

AZ 204 Certification + Cosmos DB

Contents:

- 1. Video course**
 - a. Creating a CosmosDB**
 - b. Creating a container**
 - c. Create a new item**
 - d. New item with metadata explained**
 - e. Query the container in SQL Explorer**
 - f. Calculate usage using an emulator**
 - g. Scale up-Cost Management- limit cost**
 - h. Scale out - Replicate Data Globally**
 - i. cosmosdb1acc | Default consistency**
 - j. Stored Proc**
- 2. Az204-lab 5.1-scholarhat-Creating a Cosmos DB Account - SQL API- completed successfully**
- 3. Az204-lab 5.2-scholarhat-Creating a Cosmos DB account - Table API- completed successfully**

Video course

Creating a CosmosDB

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL ...

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource Group *
[Create new](#)

Instance Details

Account Name *

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ⓘ ☐ Enable ☒ Disable

Location * ⓘ

Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot access a region, click [here](#) for more details on how to create a region access request.


Capacity mode ⓘ ☒ Provisioned throughput ☐ Serverless

[Learn more about capacity mode](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimate

Apply Free Tier Discount ☒ Apply ☐ Do Not Apply

Limit total account throughput ☒ Limit the total amount of throughput that can be provisioned on this account

 This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.

[Review + create](#)

[Previous](#)

[Next: Global Distribution](#)

Next pane: Global Distribution

S

Global Distribution (Accept Default)

Geo-Redundancy ☐ Disable ☒ Multi-region Writes ☐ Disable ☒

Next pane: Networking(Accept Default)

All public networks

Next Backup: (Accept Default)

Next Encryption: (Accept Default)

Tags

[CREATE a COSMOS DB] OK


1.b.Creating a container

Container is similar to a SQL TABLE

Database id Create new : mydbtbl

* Container id: users

* Partition key: /zipcode



New Container X

* **Database id** ⓘ

☒ Create new ☐ Use existing

☒ Share throughput across containers ⓘ

* **Database throughput (autoscale)** ⓘ

☒ Autoscale ☐ Manual

Estimate your required RU/s with [capacity calculator](#).

Database Max RU/s ⓘ

*

Your database throughput will automatically scale from **100 RU/s (10% of max RU/s) - 1000 RU/s** based on usage.

Estimated monthly cost (USD) ⓘ: **\$8.76 - \$87.60** (1 region, 100 - 1000 RU/s, \$0.00012/RU)

* **Container id** ⓘ

* **Indexing**

☒ Automatic ☐ Off

All properties in your documents will be indexed by default for flexible and efficient queries. [Learn more](#)

* **Partition key** ⓘ

Error & Solution:

Created a cosmos db account; soon after that there was an Error when creating a container

Message: {"code":"BadRequest","message":"Message: {\"Errors\":[\"Your account is currently configured with a total throughput limit of 1000 RU\\s. This operation failed because it would have increased the total throughput to 1400 RU\\s. See https:\\\\vaka.ms\\cosmos-tp-limit for more information.\"]}\"ActivityId: 75fda8d3-02a3-4231-9f94-75a9ffa408fc, Request URI: /apps/f0bcb51-a121-44e1-a35a-c5de0519a069/services/c8a9aba0-aba8-4043-b02d-bee76197bcd4/partitions/e0768073-7a6d-4eb2-8b7d-23def0a00c34/replicas/133536693015853671p, RequestStats: , SDK: Microsoft.Azure.Documents.Common/2.14.0\"}, Request URI: /dbs, RequestStats: , SDK: Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0, Microsoft.Azure.Documents.Common/2.14.0 ActivityId: c7bf6f79-a5f9-41fd-a27d-a9aeb1646d5c, Microsoft.Azure.Documents.Common/2.14.0

I deleted a database called
ToDoList-> Items collection

After this deletion this error was
gone and the container was
created. 200 OK

Conclusion: At a time there has to
be a container to meet the limit of
1000RTU

1C.Create a new item-> new item and fill key val pair

```
{
  "id": "01",
  "name": "Deepika",
  "zipcode": "60016",
  "address": "Des Plaines"
}
```

1D.New item with metadata

```
{
  "id": "01",
  "name": "Deepika",
  "zipcode": "60016",
  "address": "Des Plaines",
  "_rid": "XU12ANYicREBAAAAAAAAA==",
  "_self":
  "dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicREBAAAAAAAAA==/",
  "_etag": "\"4f00268e-0000-0200-0000-662d47fa0000\"",
  "_attachments": "attachments/",
  "_ts": 1714243578
}
```

Explained:

"_rid": resource id
"_self": connection to itself with r_id contained. Identify a record from its collection and db
_self has database id, collection id, and r_id repeated again
"_etag": for caching; for caching the content,
"_attachments": attachment information
"_ts": timestamp created and update

1.e. Query the container in SQL Explorer

```
SELECT * FROM c where c.id='02'
```

```
[
  {
    "id": "02",
    "Name": "Niranth",
    "zipcode": "47906",
    "address": "West Lafayette",
    "_rid": "XU12ANYicRECAAAAAAAAAA==",
    "_self":
    "dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicRECAAAAAAAAAA==/",
    "_etag": "\"4f00c9a4-0000-0200-0000-662d49eb0000\"",
    "_attachments": "attachments/",
    "_ts": 1714244075
  }
]
```

Or query using root ie select * from root 'instead of' select * from c

```
select * from root where root.zipcode='60016'
```

```
[
  {
    "id": "01",
    "name": "Deepika",
    "zipcode": "60016",
    "address": "Des Plaines",
    "_rid": "XU12ANYicREBAAAAAAAAA==",
    "_self":
    "dbs/XU12AA==/colls/XU12ANYicRE=/docs/XU12ANYicREBAAAAAAAAA==/",
    "_etag": "\"4f00268e-0000-0200-0000-662d47fa0000\"",
    "_attachments": "attachments/",
    "_ts": 1714243578
  }
]
```

1.e. Calculate usage using an emulator

Measure your usage in cosmosdb

[Azure Cosmos DB Capacity Calculator](https://cosmos.azure.com/capacitycalculator/)

<https://cosmos.azure.com/capacitycalculator/>

The screenshot shows the Azure Cosmos DB Capacity Calculator web interface. The browser address bar displays <https://cosmos.azure.com/capacitycalculator/>. The page title is "Microsoft Azure | Azure Cosmos DB | Capacity Calculator". A note states: "The calculator below offers you a quick estimate of the workload cost on Azure Cosmos DB. For a more precise estimate and ability to tweak more parameters, please sign in with an account you use for Azure."

Azure Cosmos DB Account Settings

The simplified Azure Cosmos DB calculator assumes commonly used settings for indexing policy, consistency, and other parameters. For a more accurate estimate, please [sign in](#) to provide your workload details.

API ☐ Azure Cosmos DB for NoSQL

Number of regions ☐ 1

Multi-region writes ☒ Disabled ☐ Enabled

Workload Information

Specify the amount of data stored in transactional and analytical store. Also specify the nature of requests for your workload.

Total data stored in transactional store ☐ 10 GB

Use Analytical Store ☒ Off ☐ On

Workload mode ☒ Steady ☐ Variable

Workload Item Sizes And Operations

For each of the items in your workload, specify its size and provide anticipated operations volume. [Learn More](#)

Sample item 1

Item size (upto 2048 KB) ☐ 1 KB

Point reads/sec in max-read region ☐ 500

Creates/sec across all regions ☐ 100

Updates/sec across all regions ☐ 100

Deletes/sec across all regions ☐ 0

Queries/sec across all regions ☐ 100

Cost Estimate

Transactional Storage

Cost per GB/month	0.25 USD
Total Data stored per region	x 10 GB
EST. STORAGE COST PER MONTH	2.50 USD

Transactional Workload

Cost per 100 RU/s per hour	0.008 USD
EST. THROUGHPUT REQUIRED Show Details	x 2,985 RU/s
EST. WORKLOAD COST/MONTH	174.34 USD

Number of regions x 1

EST. TOTAL COST/MONTH **176.84 USD**

[Create Cosmos DB account](#)

NEW TO AZURE COSMOS DB? CREATE A NEW FREE TIER ACCOUNT
[Learn more](#)

HAVE AN APP WITH BURSTY TRAFFIC? TRY OUT SERVERLESS
[Learn more](#)

MIGRATE TO AZURE COSMOS DB NOW
[Learn more](#)

1 g Scale up Cost Management- limit cost:

https://portal.azure.com/#@potterdeepikagmail.onmicrosoft.com/resource/subscriptions/076bde24-b356-46bd-b8fd-

Microsoft Azure

Home > cosmosdb1acc | Cost Management

Azure Cosmos DB's cost is calculated based on storage, throughput and additional settings such as replication throughput limit.

Current account throughput

Database: mydbtbl
Total throughput: 1000 RU/s

Number of regions: 1
Total throughput: 1000 RU/s

Total throughput limit setting

The account total throughput limit cannot be lowered below the total throughput currently provisioned across the account.

☐ Limit the account's total provisioned throughput to the amount included in the free tier discount (1000 RU/s)

☒ Allow the account's total throughput to be provisioned up to a custom amount

2000 1000 \$58 /M RU/s

Cost estimate: up to \$117 per month for provisioned throughput.

The actual cost depends on:

- which Azure regions your account is deployed to,
- whether you're using autoscale or manual provisioned throughput,
- any discount that may be associated with your account.

To estimate your provisioned throughput needs, use the [Azure Cosmos DB capacity calculator](#)

☐ No limit, allow the account's total throughput to be provisioned to any amount

Give Feedback
[Help improve this page](#)

update

Better to limit instead of choosing no limit.

1.h.Scale out - Replicate Data Globally

Settings-> Replicate Data Globally

Microsoft Azure

Home > cosmosdb1acc | Replicate data globally

Save Discard Manual failover Service-Managed Failover Feedback

Your account is currently configured with a total throughput limit of 1001 RU/s. This update isn't possible because it would increase the total throughput to 8000 RU/s.

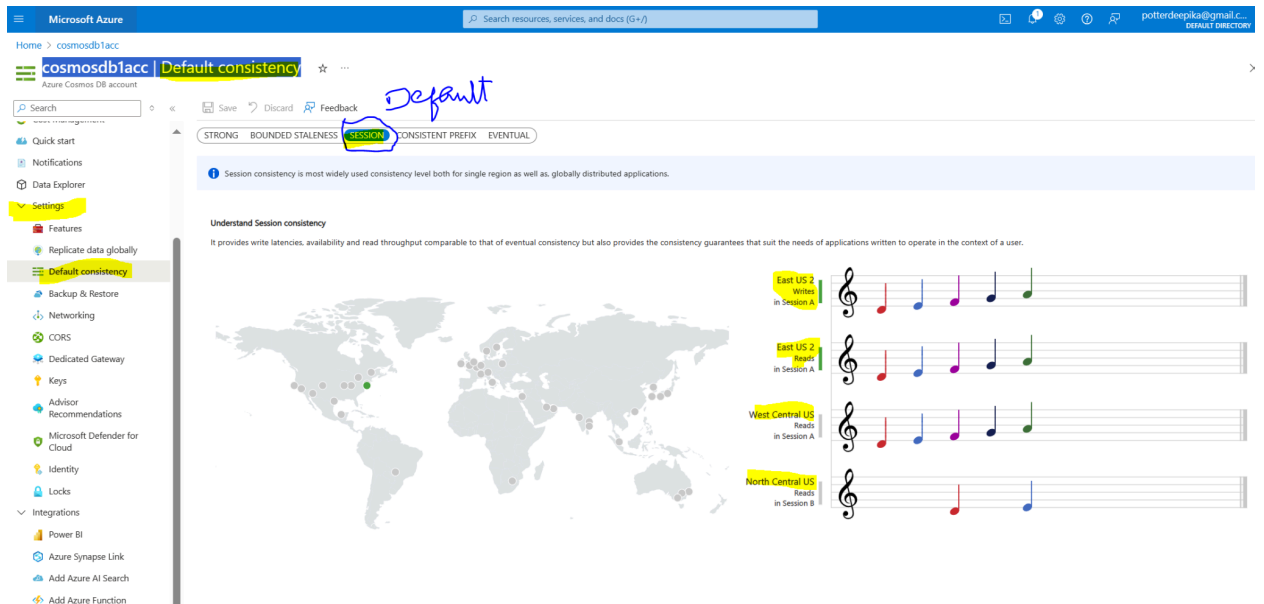
Configure regions
Multi-region writes: ☒ Disable ☐ Enable

Configure the regions for reads, writes and availability zone (supported in selected regions and can only be configured when a new region is added).

+ Add region

Write region	Availability zone	Action
East US 2	<input type="checkbox"/>	
Read regions		
South India	<input checked="" type="checkbox"/>	
Canada Central	<input checked="" type="checkbox"/>	
Australia Southeast	<input checked="" type="checkbox"/>	
Central India	<input checked="" type="checkbox"/>	
East US	<input checked="" type="checkbox"/>	
Canada East	<input checked="" type="checkbox"/>	
North Central US	<input checked="" type="checkbox"/>	

1. i. cosmosdb1acc | Default consistency



1.j. Stored Proc

Create a new sp

Predefined sp

// SAMPLE STORED PROCEDURE

```
function sample(prefix) {
    var collection = getContext().getCollection();

    // Query documents and take 1st item.
    var isAccepted = collection.queryDocuments(
        collection.getSelfLink(),
        'SELECT * FROM root r',
        function (err, feed, options) {
            if (err) throw err;

            // Check the feed and if empty, set the body to 'no docs found',
            // else take 1st element from feed
            if (!feed || !feed.length) {
                var response = getContext().getResponse();
                response.setBody('no docs found');
            }
            else {
                var response = getContext().getResponse();
                var body = { prefix: prefix, feed: feed[0] };
                response.setBody(JSON.stringify(body));
            }
        }
    );

    if (!isAccepted) throw new Error('The query was not accepted by the server.');
```


}

2. Az204-lab 5.1-scholarhat-Creating a Cosmos DB Account - SQL API

Creating a Cosmos DB Account - SQL API

Mentor: Shailendra Chauhan

Type : GuidedLab

Points : 10

Duration : 00:50:00

Lab Details

Description

In this lab, you will be learning how to create an Azure Cosmos DB Account and then work with the SQL API.

Lab Objective

Upon completion of this lab, you will be able to:

- Create an Azure Cosmos DB Account.
- Create a container within Cosmos DB Account.
- Add Default data into the table.
- Querying the added data.

Prerequisites

You should be familiar with:

- Conceptual Understanding of Azure Cosmos DB.
- You must have the Azure Portal Access.

Lab Solution

Azure Cosmos DB is a globally distributed, low latency, multi-model database for managing data at large scales. It is a cloud-based NoSQL database offered as a PaaS (Platform as a Service) from Microsoft Azure. It is a highly available, high throughput, reliable database and is often called a serverless database. Cosmos database contains the Azure Document DB and is available everywhere.

Azure Cosmos DB SQL API accounts provide support for querying items using the Structured Query Language (SQL), one of the most familiar and popular query languages, as a JSON query language.

- Step 1: Open the Azure management portal and log in to <https://portal.azure.com>
- Step 2: Click on “Create a Resource”.

Home >

Create an Azure Cosmos DB account

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Azure Cosmos DB for NoSQL
Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.
[Create](#) [Learn more](#)

Azure Cosmos DB for PostgreSQL
Fully-managed relational database service for PostgreSQL with distributed query execution, powered by the Citus open source extension. Build new apps on single or multi-node clusters—with support for JSONB, geospatial, rich indexing, and high-performance scale-out.
[Create](#) [Learn more](#)

Azure Cosmos DB for MongoDB
Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.
[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Cassandra
Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB.
[Create](#) [Learn more](#)

Azure Cosmos DB for Table
Fully managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB.
[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Gremlin
Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data.
[Create](#) [Learn more](#)

Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL ...

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Try it for free](#), for 30 days with unlimited renewals. Go

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	<input type="text" value="Subscription27"/>
Resource Group *	<input type="text" value="rg0519"/> Create new

Instance Details

Account Name *	<input type="text" value="cosmosdb-sqlapi-lab5adeepika"/>
----------------	---

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ⓘ	<input type="radio"/> Enable <input checked="" type="radio"/> Disable
Location * ⓘ	<input type="text" value="(US) East US"/> <small>Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot Click here for more details on how to create a region access request</small>
Capacity mode ⓘ	<input checked="" type="radio"/> Provisioned throughput <input type="radio"/> Serverless Learn more about capacity mode

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated

Apply Free Tier Discount	<input checked="" type="radio"/> Apply <input type="radio"/> Do Not Apply
Limit total account throughput	<input type="checkbox"/> Limit the total amount of throughput that can be provisioned on this account <small>i This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.</small>

[Review + create](#)

[Previous](#)

[Next: Global Distribution](#)

Step 3: In the search bar, search for “Azure Cosmos DB” and select it from the search results.

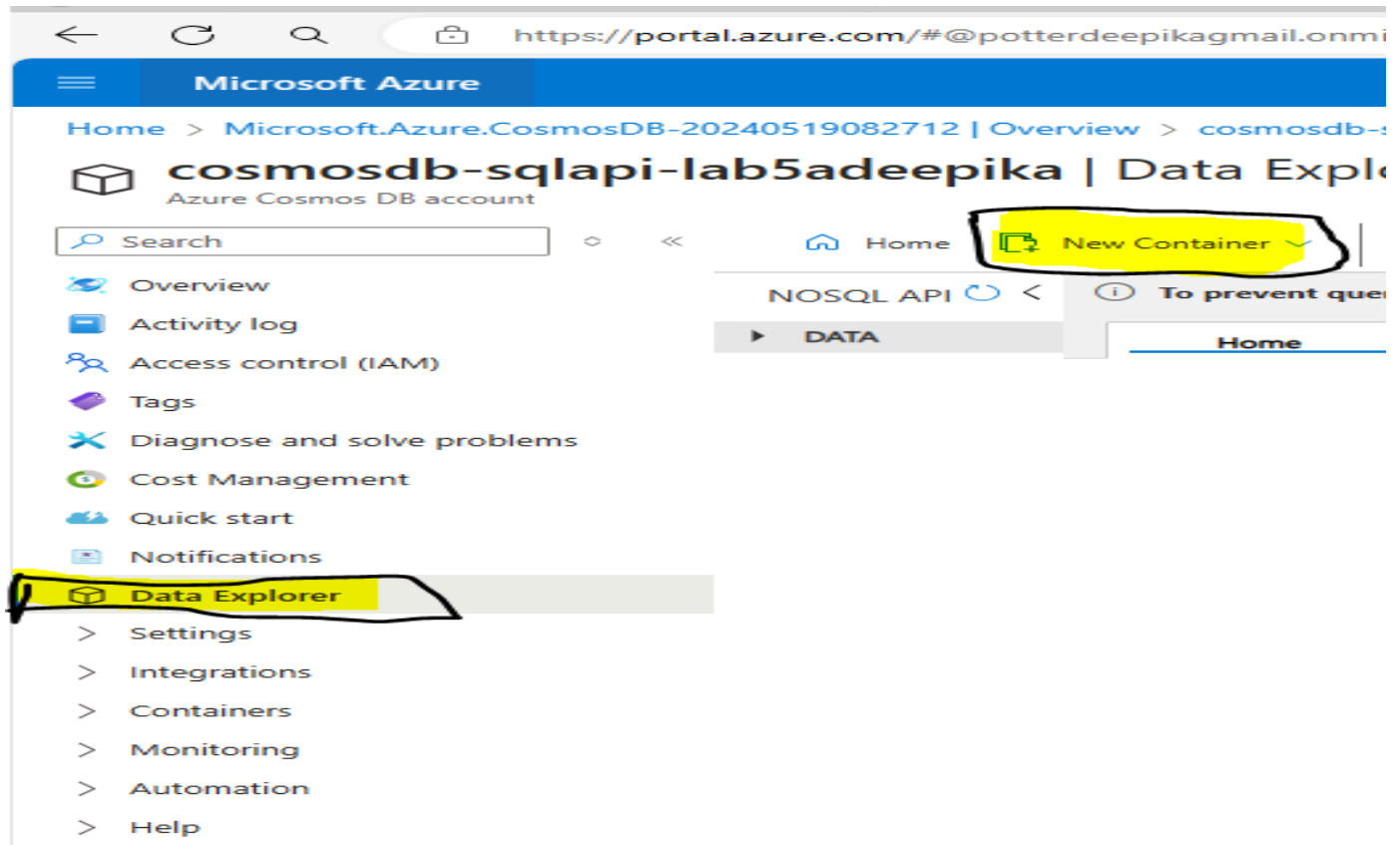
Step 4: Click on the “Create” button.

Step 5: Click on the “Create” button under Core (SQL) - Recommended as shown below.

Step 6: After clicking create button, Fill in the required details:

1. Subscription: Choose a suitable subscription.
2. Resource group: Resource group is a container that holds related resources for an Azure solution. You can keep the resource group as it is or you can also create one by clicking “Create New”.
3. Account name: Mention your account name: “cosmosdb-sqlapi-lab5adeepika”

Step 7: Click on “Review + Create”. Azure will authenticate the details you’ve filled so far and will show a message as “Validation Success”. Afterwards, click on the “Create” button.



Step 8: After clicking the CREATE button. A message will be displayed – “Your deployment is complete”. Then click on “Go to resource”.

Step 9: After Clicking on “Go to resource”, select “Data Explorer” from the left pane. Then, click on “New Container”.

Step 10: A window will be displayed on your right. Enter the “Database Id”, “Customer Id” and “/partition key”. Then, click on the “OK” button.

“Database Id” : Deepika-dbid

“Customer Id” : Customer

“/partition key” : /subscriber

New Container [X]

* Database id ⓘ
☒ Create new ☐ Use existing
Deepika-dbid

☒ Share throughput across containers ⓘ

* Database throughput (autoscale) ⓘ
☒ Autoscale ☐ Manual
Estimate your required RU/s with [capacity calculator](#).
Database Max RU/s ⓘ
3000
Your database throughput will automatically scale from 300 RU/s (10% of max RU/s) - 3000 RU/s based on usage.
Estimated monthly cost (USD) ⓘ: \$26.28 - \$262.80 (1 region, 300 - 3000 RU/s, \$0.00012/RU)

* Container id ⓘ
Customer

* Indexing
☒ Automatic ☐ Off
All properties in your documents will be indexed by default for flexible and efficient queries. [Learn more](#)

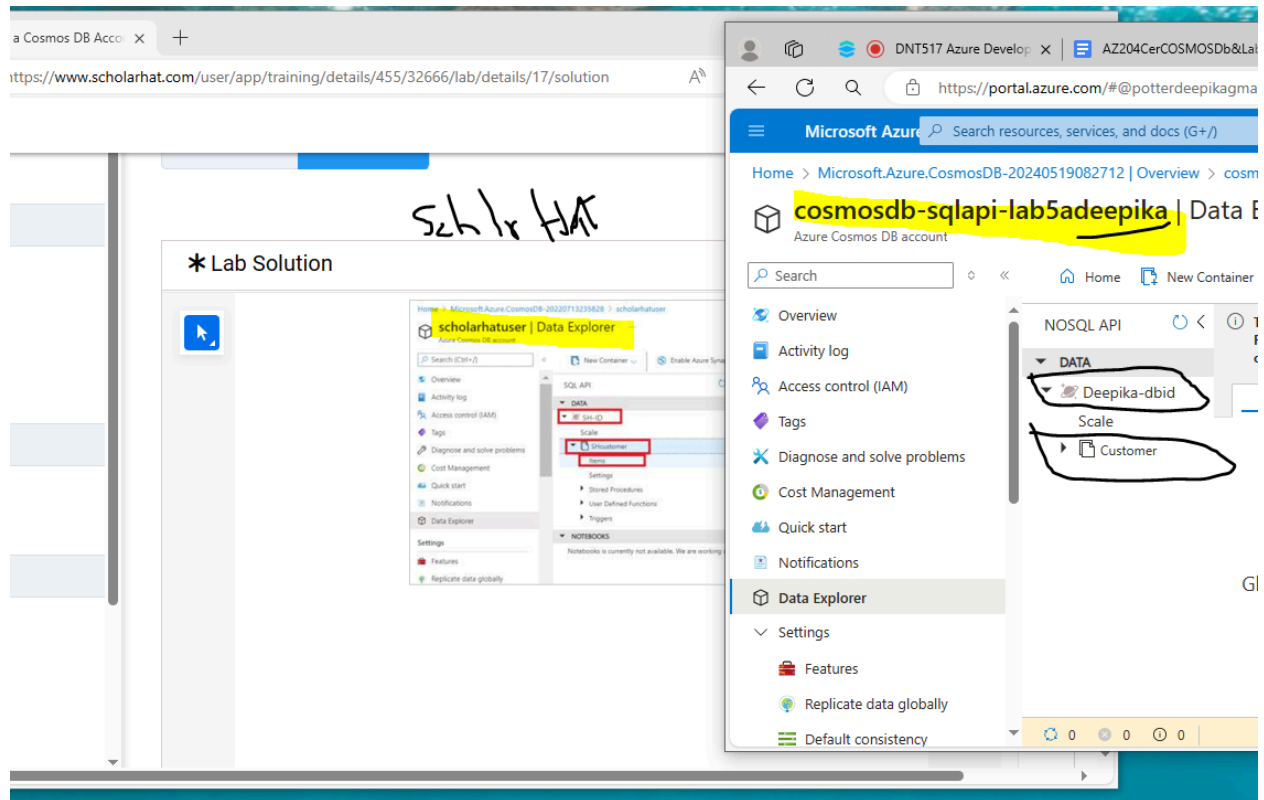
* Partition key ⓘ
/subscriber
[Add hierarchical partition key](#)

Unique keys ⓘ
[+ Add unique key](#)

Analytical store ⓘ
☐ On ☒ Off
Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. [Learn more](#)
[Enable](#)

> Advanced

OK



Step 11: Go to keys menu from left hand menus and copy the primary connection string

Step 12: Select “Data Explorer” from the left pane. Then, Click on SH-ID (your database Id) > SHcustomer > Items. The menu will expand.

“Database Id” : Deepika-dbid

“Customer Id” : Customer

“/partition key” : /subscriber

cosmosdb-sqlapi-lab5adeepika | Keys

Azure Cosmos DB account

keys Refresh Feedback

Settings

Keys

URI

https://cosmosdb-sqlapi-lab5adeepika.documents.azure.com:443/

Read-write Keys

Read-only Keys

PRIMARY KEY

AtK9hdF9mx3uSh5i0K3UCirHvplQu0ZY5VhGY5kpsYbdwF6JDzCXMSiteFoK7f62k1aele4qF82ACDbTCOeEw==

Last regenerated: 5/19/2024 (0 days ago). [Learn more](#)

SECONDARY KEY

AtK9hdF9mx3uSh5i0K3UCirHvplQu0ZY5VhGY5kpsYbdwF6JDzCXMSiteFoK7f62k1aele4qF82ACDbTCOeEw==

Last regenerated: 5/19/2024 (0 days ago). [Learn more](#)

PRIMARY CONNECTION STRING

AccountEndpoint=https://cosmosdb-sqlapi-lab5adeepika.documents.azure.com:443/;AccountKey=AtK9hdF9mx3uSh5i0K3UCirHvplQu0ZY5VhGY5kpsYbdwF6JDzCXMSiteFoK7f62k1aele4qF82ACDbTCOeEw==;

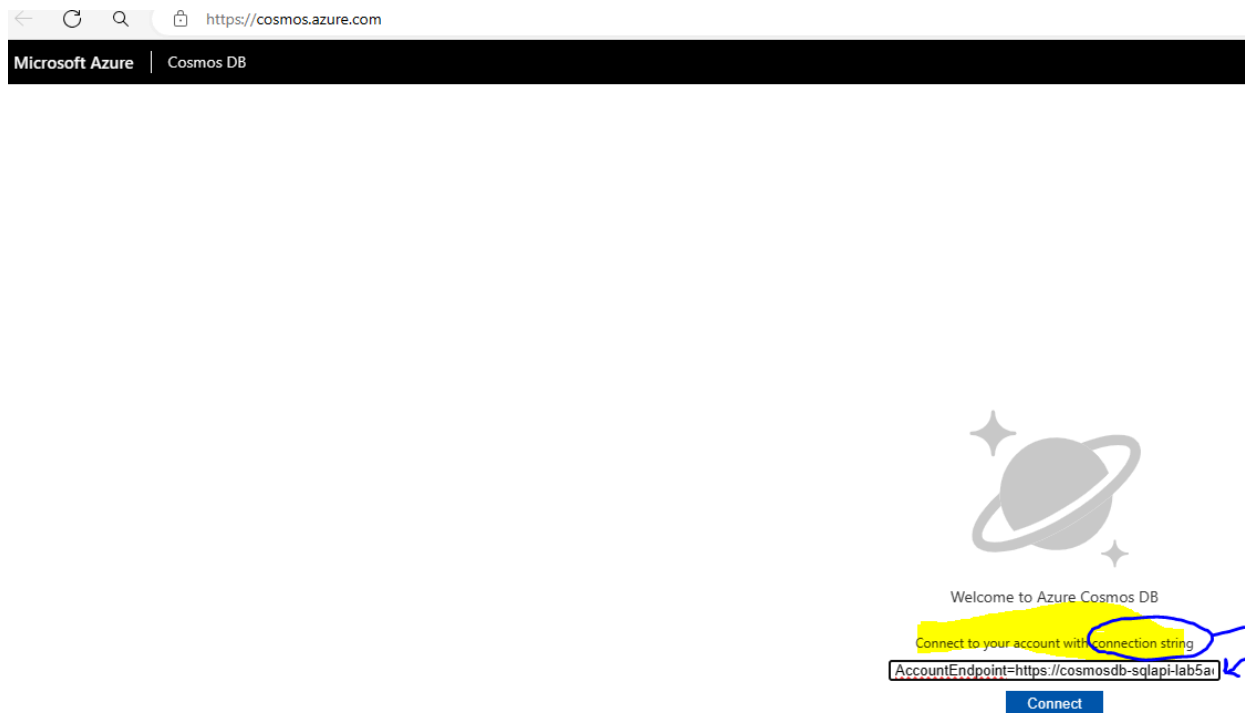
SECONDARY CONNECTION STRING

AccountEndpoint=https://cosmosdb-sqlapi-lab5adeepika.documents.azure.com:443/;AccountKey=AtK9hdF9mx3uSh5i0K3UCirHvplQu0ZY5VhGY5kpsYbdwF6JDzCXMSiteFoK7f62k1aele4qF82ACDbTCOeEw==;

SECONDARY CONNECTION STRING

AccountEndpoint=https://cosmosdb-sqlapi-lab5adeepika.documents.azure.com:443/;AccountKey=AtK9hdF9mx3uSh5i0K3UCirHvplQu0ZY5VhGY5kpsYbdwF6JDzCXMSiteFoK7f62k1aele4qF82ACDbTCOeEw==;

Using connection string connect it for full string



Step 13: Click on the “Open Full Screen” option in the top right corner.

Step 14: Click on the “Open” button under Read and Write.

Step 15: New tab will open, paste the copied key in above steps and click on connect button.

Step 16: As soon as you’ll click on “connect”, it will be redirected to a new tab in your browser.

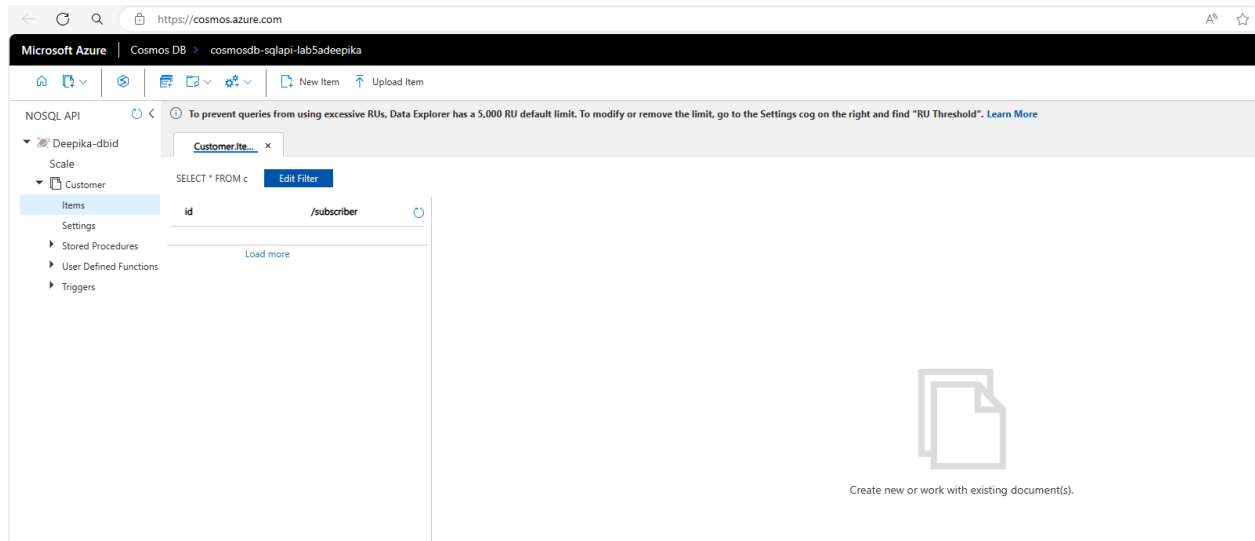
Step 17: Now, in the new tab, you’ll see “Deepika-dbid” in your left pane. Click on Deepika-dbid > Customer > Items

“Database Id” : Deepika-dbid

“Customer Id” : Customer

“/partition key” : /subscriber

Now in Maximize mode



Step 18: Select the “New Item” option in the top pane.

Remember: [scholarhat](#) scholarhat(SH-ID > SHcustomer > Items.
)

Step 19: Delete whatever is written on the screen. Copy the below command and modify it according to your database id and partition key that you’ve entered above.

```
{  
  "id": "1",  
  "Name": "ScholarhatUser1",  
  "scholarhatsubscriber": "New York"  
}  
"Database Id" : Deepika-dbid  
"Customer Id" : Customer  
"/partition key" : /subscriber
```

Derived:Deepika

```
{  
  "id": "1",  
  "Name": "ScholarhatUser1",  
  "subscriber": "New York"  
}
```

Step 20: Now, again click on “New Item”.

- Step 21: Perform step 16 again to enter the second record into the database. Delete whatever is written and paste that command. Remember to do the changes according to your needs. Click on “Save” to save the data.

Step 22: Now, you are going to query the added data. For that, Click on the “New SQL query” icon as highlighted in the below image.

Step 23: You’ll see that there’s a default query written. In case it’s not showing, you can enter the below command to display all the records that you’ve stored above.

SELECT * FROM c Click on the “Execute Query” option in the top pane.

```
{  
  "id": "2",  
  "Name": "Deepika Prasad Lab 5.1",  
  "subscriber": "Chicago"  
}  
{  
  "id": "3",  
  "Name": "Next Lab Lab 5.2",  
  "subscriber": "Chennai"  
}  
{
```

```
"id": "4",  
"Name": "Niranth",  
"subscriber": "Sbc"  
}
```

Customer.Ite... x

SELECT * FROM c [Edit Filter](#)

id	/subscriber
1	New York
2	Chicago
3	Chennai
4	Sbc

[Load more](#)

```
2 {  
3   "id": "4",  
4   "Name": "Niranth",  
5   "subscriber": "Sbc",  
6   "_rid": "gDt2AK3JPVsEAAAAAAAAA==",  
7   "_self": "dbs/gDt2AA==/colls/gDt2AK3JPVs=/docs/gDt2AK3JPVsEAAAAAAAAA==/",  
8   "_etag": "\"4801e302-0000-0100-0000-664a08ff0000\"",  
9   "_attachments": "attachments/",  
10  }
```

I/P O/P

Filter records with c.Name or c.id or c.subscriber

Input: select * from c where c.Name="Niranth"

Step 24: As soon as you'll click "Execute Query", All the records will be displayed as you scroll down the page.

Success output:

```
Customer.It... Customer.Que... x
1 SELECT * FROM c where c.Name="Niranth"
```

Results Query Stats

1 - 1

```
[
  {
    "id": "4",
    "Name": "Niranth",
    "subscriber": "Sbc",
    "_rid": "gDt2AK3JPVsEAAAAAAAAA==",
    "_self": "dbs/gDt2AA==/colls/gDt2AK3JPVs=/docs/gDt2AK3JPVsEAAAAAAAAA==/",
    "_etag": "\"4801e302-0000-0100-0000-664a08ff0000\"",
    "_attachments": "attachments/",
    "_ts": 1716127999
  }
]
```

The screenshot shows the NOSQL API interface. On the left, a tree view shows the database structure: NOSQL API > DATA > Deepika-dbid > Scale > Customer. The 'Customer' folder is selected. In the center, a query editor shows the query: `SELECT * FROM c`. Below the query editor, the 'Results' tab is active, showing '1 - 4' results. The results are displayed as a JSON array with four objects. The first object is highlighted with a yellow background. A blue arrow points from the '1 - 4' result indicator to the first object in the JSON array.

```
[
  {
    "id": "1",
    "Name": "ScholarhatUser1",
    "subscriber": "New York",
    "_rid": "gDt2AK3JPVsBAAAAAAAAA==",
    "_self": "dbs/gDt2AA==/colls/gDt2AK3JPVs=/docs/gDt2AK3JPVsBAAAAAAAAA==/",
    "_etag": "\"47015afa-0000-0100-0000-664a06c20000\"",
    "_attachments": "attachments/",
    "_ts": 1716127426
  },
  {
    "id": "2",
    "Name": "Deepika Prasad Lab 5.1",
    "subscriber": "Chicago",
    "_rid": "gDt2AK3JPVsCAAAAAAAAAA==",
    "_self": "dbs/gDt2AA==/colls/gDt2AK3JPVs=/docs/gDt2AK3JPVsCAAAAAAAAAA==/",
    "_etag": "\"4701aefe-0000-0100-0000-664a07fb0000\"",
    "_attachments": "attachments/",
    "_ts": 1716127739
  },
  {
    "id": "3",
    "Name": "Next Lab Lab 5.2",
    "subscriber": "Chennai",
    "_rid": "gDt2AK3JPVsDAAAAAAAAA==",
    "_self": "dbs/gDt2AA==/colls/gDt2AK3JPVs=/docs/gDt2AK3JPVsDAAAAAAAAA==/",
    "_etag": "\"48012301-0000-0100-0000-664a087e0000\"",
    "_attachments": "attachments/",
    "_ts": 1716127866
  }
]
```

-----***End of lab 5.1- Cosmos DB- SQL API***-----

3. Az204-lab 5.2-scholarhat-Creating a Cosmos DB account - Table API

Creating a Cosmos DB account - Table API

Level : Beginner

Mentor: Shailendra Chauhan

Type : GuidedLab

Points : 10

Duration : 00:50:00

[Lab Problem](#) [Lab Solution](#)

Lab Details

Description

In this lab, you will be learning how to create an Azure Cosmos DB Account and then work with the Table API.

Lab Objective

Upon completion of this lab, you will be able to:

- Create an Azure Cosmos DB Account.
- Create a new table within Cosmos DB Account.
- Add Default data into the table.

Prerequisites

You should be familiar with:

- Conceptual Understanding of Azure Cosmos DB.
- You must have the Azure Portal Access.

Microsoft Azure

Home > Subscription27 | Resource groups > rg0519 > Marketplace >

Create an Azure Cosmos DB account

Which API best suits your workload?

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Learn more](#)

To start, select the API to create a new account. The API selection cannot be changed after account creation.

Azure Cosmos DB for NoSQL

Azure Cosmos DB's core, or native API for working with documents. Supports fast, flexible development with familiar SQL query language and client libraries for .NET, JavaScript, Python, and Java.

[Create](#) [Learn more](#)

Azure Cosmos DB for PostgreSQL

Fully-managed relational database service for PostgreSQL with distributed query execution, powered by the Citus open source extension. Build new apps on single or multi-node clusters—with support for JSONB, geospatial, rich indexing, and high-performance scale-out.

[Create](#) [Learn more](#)

Azure Cosmos DB for MongoDB

Fully managed database service for apps written for MongoDB. Recommended if you have existing MongoDB workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Cassandra

Fully managed Cassandra database service for apps written for Apache Cassandra. Recommended if you have existing Cassandra workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Table

Fully-managed database service for apps written for Azure Table storage. Recommended if you have existing Azure Table storage workloads that you plan to migrate to Azure Cosmos DB.

[Create](#) [Learn more](#)

Azure Cosmos DB for Apache Gremlin

Fully managed graph database service using the Gremlin query language, based on Apache TinkerPop project. Recommended for new workloads that need to store relationships between data.

[Create](#) [Learn more](#)

Give Feedback

[Help improve this page](#)

Create a table api

Microsoft Azure

Search resources, services, and docs (G+)

Home > Subscription27 | Resource groups > rg0519 > Marketplace >

Create Azure Cosmos DB Account - Azure Cosmos DB for Table

Basics Global Distribution Networking Backup Policy Encryption Tags Review + create

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Try it for free](#), for 30 days with unlimited renewals. Go to production starting at \$24/month per database, multiple containers included. [Learn more](#)

Project Details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource Group * [Create new](#)

Instance Details

Account Name *

Configure availability zone settings for your account. You cannot change these settings once the account is created.

Availability Zones ☐ Enable ☒ Disable

Location *

Available locations are determined by your subscription's access and availability zone support (if that is enabled). If you don't see or cannot select your desired location, please open a support request for region access. [Click here for more details on how to create a region access request](#)

Capacity mode ☒ Provisioned throughput ☐ Serverless [Learn more about capacity mode](#)

With Azure Cosmos DB free tier, you will get the first 1000 RU/s and 25 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated \$64/month discount per account.

The subscription you have selected already has an account with free tier enabled.

Apply Free Tier Discount ☐ Apply ☒ Do Not Apply

Limit total account throughput ☐ Limit the total amount of throughput that can be provisioned on this account

i This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.

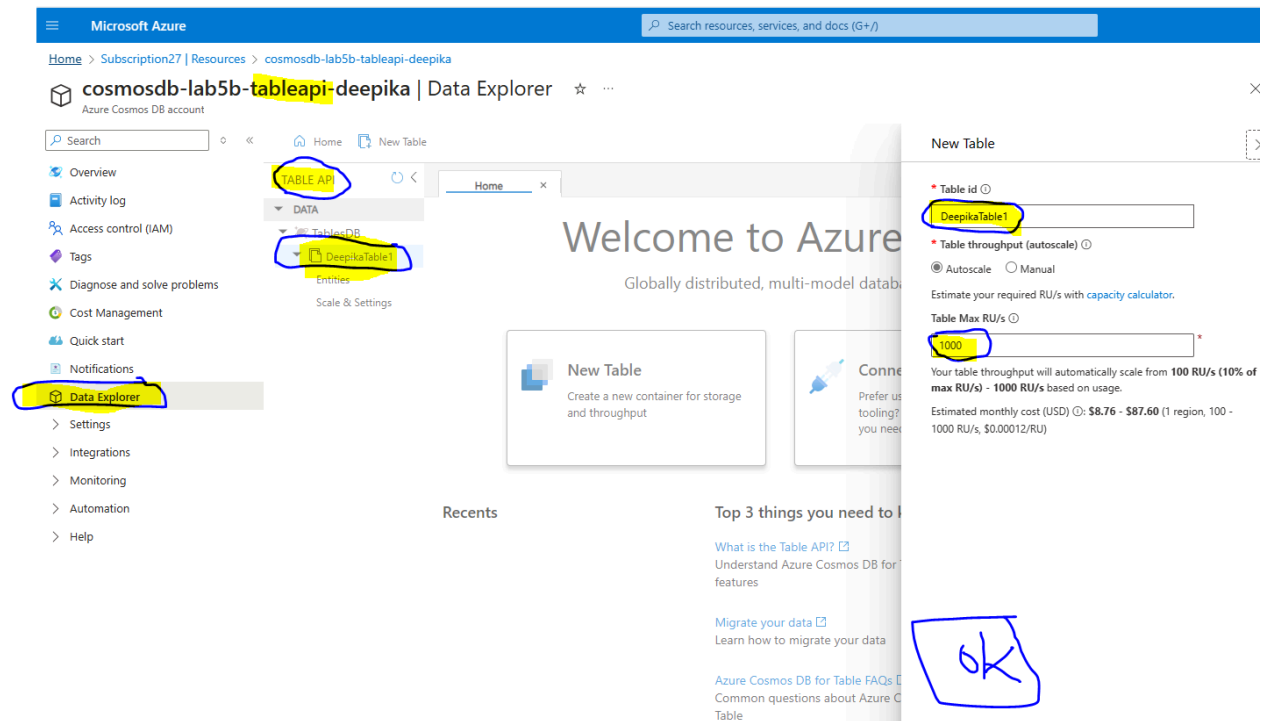
Review + create Previous Next: Global Distribution

Backup

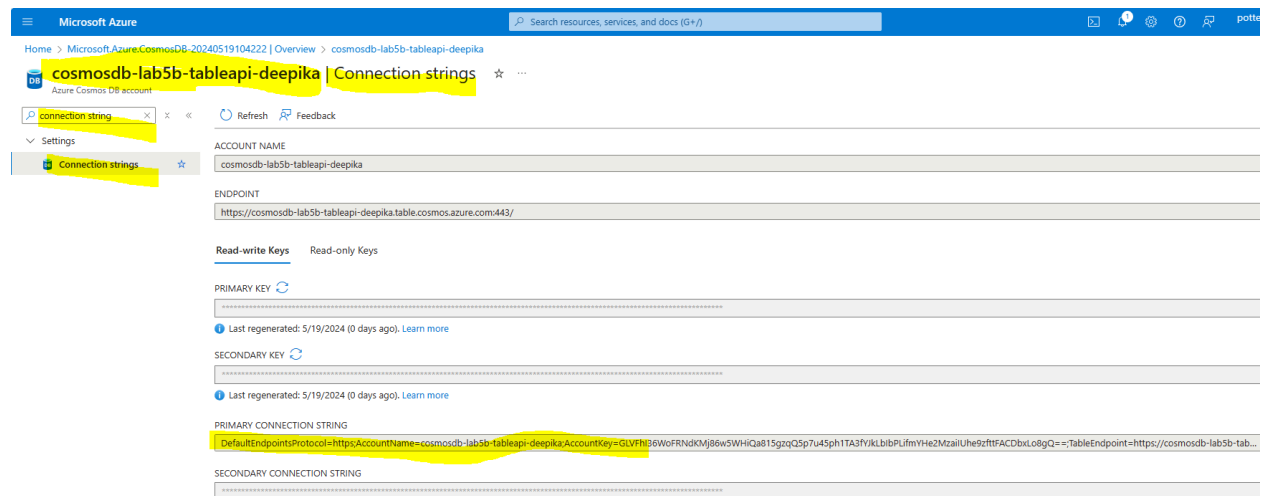
Backup storage redundancy *

- ☐ Geo-redundant backup storage
- ☐ Zone-redundant backup storage
- ☒ Locally-redundant backup storage

After Clicking on “Go to resource”, select “Data Explorer” from the left pane. Then, click on “New Table”.



On the left pane, type connection string and get the connection strings.



Step 11: Now, you are going to add default data to the table. For that, click on the “Open Full Screen” icon on the top right corner of the window.



Welcome to Azure Cosmos DB

Connect to your account with connection string

DefaultEndpointsProtocol=https;AccountName=c

Connect

Sign In with Azure Account

Microsoft Azure | Cosmos DB > cosmosdb-lab5b-tableapi-deepika

TABLE API | Home | Deepi...Entities x

TablesDB
DeepikaTable1

Property Name * Type Value

PartitionKey String DeepikaPrasad

RowKey * String 1

+ Add Property

No data available in table

Showing 0 to 0 of 0 entries

Add Entity

Microsoft Azure | Cosmos DB > cosmosdb-lab5b-tableapi-deepika

TABLE API | Home | Deepi...Entities x

TablesDB
DeepikaTable1

Property Name * Type Value

PartitionKey String DeepikaITTeam

RowKey * String 2

MemberName String Sammi

+ Add Property

PartitionKey	RowKey	Timestamp	MemberName
DeepikaITTeam	1	Sun, 19 May 2024 20:12:15 GMT	Vanitha
DeepikaITTeam	2	Sun, 19 May 2024 20:12:38 GMT	Rajan

Results 1 - 2 of 2

Add Entity

Microsoft Azure

Cosmos DB > cosmosdb-lab5b-tableapi-deepika

Home

Deepi...Entities x

TABLE API

TablesDB

DeepikaTable1

Entities

Scale & Settings

Action

And/Or

Field

Type

Operator

Value

+ Add new clause

Advanced Options

PartitionKey ^	RowKey ^	Timestamp ^	MemberName ^
DeepikaITTeam	1	Sun, 19 May 2024 20:12:15 GMT	Vanitha
DeepikaITTeam	2	Sun, 19 May 2024 20:12:38 GMT	Rajan
DeepikaITTeam	3	Sun, 19 May 2024 20:15:04 GMT	Sammi
DeepikaITTeam	4	Sun, 19 May 2024 20:15:15 GMT	Niranth
DeepikaITTeam	5	Sun, 19 May 2024 20:15:27 GMT	Shailaja

-----****End of lab 5.2****-----