subscription's id is:

arn:aws:sns:us-east-1:676190956703:EmailDelivery:db03f9fd-336b-4711-9cee-b3057d2c76f5

If it was not your intention to subscribe, [click here to unsubscribe](#).

Configure Amazon SNS topic for email delivery

Topic details: EmailDelivery

Publish to topic

Other topic actions ▾

Topic ARN arn:aws:sns:us-east-1:676190956703:EmailDelivery
Topic owner 676190956703
Region us-east-1
Display name

Subscriptions

Create subscription

Request confirmations

Confirm subscription

Other subscription actions ▾



Filter

<input type="checkbox"/>	Subscription ID	Protocol	Endpoint	Subscriber
<input type="checkbox"/>	arn:aws:sns:us-east-1:676190956703:EmailDelivery:db03f9fd-336b-4711-9cee-b3057d2c...	email	c[redacted]@amaz...	676190956703

Configuring DynamoDB delete event notifications

1. Configure CloudWatch Events rule to invoke Lambda function
2. **Configure Lambda function for processing event notification**
3. Set up Amazon SNS topic for email notification

Configure Lambda function

Author from scratch

Start with a simple "hello world" example.



Configure Lambda function

Author from scratch [Info](#)

Name

Runtime

Configure Lambda function

Role

Defines the permissions of your function. Note that new roles may not be available for a few minutes after creation. [Learn more](#) about Lambda execution roles.


Create a new role from one or more templates. ▼

Lambda automatically creates a role with permissions from the selected policy templates. Basic Lambda permissions (such as logging to Amazon CloudWatch) are automatically added. If your function accesses a VPC, the required permissions are also added.

Role name

Enter a name for your new role.

DynamoDBAlertsMailDelivery

 This new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Policy templates

Choose one or more policy templates. A role will be generated for you before your function is created. [Learn more](#) about the permissions that each policy template will add to your role.

Amazon SNS publish policy ✕

Sample CloudWatch Event

```
"detail": {
  "eventVersion": "1.06",
  "eventID": "c7650985-8f79-4b8f-a09d-8966247554ad",
  "eventTime": "2018-11-07T15:28:10Z",
  "requestParameters": {
    "tableName": "TestDelTable"
  },
  "eventType": "AwsApiCall",
  "responseElements": {
    "tableDescription": {
      "tableArn": "arn:aws:dynamodb:us-east-1:111122223333:table/TestDelTable",
      "provisionedThroughput": {
        "writeCapacityUnits": 5.0,
        "numberOfDecreasesToday": 0.0,
        "readCapacityUnits": 5.0
      },
      "tableSizeBytes": 0.0,
      "tableName": "TestDelTable",
      "tableStatus": "DELETING",
      "tableId": "1903977f-1d73-4c7c-9768-6f6b6f437a36",
      "itemCount": 0.0
    },
  },
},
```

Sample CloudWatch Event

```
"awsRegion": "us-east-1",
"eventName": "DeleteTable",
"readOnly": "False",
"userIdentity": {
  "userName": "Bob",
  "principalId": "AIDAIDJA...",
  "accessKeyId": "AKIA...",
  "type": "IAMUser",
  "arn": "arn:aws:iam::111122223333:user/Bob",
  "accountId": "111122223333"
},
"eventSource": "dynamodb.amazonaws.com",
"requestID": "84C8DULU00HN3U8ACKFUH7EH9BVV4KQNSO5AEMVJF66Q9ASUAAJG",
"apiVersion": "2012-08-10",
"userAgent": "aws-cli/1.15.83 Python/3.7.0 Darwin/17.7.0 botocore/1.10.82",
"sourceIPAddress": "72...",
"managementEvent": "True",
"recipientAccountId": "111122223333"
},
```

Configure Lambda function

```
lambda_function x (+)
1 import json
2 import boto3
3 import datetime
4 def lambda_handler(event, context):
5
6     detail = event['detail']
7     action = detail['eventName']
8     identity = detail['userIdentity']
9     source = detail['eventSource']
10    invokerARN = identity['arn']
11    invokerID = invokerARN.split(":")[5]
12    account = identity['accountId']
13
```

Configure Lambda function

```
14 message = 'Event processed at %s\n\n' % str(datetime.datetime.now())
15
16 if 'errorMessage' in detail:
17     message += 'It appears that %s attempted to delete a table which failed ' % invokerID
18     message += 'with the error message:\n\n'
19     message += '    %s\n' % detail['errorMessage']
20
21     subject = 'Attempted delete of DynamoDB table in account %s by %s' % (account, invokerID)
22 else:
23     tableName = detail['requestParameters']['tableName']
24     message += '%s deleted the table %s in account %s\n' % (invokerID, tableName, account )
25
26     subject = 'Table %s in account %s deleted by %s' % (tableName, account, invokerID )
27
```

Configure Lambda function

```
28 message += '\n\nAPI Invocation details:\n'
29 message += '    DynamoDB API:      ' + action + '\n'
30 message += '    Invoked by: ' + invokerARN + '\n'
31 message += '    Parameters: \n'
32
33 requestParms = detail['requestParameters']
34 ▾ if requestParms is None or requestParms == "None" or len(requestParms) == 0:
35     message += '        None\n'
36 ▾ else:
37 ▾     for k, v in requestParms.items():
38         message += '            %s: %s\n' % (k, v)
39
```

Configure Lambda function

```
40 client = boto3.client('sns')
41 response = client.publish(
42     TargetArn="arn:aws:sns:us-east-1:676190956703:EmailDelivery",
43     Message=message,
44     MessageStructure='string',
45     Subject=subject
46 )
47
48 return 'Done'
```

Configuring DynamoDB delete event notifications

1. **Configure CloudWatch Events rule to invoke Lambda function**
2. Configure Lambda function for processing event notification
3. Setup Amazon SNS topic for email notification

Configure CloudWatch Events rule

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern ⓘ Schedule ⓘ

Build event pattern to match events by service ▾

Service Name ▾

Event Type ▾

For AWS API call events, CloudWatch Events supports the same read/write APIs as CloudTrail does. Read-only APIs, such as those that begin with **List**, **Get**, or **Describe** are not supported by CloudWatch Events. [See more details](#) about which services are supported by CloudTrail.

Any operation Specific operation(s)

Configure CloudWatch Events rule

```
▼ Event Pattern Preview Copy to clipboard Edit
{
  "source": [
    "aws.dynamodb"
  ],
  "detail-type": [
    "AWS API Call via CloudTrail"
  ],
  "detail": {
    "eventSource": [
      "dynamodb.amazonaws.com"
    ],
    "eventName": [
      "DeleteTable"
    ]
  ]
}
```

Configure CloudWatch Events rule

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function ▼



Function*

DynamoDBAlerts ▼

- ▶ Configure version/alias
- ▶ Configure input

Jane deletes her table

```
% date -u; aws --profile jane dynamodb delete-table --table-name JanesTable
Sun Nov 11 00:25:37 UTC 2018
{
  "TableDescription": {
    "TableName": "JanesTable",
    ...
```

From	Subject	Date Received	Size
AWS Notifications	Table JanesTable in account 676190956703 deleted by user/Jane	Sat 11/10/18, 7:26 PM	20.8 KB

Event processed at 2018-11-11 00:25:38.814578

user/Jane deleted the table JanesTable in account 676190956703

API Invocation details:

DynamoDB API: DeleteTable

...

Controlling access to your data

Amazon DynamoDB access control

Base authorization

Action—the API that is being used

Resource—the table/index/backup/stream that is being accessed

Fine-grained controls

LeadingKeys—the key of the record being accessed

Attributes—the attributes (fields) of the records being accessed

Select—the select parameter of a query or scan request

ReturnValues—whether the requestor asked for values to be returned

ReturnConsumeCapacity—whether the requestor asked for the consumed capacity

Fine-grained access control example scenario

ProductCatalog table, Partition Key = id

Books

Ids in the 100s

Attributes: Title, ISBN, Authors, Price, Dimensions, PageCount, inPublication, ProductCategory

Bicycles

Ids in the 200s

Attributes: Title, Description, BicycleType, Brand, Price, Color, ProductCategory

Users

Alice (works on books)

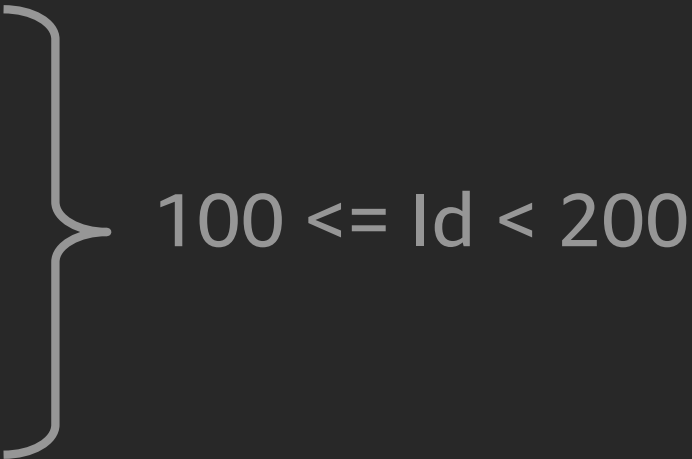
Bob (works on bicycles)

Alice's permissions (Books only)

```
{
  "Effect": "Allow",
  "Action": "dynamodb:ListTables",
  "Resource": "*"
},
{
  "Sid": "FineGrainedTableAccess",
  "Effect": "Allow",
  "Action": [
    "dynamodb:PutItem",
    "dynamodb:GetItem",
    "dynamodb:UpdateItem",
    "dynamodb:Query"
  ],
  "Resource": "arn:aws:dynamodb:us-east-1:676190956703:table/ProductCatalogTable",
```

Alice's permissions (Books only), part 2

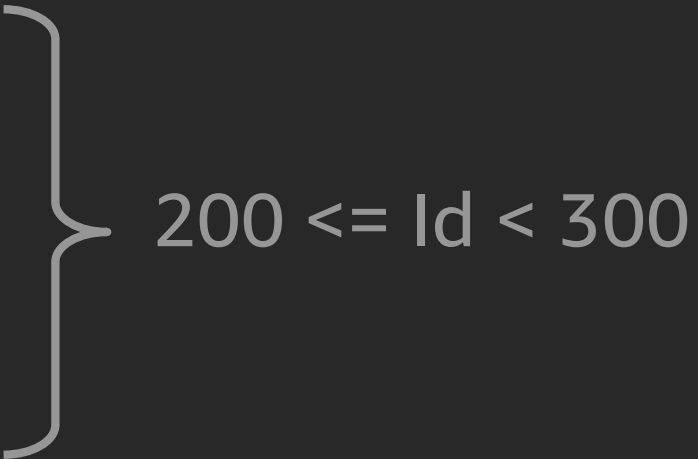
```
"condition": {  
  "ForAllValues:NumericGreaterThanOrEqualTo": {  
    "dynamodb:LeadingKeys": "100"  
  },  
  "ForAllValues:NumericLessThan": {  
    "dynamodb:LeadingKeys": "200"  
  },  
  "ForAllValues:StringLike": {  
    "dynamodb:Attributes": [  
      "Id", "Title", "ISBN", "Authors", "Price",  
      "Dimensions", "PageCount", "inPublication", "ProductCategory"  
    ]  
  }  
}
```



100 <= Id < 200

Bob's permissions (Bikes only) , part 2 (skipped part 1)

```
"condition": {
  "ForAllValues:NumericGreaterThanOrEqualTo": {
    "dynamodb:LeadingKeys": "200"
  },
  "ForAllValues:NumericLessThan": {
    "dynamodb:LeadingKeys": "300"
  },
  "ForAllValues:StringLike": {
    "dynamodb:Attributes": [
      "Id", "Title", "Description", "BicycleType",
      "Brand", "Price", "Color", "ProductCategory"
    ]
  }
}
```



200 <= Id < 300

Permission impacts with Alice

```
% aws --profile Alice dynamodb get-item --table-name ProductCatalogTable \  
--key '{ "Id" : { "N" : "99" } }' \  
--projection-expression 'Id, Title'
```

An error occurred (AccessDeniedException) when calling the GetItem operation: User: arn:aws:iam::676190956703:user/Alice is not authorized to perform: dynamodb:GetItem on resource: arn:aws:dynamodb:us-east-1:676190956703:table/ProductCatalog

More permission impacts with Alice

```
% aws --profile Alice dynamodb get-item --table-name ProductCatalogTable \  
--key '{ "Id" : { "N" : "101" } }' \  
--projection-expression 'Id, Title'  
{  
  "Item": {  
    "Id": {  
      "N": "101"  
    },  
    "Title": {  
      "S": "Book 101 Title"  
    }  
  }  
}
```

More permission impacts with Alice

```
% aws --profile Alice dynamodb get-item --table-name ProductCatalogTable \  
--key '{ "Id" : { "N" : "101" } }' \  
--projection-expression 'Id, Brand'
```

An error occurred (AccessDeniedException) when calling the GetItem operation: User: arn:aws:iam::676190956703:user/Alice is not authorized to perform: dynamodb:GetItem on resource: arn:aws:dynamodb:us-east-1:676190956703:table/ProductCatalogTable

Permission impacts with Bob

```
% aws --profile Bob dynamodb get-item --table-name ProductCatalogTable \  
                                         --key '{ "Id" : { "N" : "199" } }' \  
                                         --projection-expression 'Id, Title, Brand'
```

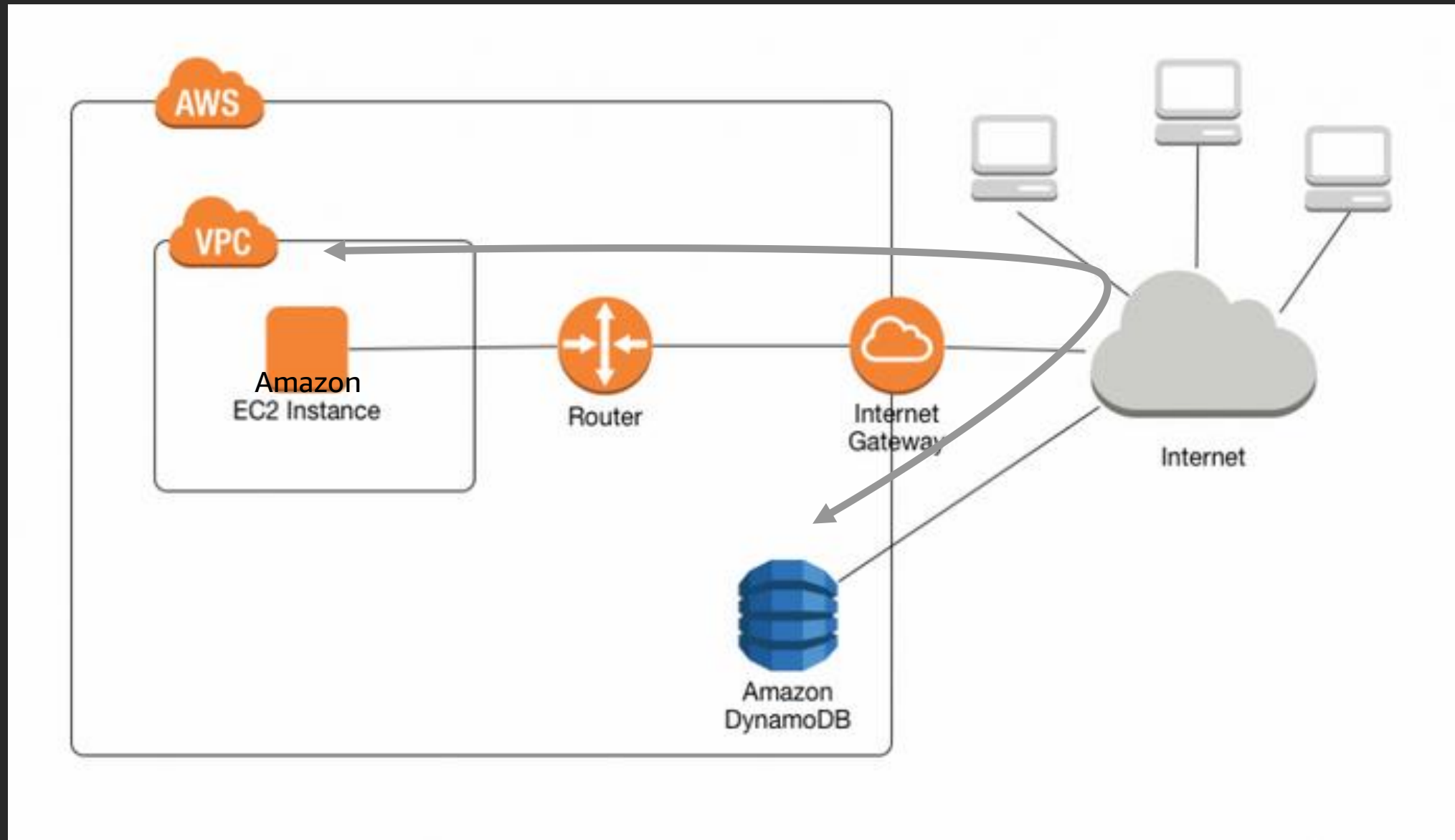
An error occurred (AccessDeniedException) when calling the GetItem operation: User: arn:aws:iam::676190956703:user/Bob is not authorized to perform: dynamodb:GetItem on resource: arn:aws:dynamodb:us-east-1:676190956703:table/ProductCatalogTable

More permission impacts with Bob

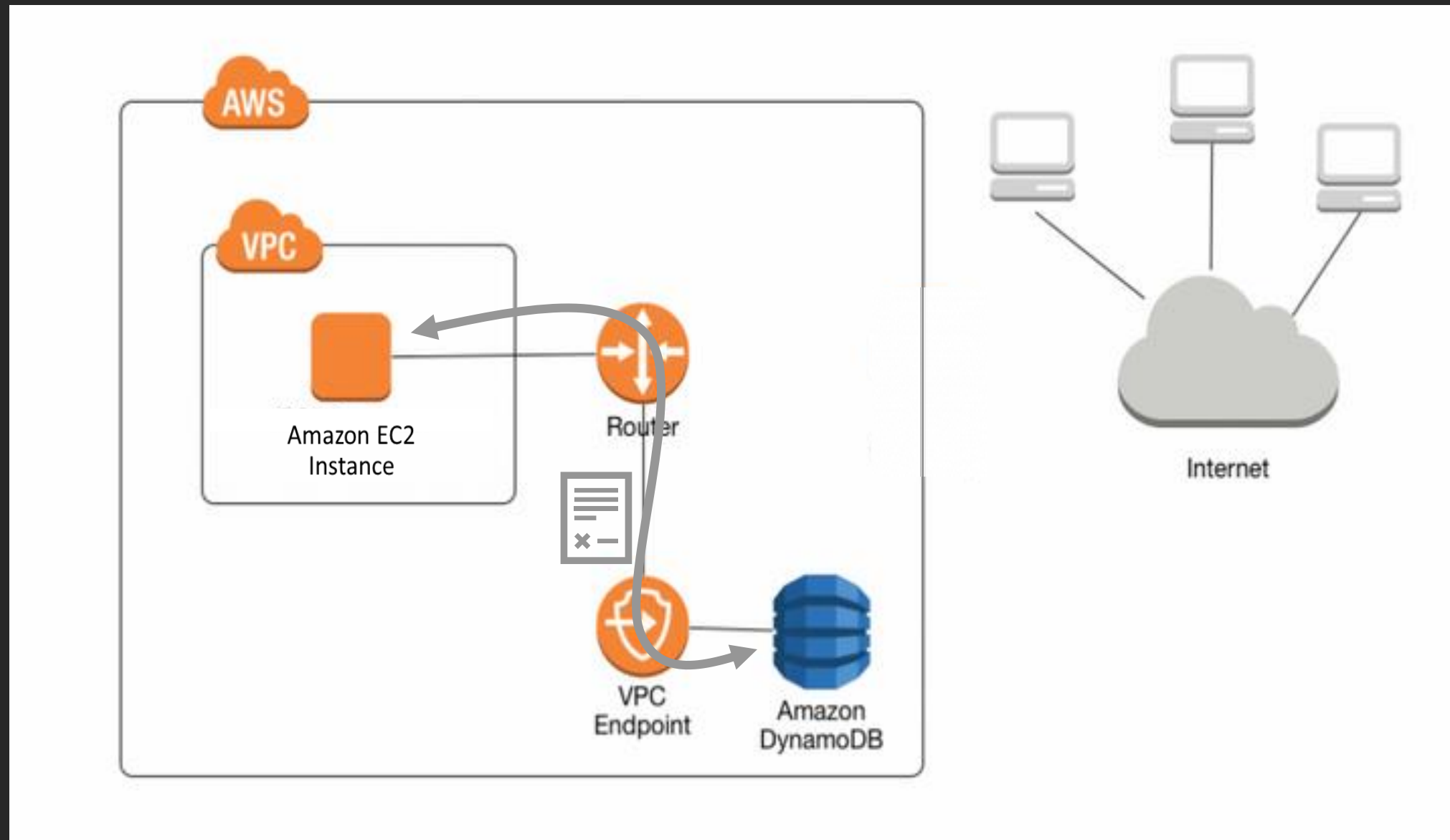
```
% aws --profile Bob dynamodb get-item --table-name ProductCatalog \
--key '{ "Id" : { "N" : "201" } }' \
--projection-expression 'Id, Title, Brand'
{
  "Item": {
    "Id": {
      "N": "201"
    },
    "Title": {
      "S": "18-Bike-201"
    },
    "Brand": {
      "S": "Mountain A"
    }
  }
}
```

Limiting network access while using DynamoDB

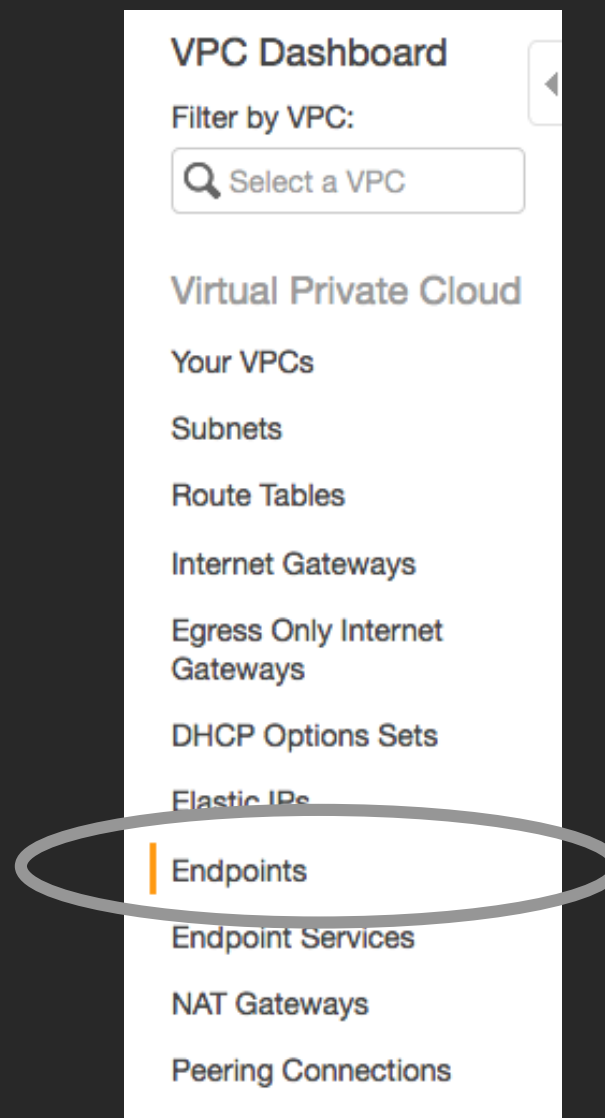
Traditional virtual private cloud (VPC) access to DynamoDB



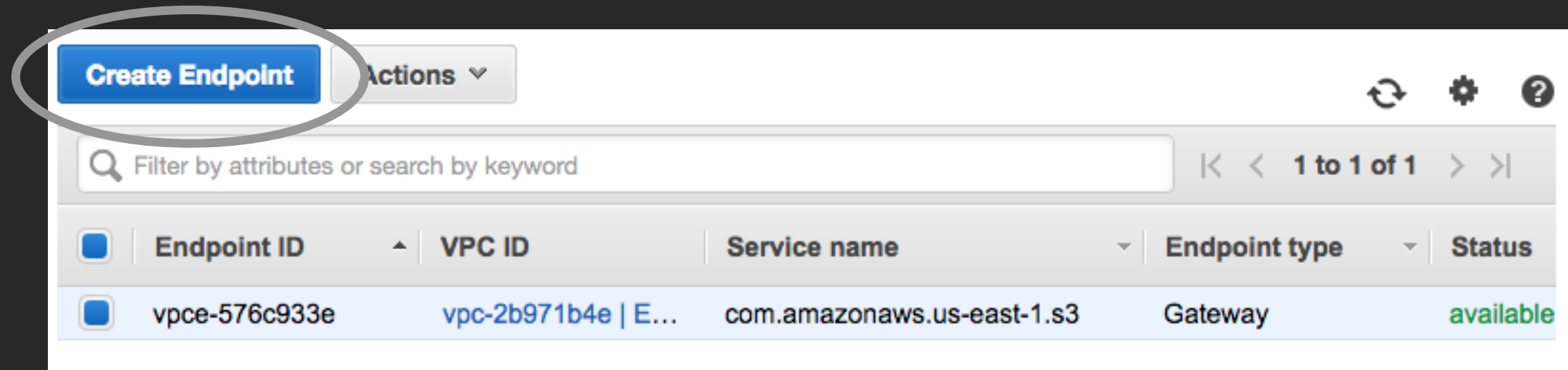
VPC endpoint access to DynamoDB



Setting up a VPC endpoint for DynamoDB



Viewing VPC endpoints



The screenshot displays the AWS VPC console interface for viewing VPC endpoints. At the top left, a blue button labeled "Create Endpoint" is circled in red. To its right is a grey "Actions" dropdown menu. Below these are icons for refresh, settings, and help. A search bar contains the text "Filter by attributes or search by keyword". To the right of the search bar, navigation controls show "1 to 1 of 1". Below the search bar is a table with the following columns: Endpoint ID, VPC ID, Service name, Endpoint type, and Status. A single row is visible in the table.

<input type="checkbox"/>	Endpoint ID	VPC ID	Service name	Endpoint type	Status
<input type="checkbox"/>	vpce-576c933e	vpc-2b971b4e E...	com.amazonaws.us-east-1.s3	Gateway	available

Selecting Amazon DynamoDB

Create Endpoint

A VPC endpoint allows you to securely connect your VPC to another service.

An interface endpoint is powered by [PrivateLink](#), and uses an elastic network interface (ENI) as an entry point for traffic destined to the service.

A gateway endpoint serves as a target for a route in your route table for traffic destined for the service.

- Service category**
- AWS services
 - Find service by name
 - Your AWS Marketplace services

<input type="radio"/>	com.amazonaws.us-east-1.config	amazon	Interface
<input checked="" type="radio"/>	com.amazonaws.us-east-1.dynamodb	amazon	Gateway
<input type="radio"/>	com.amazonaws.us-east-1.ec2	amazon	Interface
<input type="radio"/>	com.amazonaws.us-east-1.ec2messages	amazon	Interface

Warning: source IP address will change



Warning

When you use an endpoint, the source IP addresses from your instances in your affected subnets for accessing the AWS service in the same region will be private IP addresses, not public IP addresses. Existing connections from your affected subnets to the AWS service that use public IP addresses may be dropped. Ensure that you don't have critical tasks running when you create or modify an endpoint.

Source IP address restriction policy for Bob

```
"Effect": "Allow",
"Action": [ "dynamodb:*" ],
"Resource": [ "*" ],
"Condition": {
    "IpAddress": {
        "aws:SourceIp": [ "<CORP_IP_ADDRESS>" ]
    }
}
```

Bob accesses his database from his VPC

```
% aws --profile Bob dynamodb list-tables
```

```
{ "TableNames": [ "MyTestTable" ] }
```

```
==== Dynamodb VPC endpoint is enabled in Bob's VPC ====
```

```
% aws --profile Bob dynamodb list-tables
```

```
An error occurred (AccessDeniedException) when calling the ListTables operation: User: arn:aws:iam::676190956703:user/Bob is not authorized to perform: dynamodb:ListTables on resource: *
```



Source IP or VPC endpoint restriction policy

```
"Effect": "Allow",
"Action": [ "dynamodb:*" ],
"Resource": [ "*" ],
"Condition": {
    "IpAddressIfExists": {
        "aws:SourceIp": [ "<CORP_IP_ADDRESS>" ]
    },
    "StringEqualsIfExists": {
        "aws:SourceVpce": [ "vpce-0a87fbf7ad21cd41f" ]
    }
}
```

VPC endpoint-only restriction policy


```
"Effect": "Allow",  
"Action": [ "dynamodb:*" ],  
"Resource": [ "*" ],  
"Condition": {  
    "StringEquals": {  
        "aws:SourceVpce": [ "vpce-0a87fbf7ad21cd41f" ]  
    }  
}
```

Select your VPC and route table for the VPC endpoint

VPC*  

Configure route tables A rule with destination **pl-02cd2c6b (com.amazonaws.us-east-1.dynamodb)** and a target with this endpoints' ID (e.g. vpce-12345678) will be added to the route tables you select below.

Subnets associated with selected route tables will be able to access this endpoint.



Route Table ID	Main	Associated With
<input checked="" type="checkbox"/> rtb-142aa371	Yes	subnet-5c7e9277 EC2-Public-sub1

VPC endpoint policy

Limit policy

Limits what the VPC endpoint can be used for

Does not grant access—IAM policies grant access


Controls

Principals—who can use the endpoint

Actions—what DynamoDB APIs they can invoke

Resources—what resources can be accessed

Set your VPC endpoint policy: full access

- Policy***
- Full Access - Allow access by any user or service within the VPC using credentials from any AWS accounts to any resources in this AWS service. All policies — IAM user policies, VPC endpoint policies, and AWS service-specific policies (e.g. Amazon S3 bucket policies, any S3 ACL policies) — must grant the necessary permissions for access to succeed. 
 - Custom

- Enables all DynamoDB access within and across AWS accounts
- Existing applications that use DynamoDB under the hood continue working

VPC endpoint policy: only my organization

```
"statement": [ {  
  "Effect": "Allow",  
  "Principal": "*",  
  "Action": "dynamodb:*",  
  "Resource": "*",  
  "Condition": {  
    "StringEquals": { "aws:PrincipalOrgID": "<MY-ORGANIZATION-ID" }  
  }  
} ]
```

Any resource

The principal's organization

DynamoDB Tagging

- Labels attached to AWS resources
- Tags make it easier to manage, search, filter
- View AWS bills broken down by tags
- Tag at creation time
- Allowed operations: Add, List, Edit, Delete

Further exploration

Continuous backup and point-in-time restore

<https://aws.amazon.com/dynamodb/backup-restore/>

Includes ability to restore deleted tables

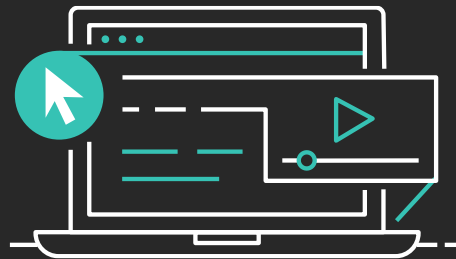
Monitoring Amazon DynamoDB configuration with AWS Config

Automates service configuration monitoring tasks

<https://docs.aws.amazon.com/config/latest/developerguide/WhatIsConfig.html>

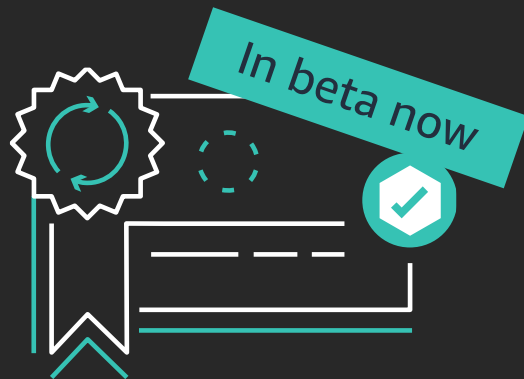
Learn databases with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate database skills



25+ free digital training courses cover topics and services related to databases, including:

- Amazon Aurora
- Amazon Neptune
- Amazon DocumentDB
- Amazon DynamoDB
- Amazon ElastiCache
- Amazon Redshift
- Amazon RDS



Validate expertise with the new **AWS Certified Database - Specialty** beta exam

Visit aws.training

Thank you!

Padma Malligarjunan
@theRealPadma

Somu Perianayagam
@somu_peri



Please complete the session survey in the mobile app.