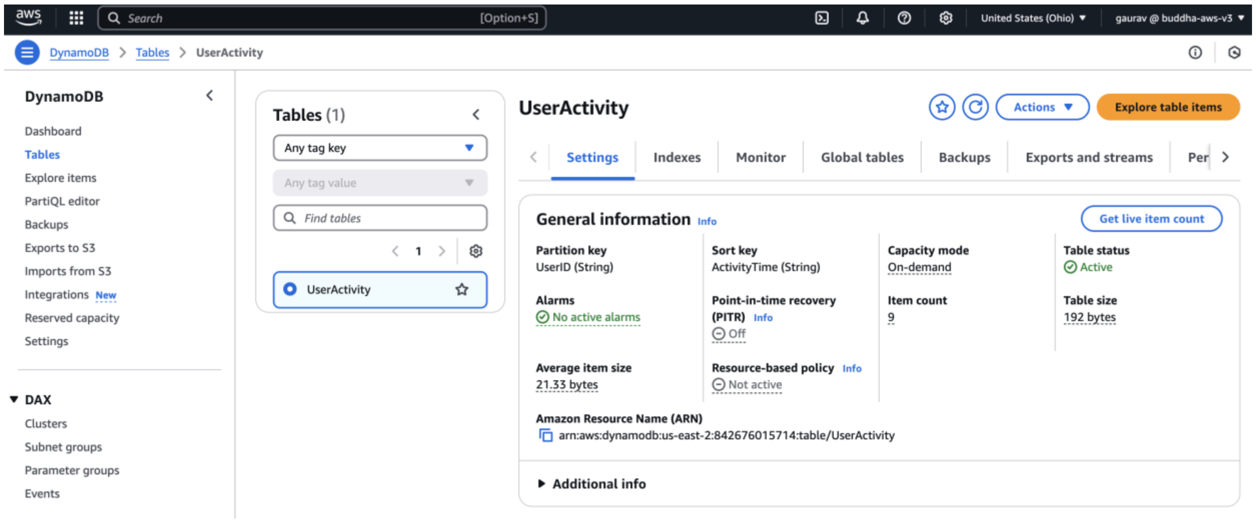
Create a DynamoDB table named "UserActivity" with a primary key "UserID" (partition key) and "ActivityTime" (sort key).



DynamoDB > Tables > UserActivity

UserActivity

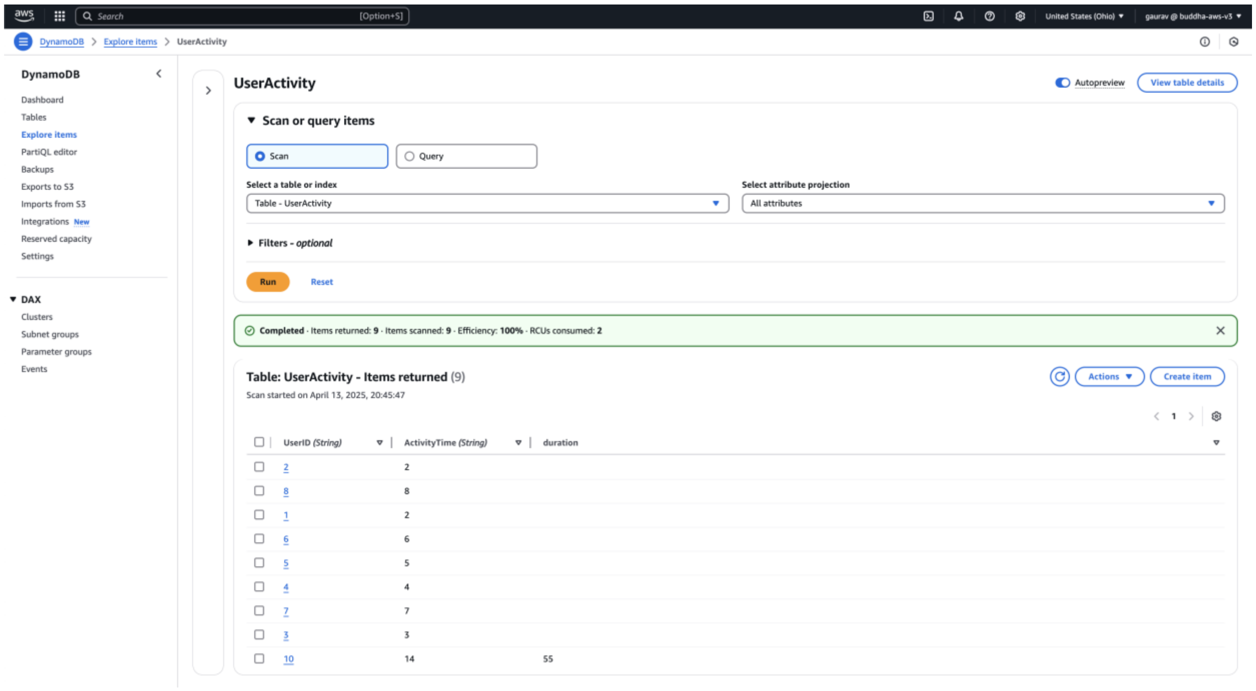
Settings | Indexes | Monitor | Global tables | Backups | Exports and streams

General information

Partition key UserID (String)	**Sort key** ActivityTime (String)	**Capacity mode** On-demand	**Table status** Active
Alarms No active alarms	**Point-in-time recovery (PITR)** Off	**Item count** 9	**Table size** 192 bytes
Average item size 21.33 bytes	**Resource-based policy** Not active		

Amazon Resource Name (ARN)
arn:aws:dynamodb:us-east-2:842676015714:table/UserActivity

Added 5 items in the database



DynamoDB > Explore items > UserActivity

UserActivity

Scan or query items: [✓ Scan] Query

Select a table or index: Table - UserActivity

Select attribute projection: All attributes

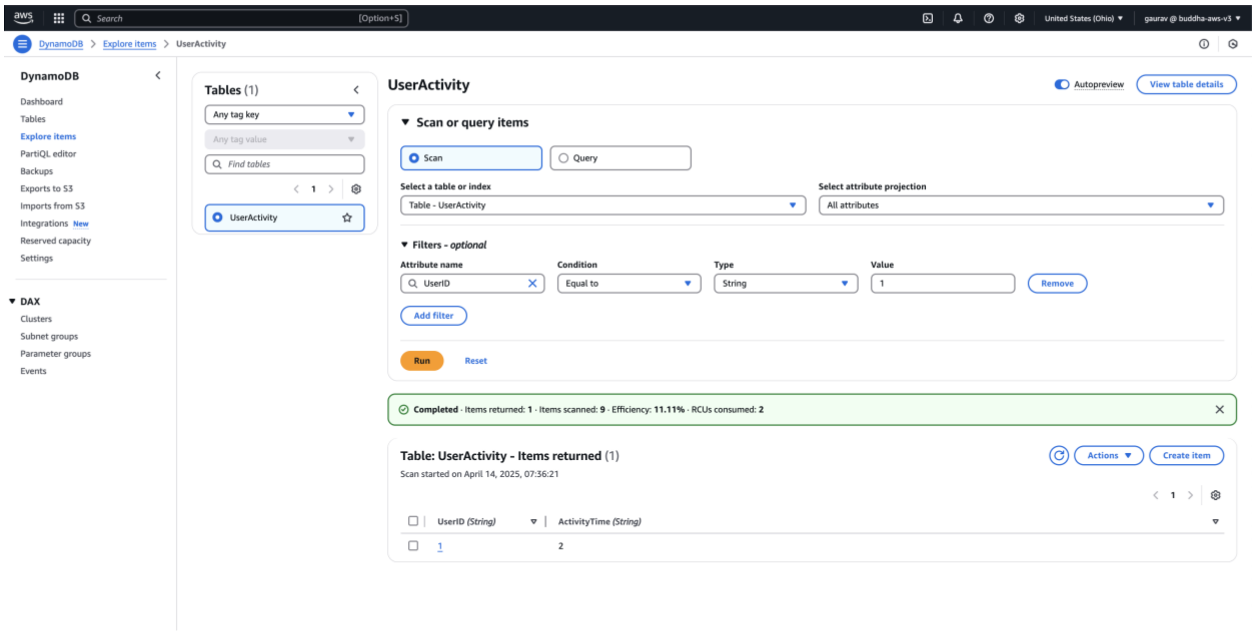
Completed · Items returned: **9** · Items scanned: **9** · Efficiency: **100%** · RCUs consumed: **2**

Table: UserActivity - Items returned (9)

Scan started on April 13, 2025, 20:45:47

UserID *(String)*	ActivityTime *(String)*	duration
2	2	
8	8	
1	2	
6	6	
5	5	
4	4	
7	7	
3	3	
10	14	55

Query to get all the activities of specific user id.



UserActivity

Scan or query items

[✓ Scan] Query

Select a table or index: Table - UserActivity

Select attribute projection: All attributes

Filters - *optional*

Attribute name	Condition	Type	Value
UserID	Equal to	String	1

Add filter · Run · Reset

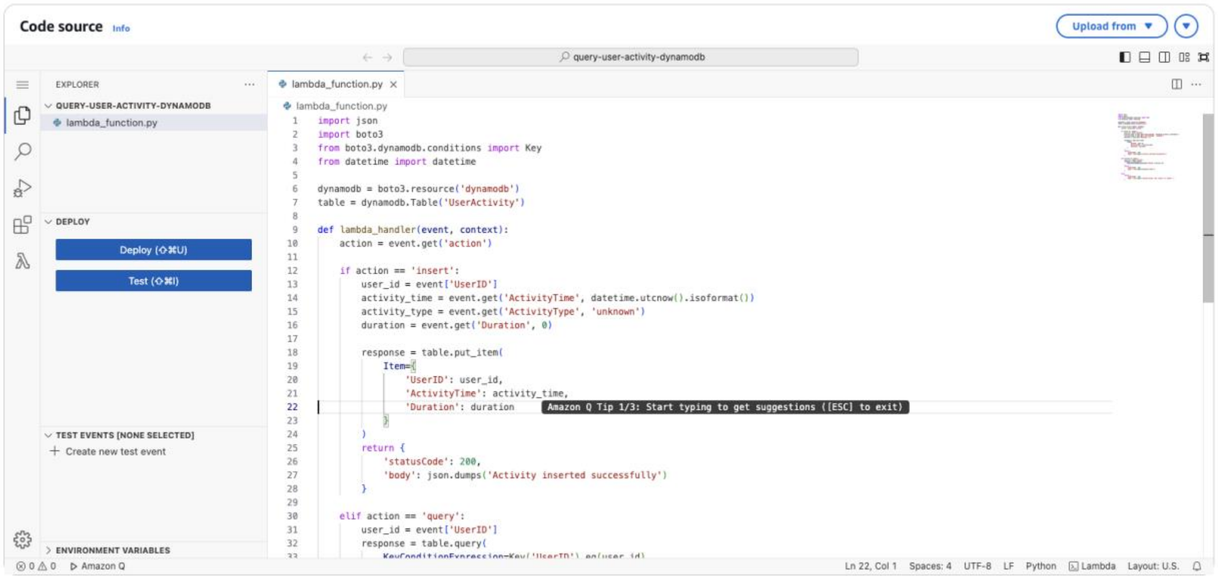
Completed · Items returned: **1** · Items scanned: **9** · Efficiency: **11.11%** · RCUs consumed: **2**

Table: UserActivity - Items returned (1)

Scan started on April 14, 2025, 07:36:21

UserID *(String)*	ActivityTime *(String)*
1	2

Code to insert and get data from dynamodb

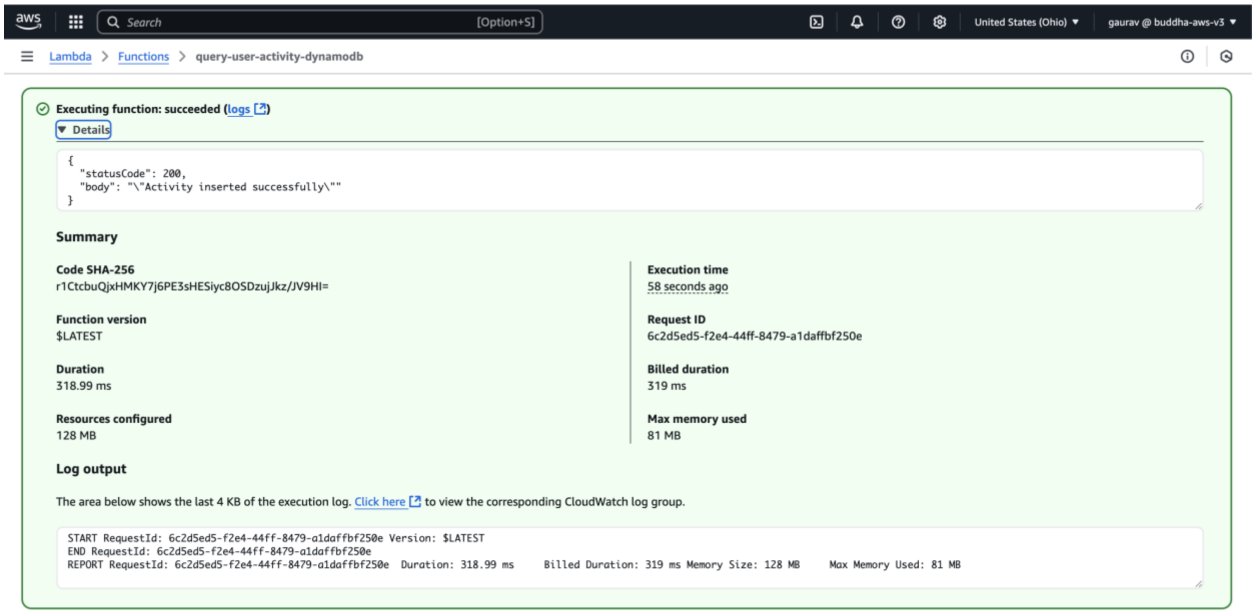


Code source Info
Upload from
query-user-activity-dynamodb
EXPLORER
QUERY-USER-ACTIVITY-DYNAMODB
lambda_function.py
DEPLOY
Deploy (⇧⌘U)
Test (⇧⌘I)
TEST EVENTS [NONE SELECTED]
Create new test event
ENVIRONMENT VARIABLES
Amazon Q
Ln 22, Col 1 Spaces: 4 UTF-8 LF Python Lambda Layout: U.S.

```python
import json
import boto3
from boto3.dynamodb.conditions import Key
from datetime import datetime

dynamodb = boto3.resource('dynamodb')
table = dynamodb.Table('UserActivity')

def lambda_handler(event, context):
    action = event.get('action')

    if action == 'insert':
        user_id = event['UserID']
        activity_time = event.get('ActivityTime', datetime.utcnow().isoformat())
        activity_type = event.get('ActivityType', 'unknown')
        duration = event.get('Duration', 0)

        response = table.put_item(
            Item={
                'UserID': user_id,
                'ActivityTime': activity_time,
                'Duration': duration
            }
        )
        return {
            'statusCode': 200,
            'body': json.dumps('Activity inserted successfully')
        }

    elif action == 'query':
        user_id = event['UserID']
        response = table.query(
            KeyConditionExpression=Key('UserID').eq(user_id)
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

Query Executed Successfully



Lambda > Functions > query-user-activity-dynamodb

Executing function: succeeded (logs)

Details

```
{
  "statusCode": 200,
  "body": "\"Activity inserted successfully\""
}
```

Summary

Code SHA-256
r1CtcbuQjxHMKY7j6PE3sHESiyc8OSDzujJkz/JV9HI=

Execution time
58 seconds ago

Function version
$LATEST

Request ID
6c2d5ed5-f2e4-44ff-8479-a1daffbf250e

Duration
318.99 ms

Billed duration
319 ms

Resources configured
128 MB

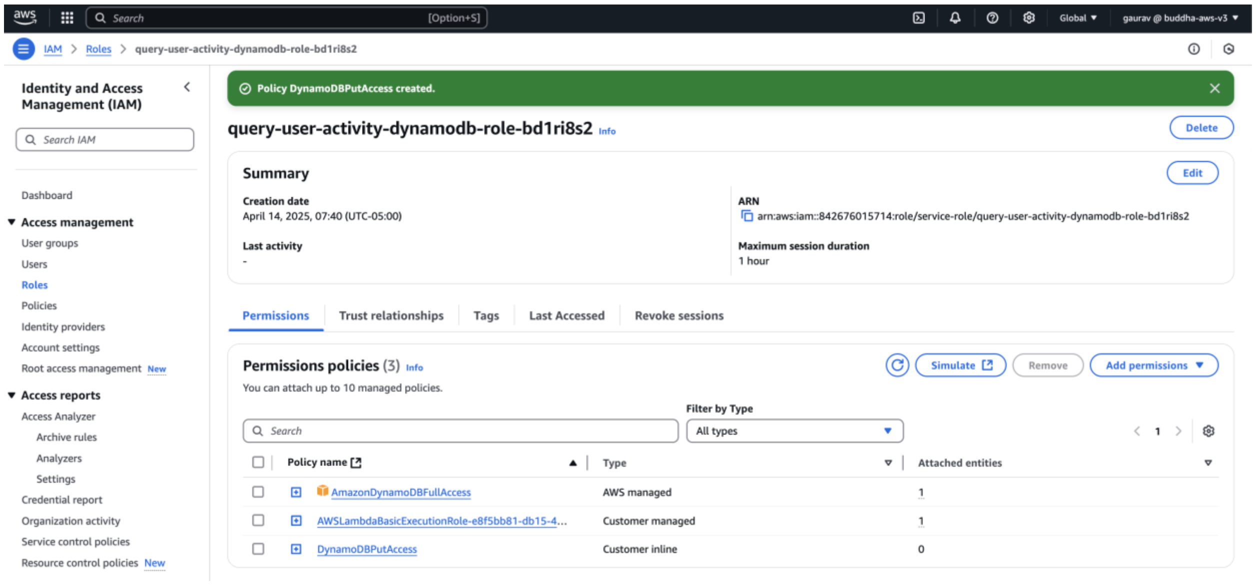
Max memory used
81 MB

Log output

The area below shows the last 4 KB of the execution log. Click here to view the corresponding CloudWatch log group.

```
START RequestId: 6c2d5ed5-f2e4-44ff-8479-a1daffbf250e Version: $LATEST
END RequestId: 6c2d5ed5-f2e4-44ff-8479-a1daffbf250e
REPORT RequestId: 6c2d5ed5-f2e4-44ff-8479-a1daffbf250e  Duration: 318.99 ms     Billed Duration: 319 ms Memory Size: 128 MB     Max Memory Used: 81 MB
```

IAM Role for S3 to access dynamodb



IAM > Roles > query-user-activity-dynamodb-role-bd1ri8s2

Policy DynamoDBPutAccess created.

query-user-activity-dynamodb-role-bd1ri8s2

Summary

Creation date
April 14, 2025, 07:40 (UTC-05:00)

ARN
arn:aws:iam::842676015714:role/service-role/query-user-activity-dynamodb-role-bd1ri8s2

Last activity
-

Maximum session duration
1 hour

Permissions policies (3)

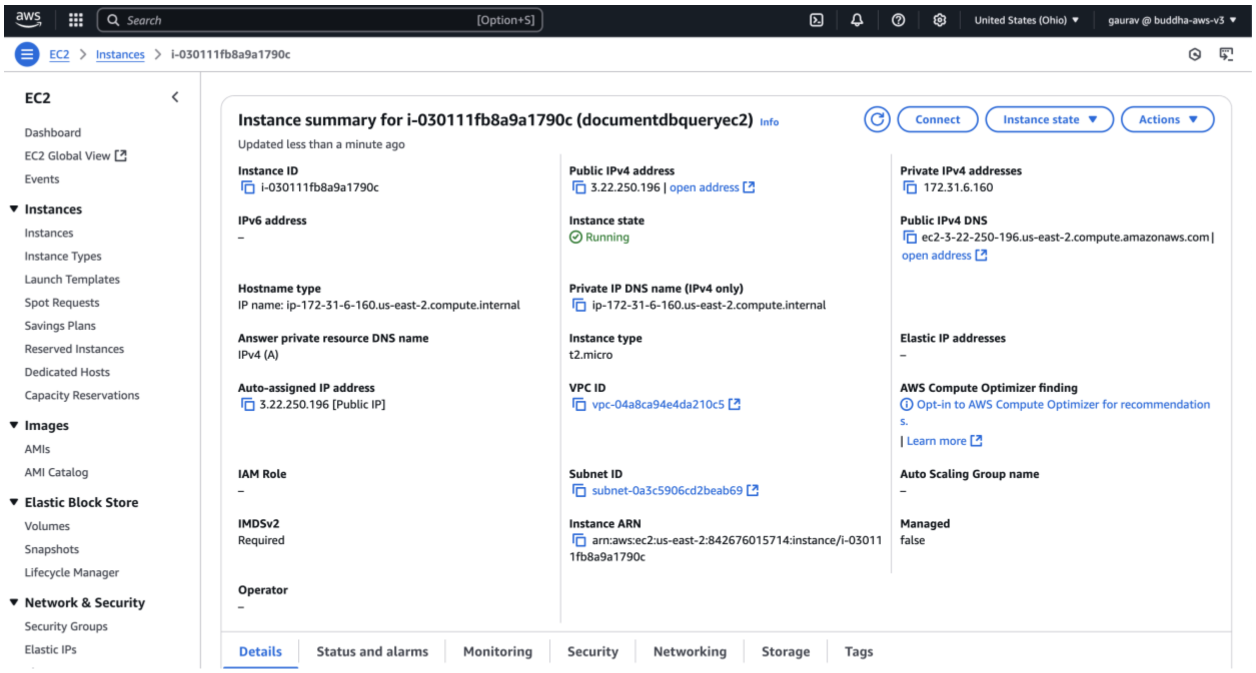
You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonDynamoDBFullAccess	AWS managed	1
AWSLambdaBasicExecutionRole-e8f5bb81-db15-4...	Customer managed	1
DynamoDBPutAccess	Customer inline	0

Task 2

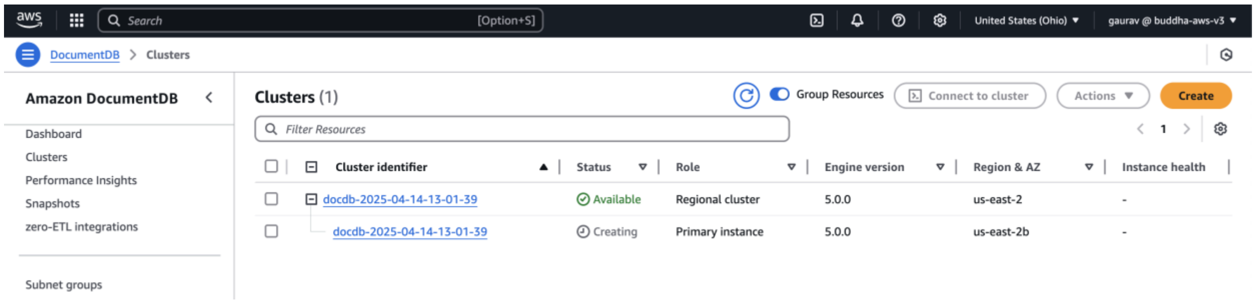
EC2 Instance to connect to the Document DB



Instance summary for i-030111fb8a9a1790c (documentdbqueryec2) Info
Updated less than a minute ago

Instance ID i-030111fb8a9a1790c	**Public IPv4 address** 3.22.250.196 | open address	**Private IPv4 addresses** 172.31.6.160
IPv6 address –	**Instance state** Running	**Public IPv4 DNS** ec2-3-22-250-196.us-east-2.compute.amazonaws.com | open address
Hostname type IP name: ip-172-31-6-160.us-east-2.compute.internal	**Private IP DNS name (IPv4 only)** ip-172-31-6-160.us-east-2.compute.internal	
Answer private resource DNS name IPv4 (A)	**Instance type** t2.micro	**Elastic IP addresses** –
Auto-assigned IP address 3.22.250.196 [Public IP]	**VPC ID** vpc-04a8ca94e4da210c5	**AWS Compute Optimizer finding** Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role –	**Subnet ID** subnet-0a3c5906cd2beab69	**Auto Scaling Group name** –
IMDSv2 Required	**Instance ARN** arn:aws:ec2:us-east-2:842676015714:instance/i-030111fb8a9a1790c	**Managed** false
Operator –		

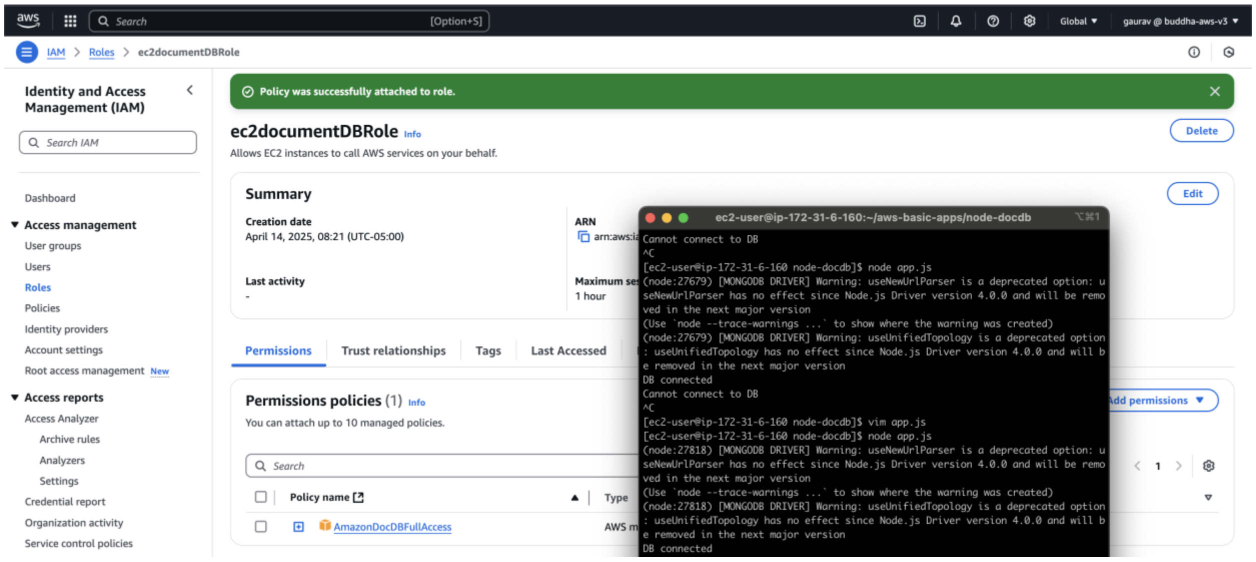
Document DB cluster



Clusters (1)

Cluster identifier	Status	Role	Engine version	Region & AZ	Instance health
docdb-2025-04-14-13-01-39	Available	Regional cluster	5.0.0	us-east-2	-
docdb-2025-04-14-13-01-39	Creating	Primary instance	5.0.0	us-east-2b	-

Connected to the Document DB from EC2 instance and ran the command.



Policy was successfully attached to role.

ec2documentDBRole

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: April 14, 2025, 08:21 (UTC-05:00)

Last activity: -

Maximum session duration: 1 hour

Permissions policies (1)

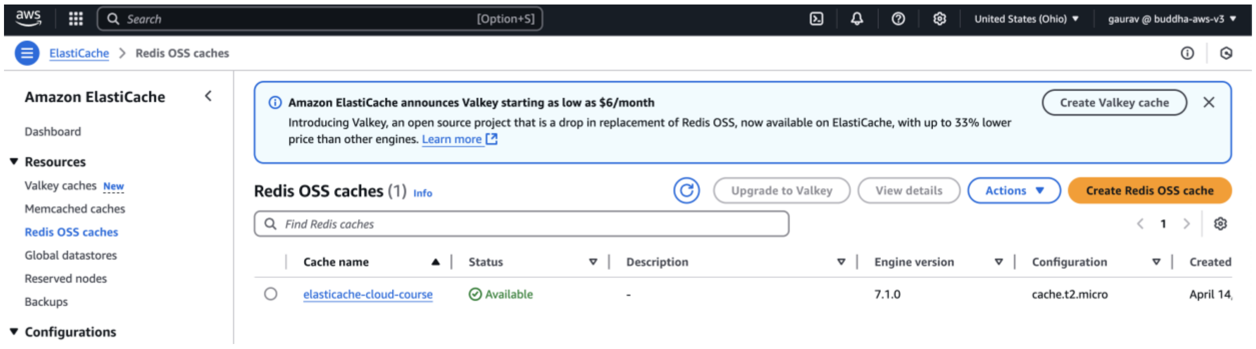
You can attach up to 10 managed policies.

Policy name	Type
AmazonDocDBFullAccess	AWS m

```
ec2-user@ip-172-31-6-160:~/aws-basic-apps/node-docdb
Cannot connect to DB
^C
[ec2-user@ip-172-31-6-160 node-docdb]$ node app.js
(node:27679) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: u
seNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be remo
ved in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:27679) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option
: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will b
e removed in the next major version
DB connected
Cannot connect to DB
^C
[ec2-user@ip-172-31-6-160 node-docdb]$ vim app.js
[ec2-user@ip-172-31-6-160 node-docdb]$ node app.js
(node:27818) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: u
seNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be remo
ved in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:27818) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option
: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will b
e removed in the next major version
DB connected
```

Task 3:

Created Elastic cache cluster



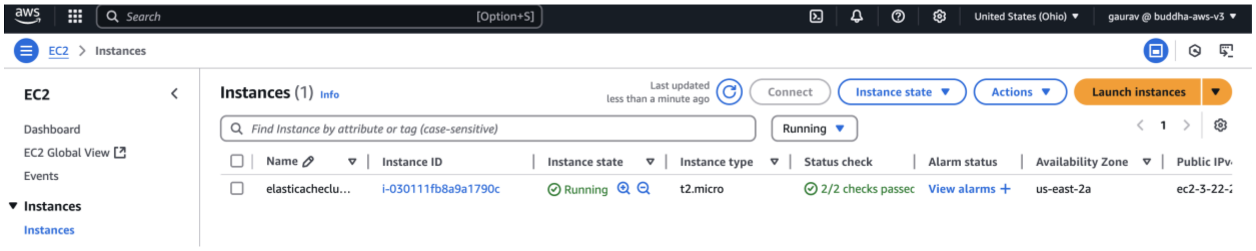
Amazon ElastiCache announces Valkey starting as low as $6/month

Introducing Valkey, an open source project that is a drop in replacement of Redis OSS, now available on ElastiCache, with up to 33% lower price than other engines. Learn more

Redis OSS caches (1)

Cache name	Status	Description	Engine version	Configuration	Created
elasticache-cloud-course	Available	-	7.1.0	cache.t2.micro	April 14,

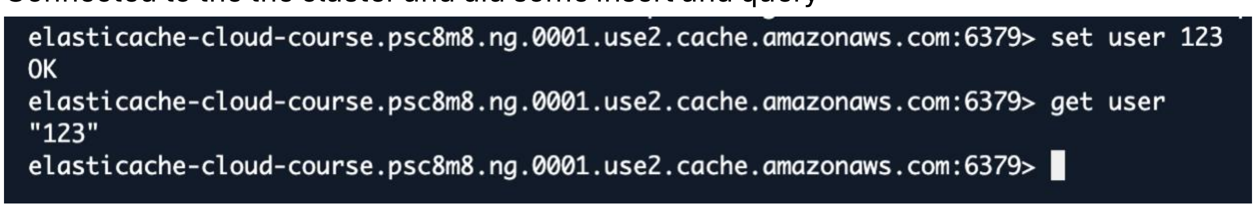
EC2 Instance to connect to Elastic cache



Instances (1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv
elasticacheclu...	i-030111fb8a9a1790c	Running	t2.micro	2/2 checks passec	View alarms	us-east-2a	ec2-3-22-

Connected to the the cluster and did some insert and query



```
elasticache-cloud-course.psc8m8.ng.0001.use2.cache.amazonaws.com:6379> set user 123
OK
elasticache-cloud-course.psc8m8.ng.0001.use2.cache.amazonaws.com:6379> get user
"123"
elasticache-cloud-course.psc8m8.ng.0001.use2.cache.amazonaws.com:6379>
```