Lab Manual- Setup and Manage Azure CosmoDB

Prepared for:

Date: 18th Nov 2021

Prepared by:

Document Name: Lab Manual

Document Number AZLab982

Contributor:

Contents

1. Introduction

Azure Cosmos DB is a Microsoft Azure database service, fastest-growing Azure service that is available in all Azure regions, that fits any web application, mobile application, gaming or IoT application that requires processing, reading and writing a massive amount of data.

Azure Cosmos DB provides the applications with:

- The **least possible low latency**, less than 10-ms latencies for both reads and writes workloads due to using SSD backed storage and the multi-master replication for your data wherever your users are, allowing the users to connect to the closest replica to them
- The **highest possible availability**, up to 99.999% high availability of the data stored in the Cosmos DB due to the ability to perform regional failover for your Cosmos DB account databases
- The **highest possible throughput**
- **Variant consistency options**, with the five well-defined consistency options, include strong, bounded staleness, session, consistent prefix, and eventual that provides full flexibility and low cost-to-performance ratio
- **Advanced security options**, where the Cosmos DB data will be encrypted at rest and in motion, in addition to the row-level authorization
- **Elastically and globally scalable** for the read and write operations, that helps in dealing with unexpected workload spikes, with the ability to modify the current structure anytime by adding or remove any of the Azure regions to your Cosmos account
- Support for **multiple APIs** to work with your data that is stored in the Cosmos database, such as the SQL Core API, Cassandra, MongoDB, Gremlin and Azure Table Storage, providing you with an easy way to migrate the application to Cosmos DB without the need to perform significant changes. It requires no schema or index management is, as all data will be automatically indexed, and requires no complex multi-datacenter management or database software deployments or upgrade

For now, we are familiar with the Azure Cosmos DB concept and why we need to use it to host the data used by the critical applications. Let us now see how to create a new database account.

2. Relational vs Non- relational

When working in the database world, you need to distinguish between the relational and non-relational database engines.

In relational database engines, such as Microsoft SQL Server, MySQL, PostgreSQL, and SQLite3, the data will be stored in tables in the shape of columns and rows, with the ability to link data from different tables using foreign key constraints. Besides, the schema of the relational database tables, which defines the name of the columns, the type of data that can be stored in these columns and the constraints on the table columns, should be defined.

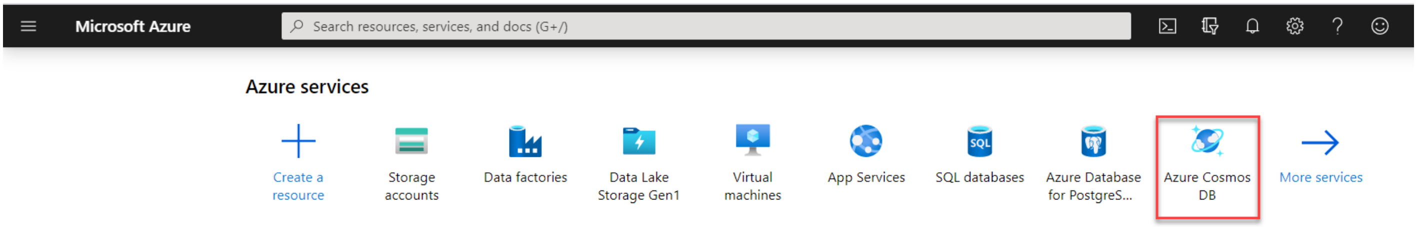
In the non-relational database engines, also called NoSQL database engines, such as MongoDB, DocumentDB, Cassandra, Coachbase and HBase, the data will **NOT** be stored in tables with defined schemas. Instead, the data will be stored in an optimized storage model, such as Key-value stores, Graph stores, stores, and Document stores.

3. Create an Azure Cosmos DB Account

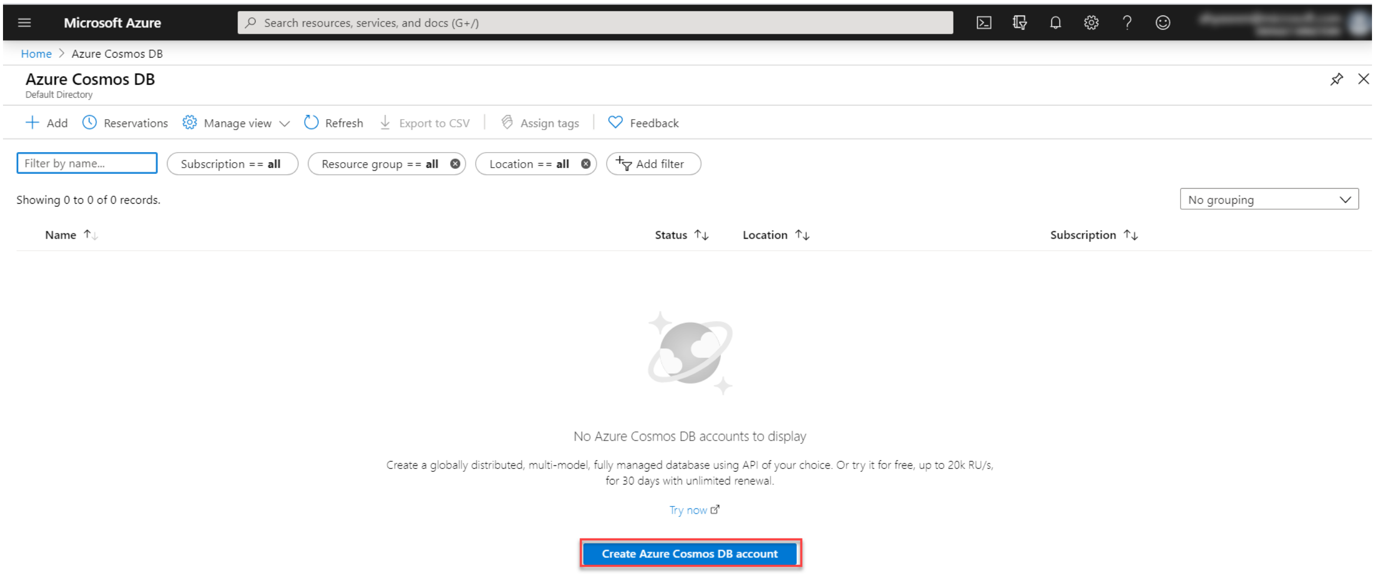
In order to create an Azure Cosmos DB account, you need to have an Azure subscription or create a free Azure Cosmos DB trial account.

When your subscription or free account is ready, open the Azure portal site, https://portal.azure.com/, in your Internet browser and login using the subscription account.

From the Microsoft Azure portal, search for the Azure Cosmos DB server, or click on the icon directly if you can detect it, as below:



In the opened page, click on the **Create Azure Cosmos DB account** option, to create a new Cosmos DB account, as shown below:

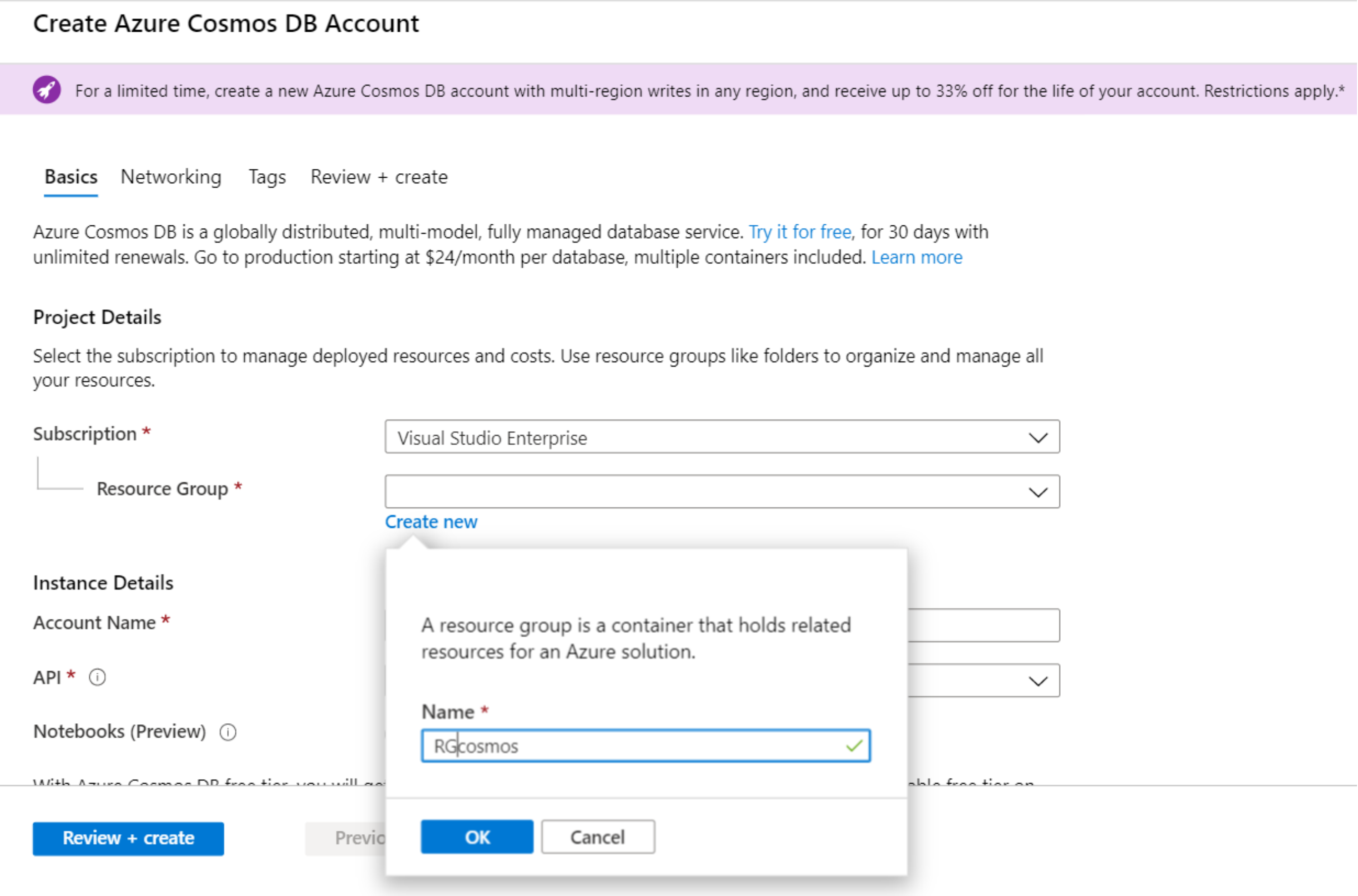


Showing 0 to 0 of 0 records.

No Azure Cosmos DB accounts to display

Create a globally distributed, multi-model, fully managed database using API of your choice. Or try it for free, up to 20k RU/s, for 30 days with unlimited renewal.

From the Create Azure Cosmos DB Account page, select the Azure subscription under which you plan to create the Cosmos DB account and the resource group, the container of resources for that Azure service that is used to manage all resources, that will be used for the Cosmos DB account. You can use an existing resource group or create a new resource group, as shown below:



Create Azure Cosmos DB Account

For a limited time, create a new Azure Cosmos DB account with multi-region writes in any region, and receive up to 33% off for the life of your account. Restrictions apply.*

Azure Cosmos DB is a globally distributed, multi-model, fully managed database service. 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 * Visual Studio Enterprise

Resource Group *

Instance Details

Account Name *

API *

Notebooks (Preview)

With Azure Cosmos DB free tier, you will ge … able free tier on

A resource group is a container that holds related resources for an Azure solution.

Name * RGcosmos

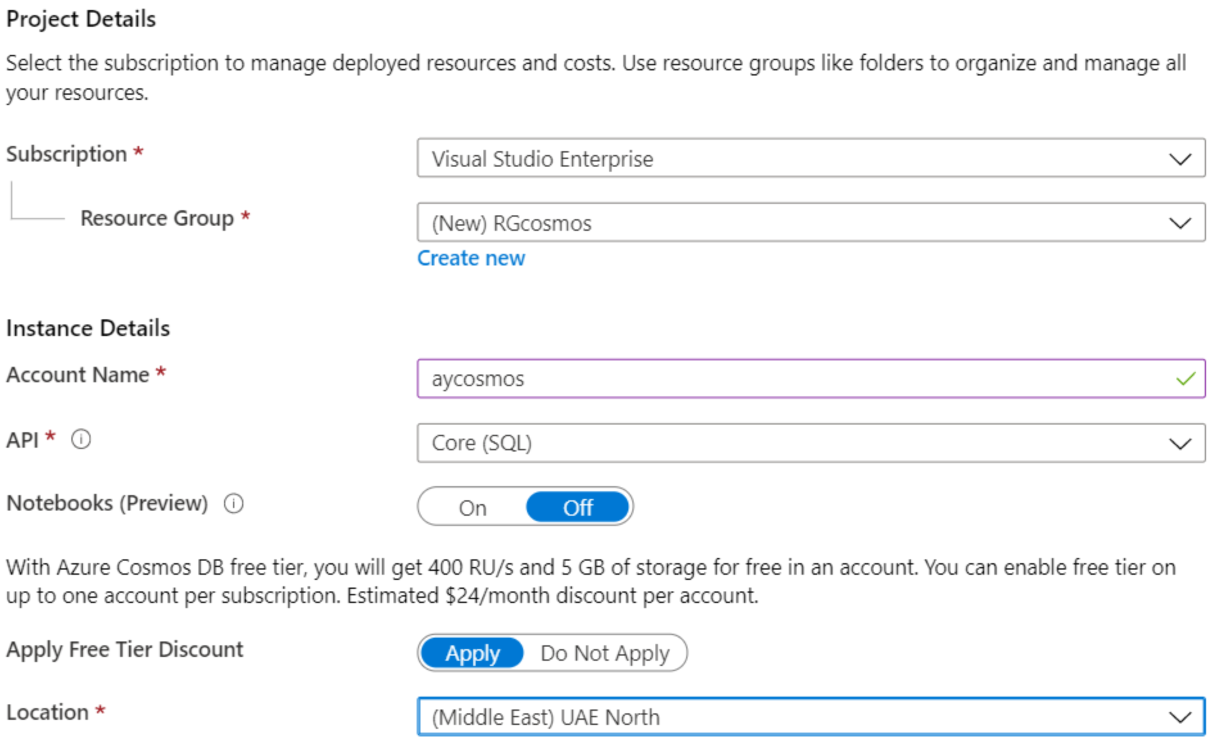
After that, you will be requested to provide all details for the Cosmos DB instance that will be created include a unique **Account name** that will be used to identify the Azure Cosmos account. You can use only lowercase letters, numbers, and the hyphen character to in the account name.

The **API** is the type of database account that will be created. You can choose from the previously mentioned five API supported by the Cosmos DB. In this demo, we will use the Core (SQL) API.

You need also to provide the **Location**, which describes the nearest datacenter to your current location in order to provide the fastest access to the data.

You can also enable the **Multi-region write** option to replicate your data to multiple selected regions. For now, we can create it in one region and add more regions later.

After providing your selections click on **Review + Create** option as below:



Project Details

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

Subscription * Visual Studio Enterprise

Resource Group * (New) RGcosmos

Create new

Instance Details

Account Name * aycosmos ✓

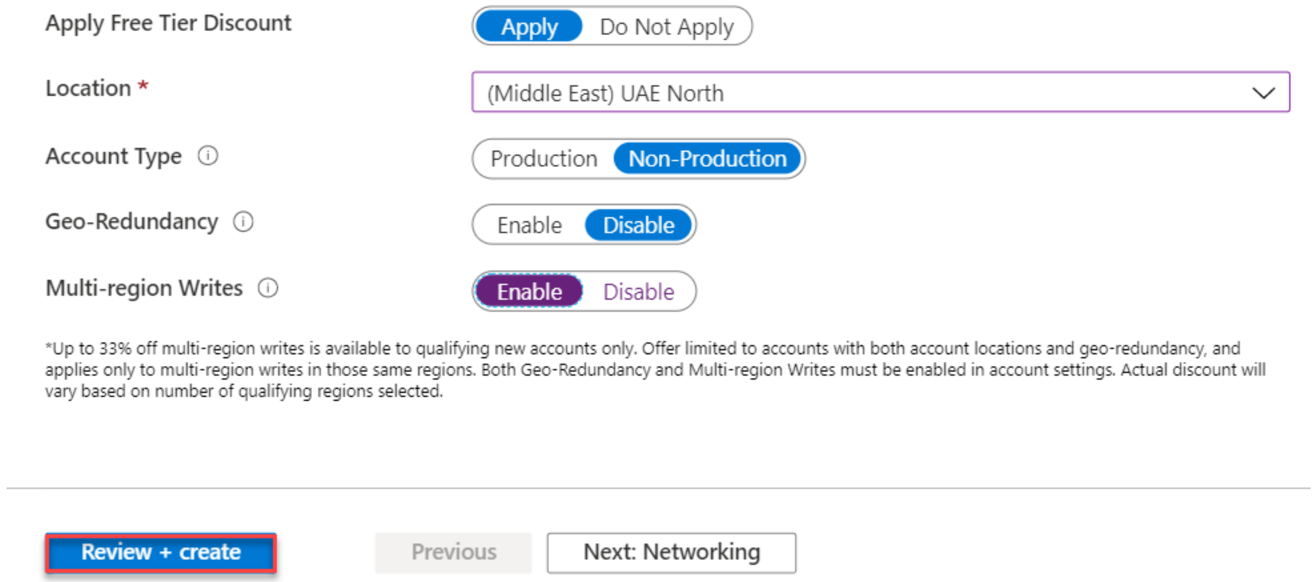
API * ⓘ Core (SQL)

Notebooks (Preview) ⓘ On [✓ Off]

With Azure Cosmos DB free tier, you will get 400 RU/s and 5 GB of storage for free in an account. You can enable free tier on up to one account per subscription. Estimated $24/month discount per account.

Apply Free Tier Discount [✓ Apply] Do Not Apply

Location * (Middle East) UAE North



Apply Free Tier Discount [✓ Apply] Do Not Apply

Location * (Middle East) UAE North

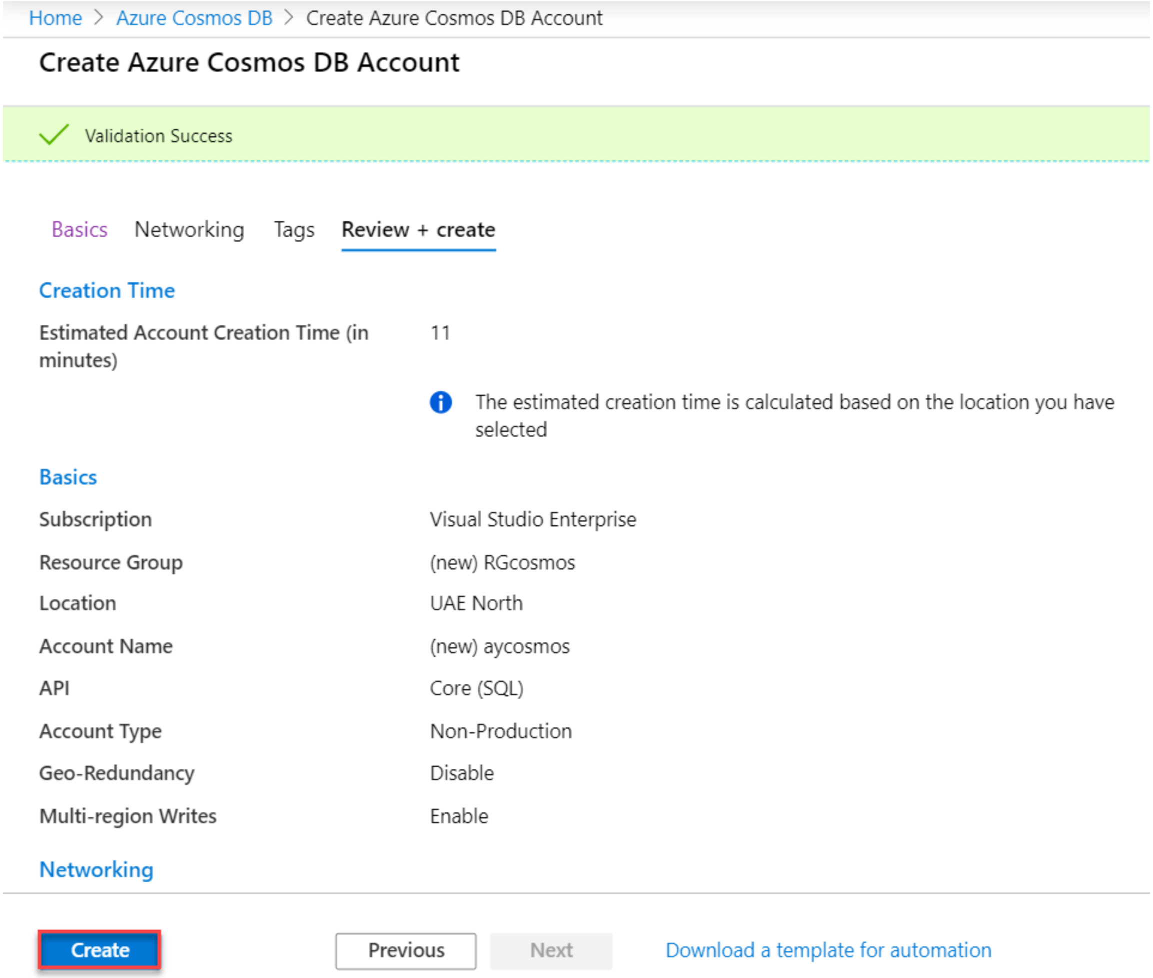
Account Type ⓘ Production [✓ Non-Production]

Geo-Redundancy ⓘ Enable [✓ Disable]

Multi-region Writes ⓘ [✓ Enable] Disable

*Up to 33% off multi-region writes is available to qualifying new accounts only. Offer limited to accounts with both account locations and geo-redundancy, and applies only to multi-region writes in those same regions. Both Geo-Redundancy and Multi-region Writes must be enabled in account settings. Actual discount will vary based on number of qualifying regions selected.

In the Review + Create page, review all the selections then click on the **Create** option to create your Azure Cosmos DB account, as shown below:


Home > Azure Cosmos DB > Create Azure Cosmos DB Account

Create Azure Cosmos DB Account

✓ Validation Success

Basics Networking Tags Review + create

Creation Time

Estimated Account Creation Time (in minutes)	11

The estimated creation time is calculated based on the location you have selected

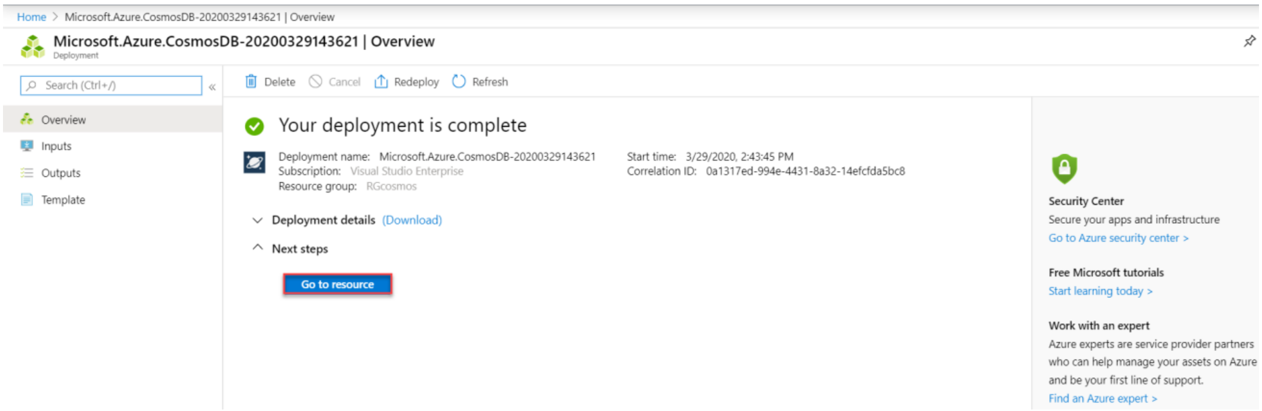
Basics

Subscription	Visual Studio Enterprise
Resource Group	(new) RGcosmos
Location	UAE North
Account Name	(new) aycosmos
API	Core (SQL)
Account Type	Non-Production
Geo-Redundancy	Disable
Multi-region Writes	Enable

Networking

After a few minutes, the database account will be created and deployed completely.

Click on the **Go to resource** option to start working in the database account, as shown below:



Microsoft.Azure.CosmosDB-20200329143621 | Overview

Deployment

Your deployment is complete

Deployment name: Microsoft.Azure.CosmosDB-20200329143621
Subscription: Visual Studio Enterprise
Resource group: RGcosmos

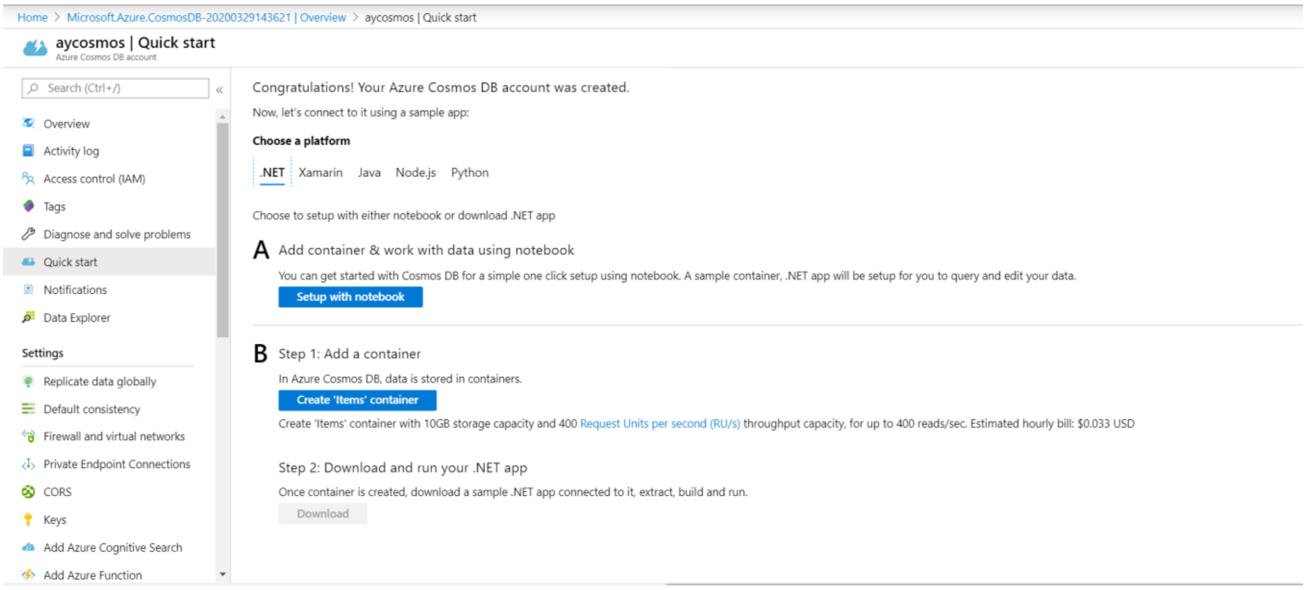
Start time: 3/29/2020, 2:43:45 PM
Correlation ID: 0a1317ed-994e-4431-8a32-14efcfda5bc8

Security Center
Secure your apps and infrastructure
Go to Azure security center >

Free Microsoft tutorials
Start learning today >

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
Find an Azure expert >

The first opened page is the Quickstart page that allows you to select from the available platforms and the next step to proceed with the Azure Cosmos DB account. Click on the Overview page from the below:



Congratulations! Your Azure Cosmos DB account was created.

Now, let's connect to it using a sample app:

Choose a platform

.NET Xamarin Java Node.js Python

Choose to setup with either notebook or download .NET app

A Add container & work with data using notebook

You can get started with Cosmos DB for a simple one click setup using notebook. A sample container, .NET app will be setup for you to query and edit your data.

Setup with notebook

B Step 1: Add a container

In Azure Cosmos DB, data is stored in containers.

Create 'Items' container

Create 'Items' container with 10GB storage capacity and 400 Request Units per second (RU/s) throughput capacity, for up to 400 reads/sec. Estimated hourly bill: $0.033 USD

Step 2: Download and run your .NET app

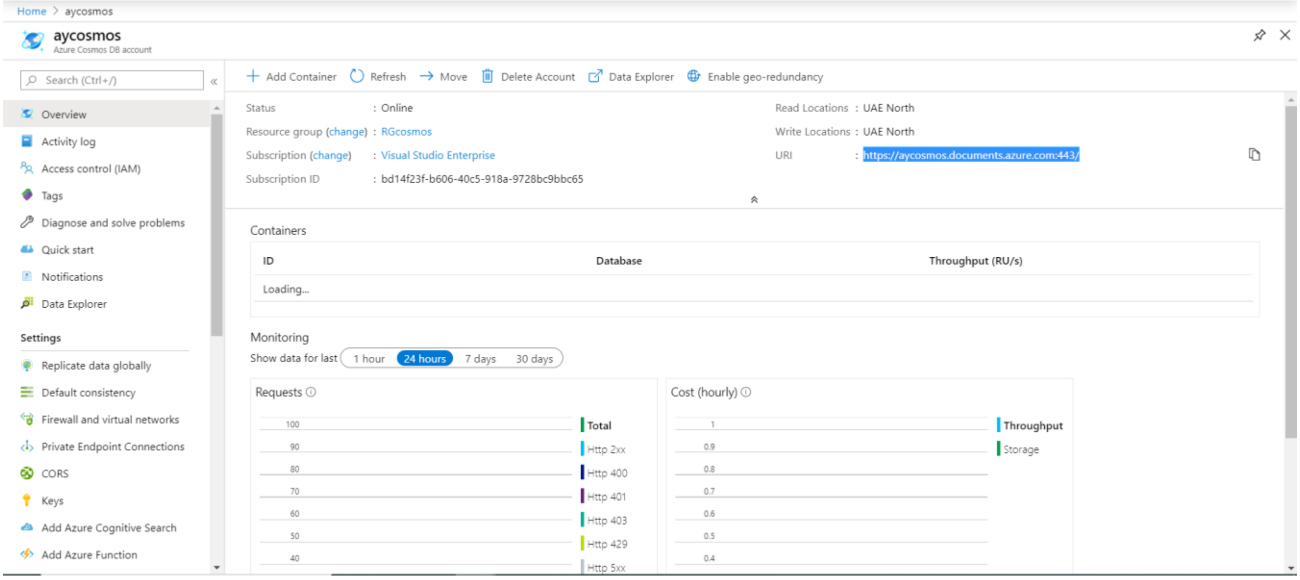
Once container is created, download a sample .NET app connected to it, extract, build and run.

Download

The Overview page shows the general information about the database account, the list of containers available under the database account and finally the monitoring dashboard for the Azure Cosmos DB account.

You can see the URL that is used to connect to the database account will be like https://**aycosmos**.documents.azure.com:443/

As it is clear below:



Status : Online

Resource group (change) : RGcosmos

Subscription (change) : Visual Studio Enterprise

Subscription ID : bd14f23f-b606-40c5-918a-9728bc9bbc65

Read Locations : UAE North

Write Locations : UAE North

URI : https://aycosmos.documents.azure.com:443/

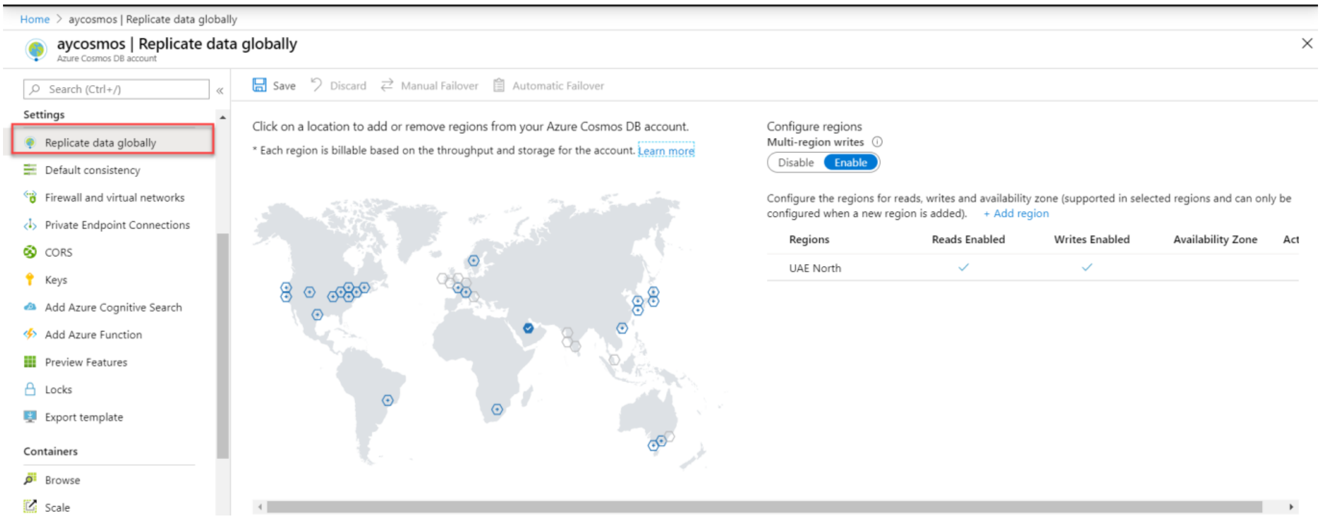
Containers

ID	Database	Throughput (RU/s)
Loading...		

Monitoring

Show data for last 1 hour 24 hours 7 days 30 days

If you enable the Multi-region option, you can easily move to the **Replicate data globally** page and select the regions that you plan to replicate your data to, in order to provide faster access to the data to your global customers, as below:



Home > aycosmos | Replicate data globally

aycosmos | Replicate data globally

Azure Cosmos DB account

Click on a location to add or remove regions from your Azure Cosmos DB account.

* Each region is billable based on the throughput and storage for the account. Learn more

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

Regions	Reads Enabled	Writes Enabled	Availability Zone	Act
UAE North	✓	✓		

4. Create New Container

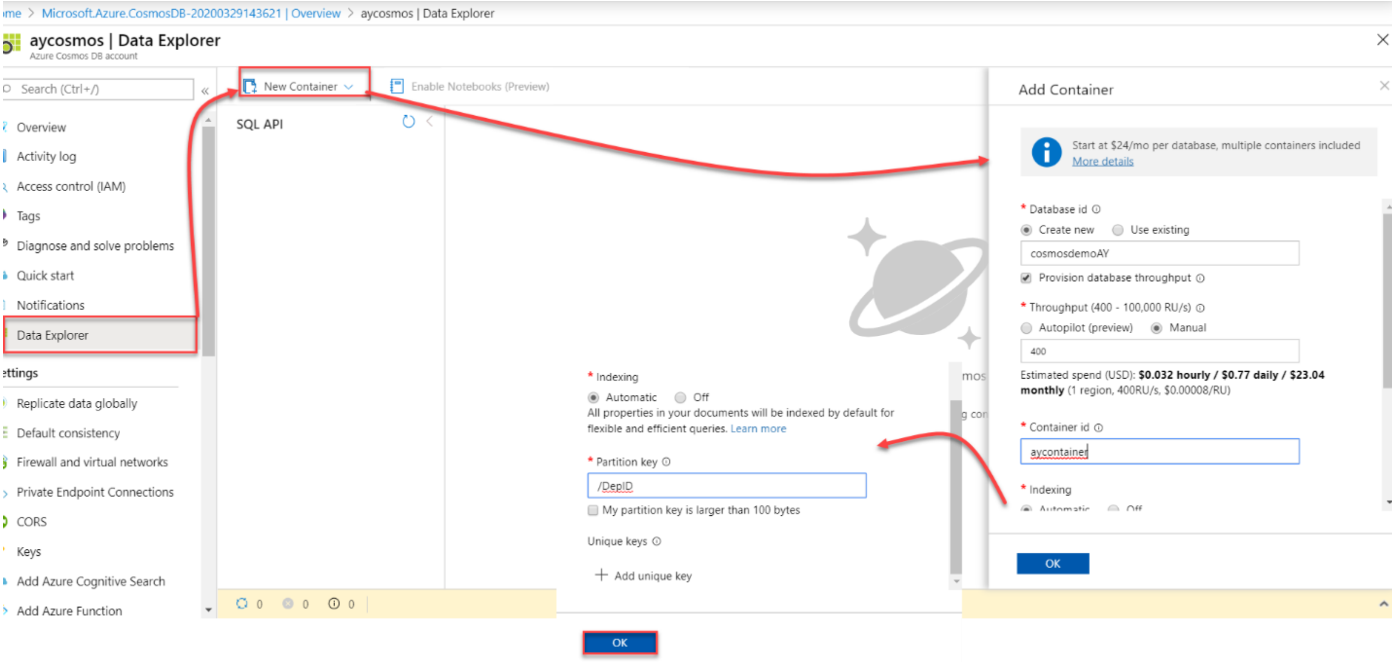
Now the Azure Cosmos DB account is ready, and we can create a database under that account and add data containers.

To achieve that, move to the **Data Explorer** page and click on the **New Container** option.

In the Add Container page, provide the name of the database under which the container will be created, with the ability to create more than one container per database, or select an existing database if available, with the ability to share the throughput provisioned to the database across all the containers within the database, to save the cost.

In this demo, we will keep the throughput value at 400 request units per second (RU/s).

Also, provide a unique name for that container, whether to automatically index all data to gain the best data access performance and the partition key that will be used for dividing the data into logical partitions within the same container, then click **OK** to create the database and the container as shown below:



ıme > Microsoft.Azure.CosmosDB-20200329143621 | Overview > aycosmos | Data Explorer

aycosmos | Data Explorer

Azure Cosmos DB account

Add Container

Start at $24/mo per database, multiple containers included More details

* Database id: ● Create new ○ Use existing

cosmosdemoAY

☑ Provision database throughput

* Throughput (400 - 100,000 RU/s): ○ Autopilot (preview) ● Manual

400

Estimated spend (USD): **$0.032 hourly / $0.77 daily / $23.04 monthly** (1 region, 400RU/s, $0.00008/RU)

* Container id

aycontainer

* Indexing

● Automatic ○ Off

All properties in your documents will be indexed by default for flexible and efficient queries. Learn more

* Partition key

/DepID

□ My partition key is larger than 100 bytes

Unique keys

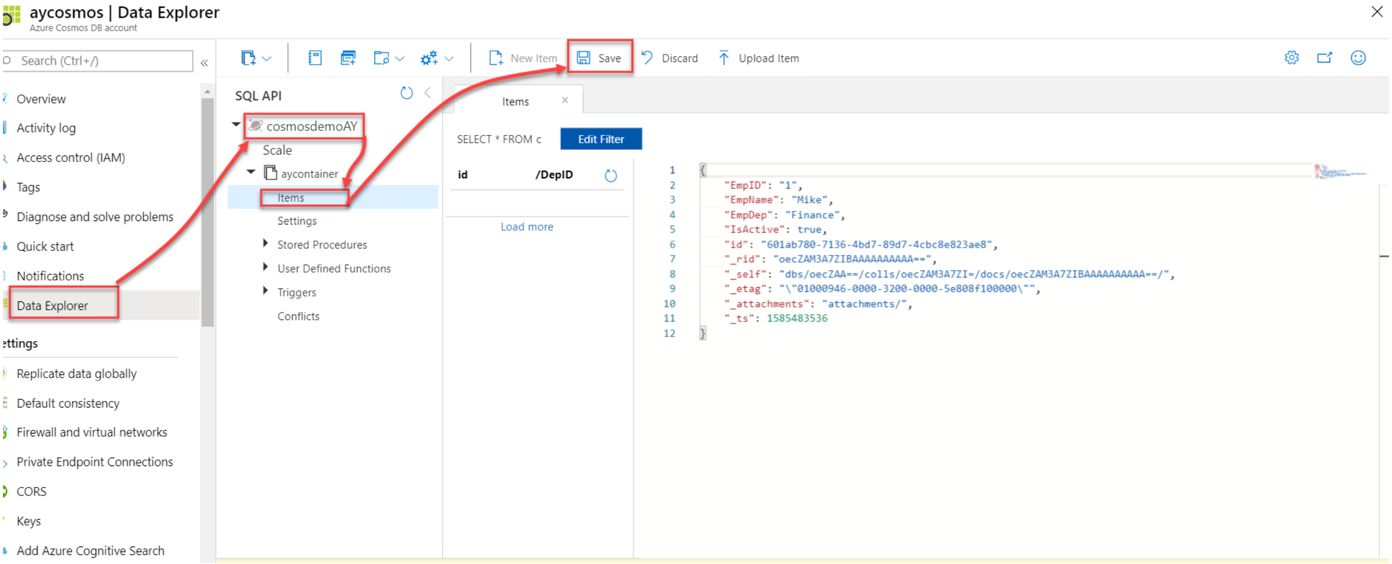
+ Add unique key

OK

1. Add New Item

To add a new data item to your container, move to the Data Explorer page, expand the database and browse for the container under which you want to add the data.

Under your container, click on Items then choose **New Item** option to add a new employee data. After adding the EmpID, EmpName, EmpDep and IsActive columns values, click on **Save**, where new six metadata fields will be added to the item, as shown below:

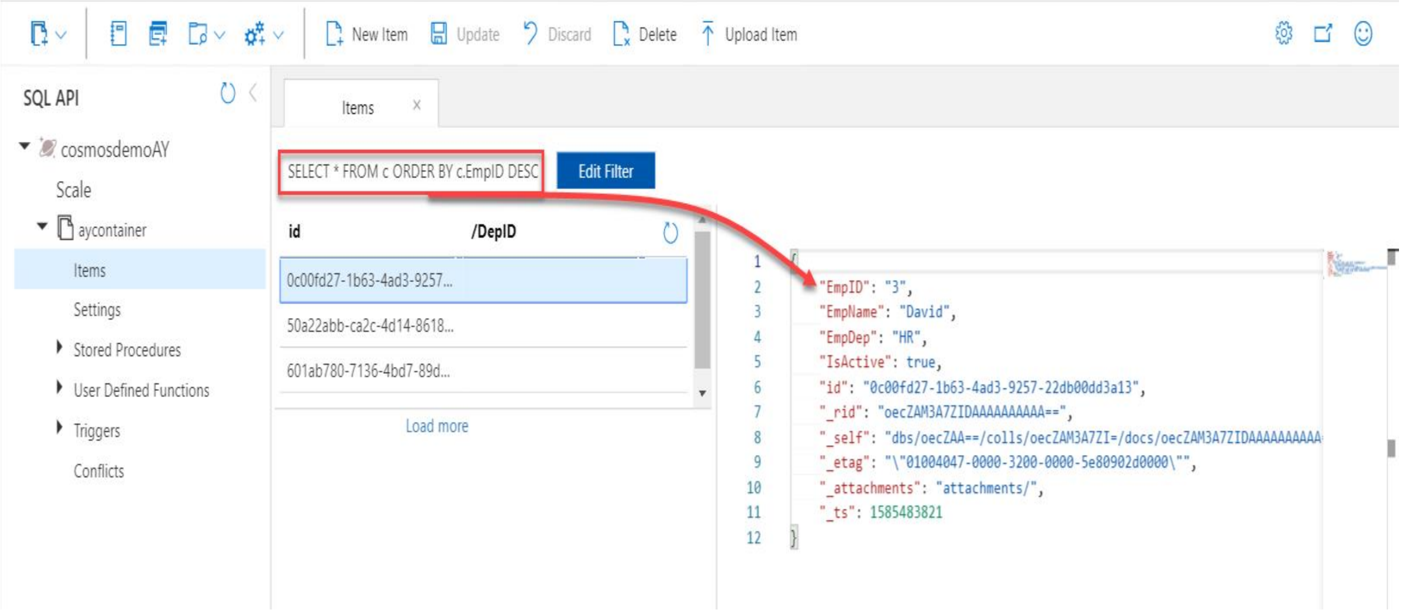


```
{
    "EmpID": "1",
    "EmpName": "Mike",
    "EmpDep": "Finance",
    "IsActive": true,
    "id": "601ab780-7136-4bd7-89d7-4cbc8e823ae8",
    "_rid": "oecZAM3A7ZIBAAAAAAAAAA==",
    "_self": "dbs/oecZAA==/colls/oecZAM3A7ZI=/docs/oecZAM3A7ZIBAAAAAAAAAA==/",
    "_etag": "\"01000946-0000-3200-0000-5e808f100000\"",
    "_attachments": "attachments/",
    "_ts": 1585483536
}
```

In this demo, we will repeat this process three times to add new items.

2. Query Data

To query the inserted data, you can easily click on the **Edit Filter** option in the Items page, where you can edit the existing query by adding a filter or order the data in a specific order. In this example, we will sort the items based on the employee ID, as shown below:

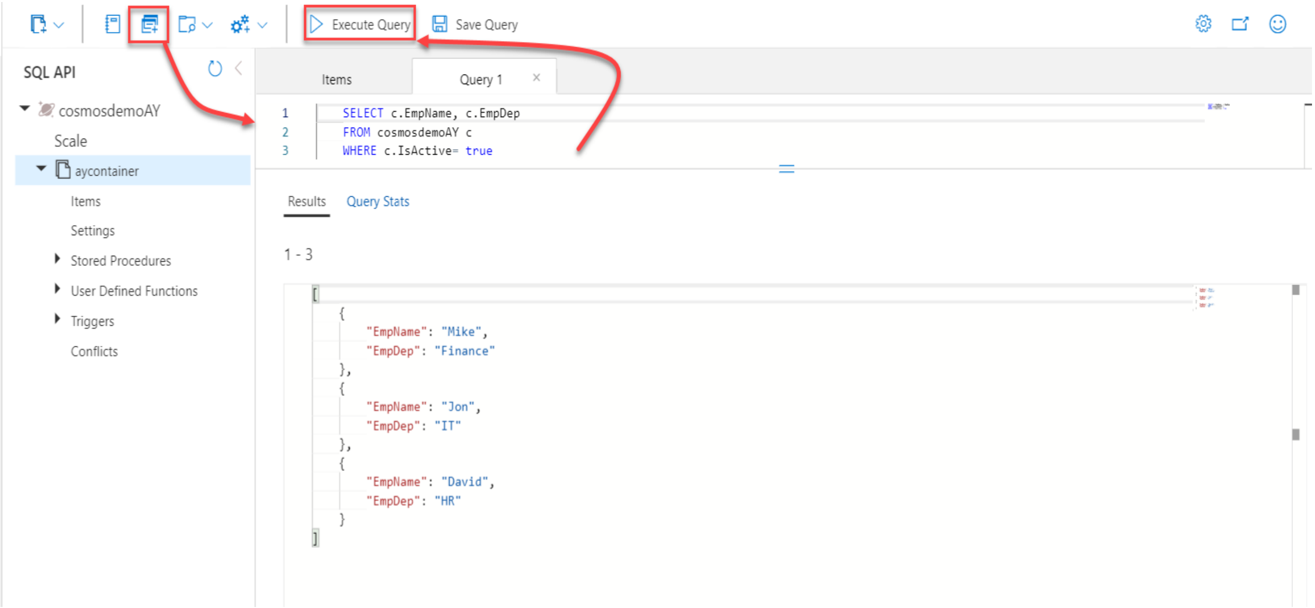


```
SELECT * FROM c ORDER BY c.EmpID DESC
```

```
{
    "EmpID": "3",
    "EmpName": "David",
    "EmpDep": "HR",
    "IsActive": true,
    "id": "0c00fd27-1b63-4ad3-9257-22db00dd3a13",
    "_rid": "oecZAM3A7ZIDAAAAAAAAAA==",
    "_self": "dbs/oecZAA==/colls/oecZAM3A7ZI=/docs/oecZAM3A7ZIDAAAAAAAAAA
    "_etag": "\"01004047-0000-3200-0000-5e80902d0000\"",
    "_attachments": "attachments/",
    "_ts": 1585483821
}
```

A more flexible option is to open a query window and write your SQL query to select specific columns and retrieve the items that meet the provided condition.

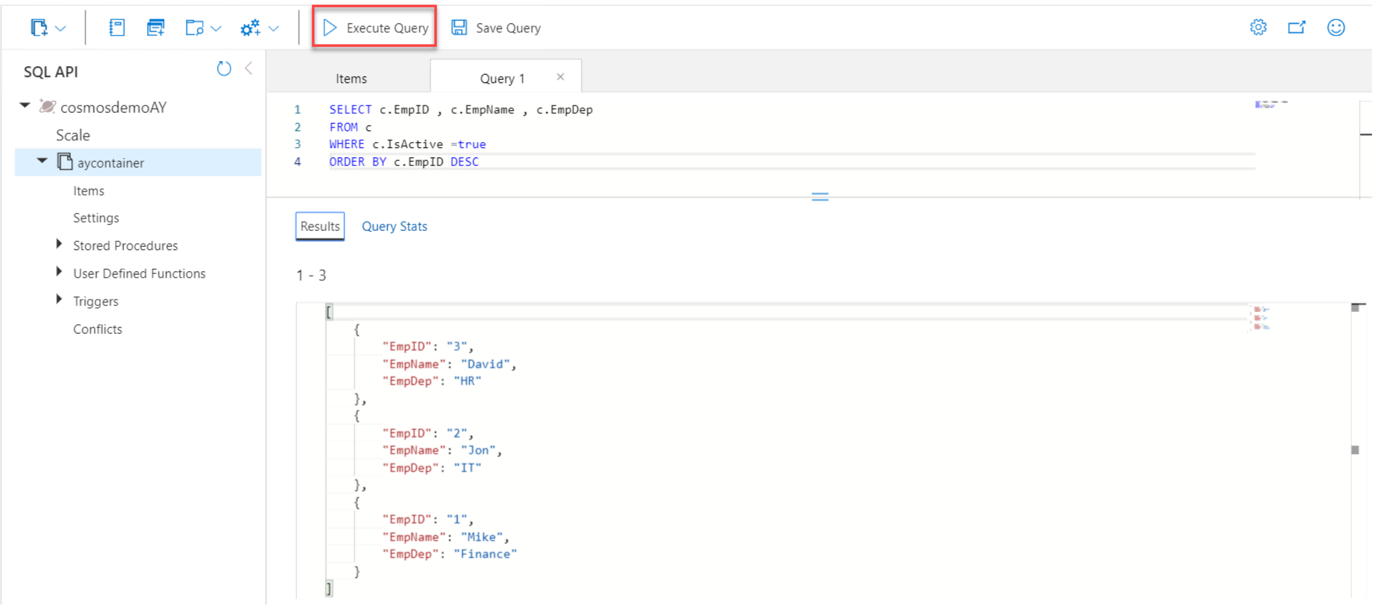
In the below example, we will use the SQL query below to retrieve the name and department for the active employees. After writing the query, click on **Execute Query** button, and the result will be displayed, as shown below:



```
SELECT c.EmpName, c.EmpDep
FROM cosmosdemoAY c
WHERE c.IsActive= true
```

```
[
    {
        "EmpName": "Mike",
        "EmpDep": "Finance"
    },
    {
        "EmpName": "Jon",
        "EmpDep": "IT"
    },
    {
        "EmpName": "David",
        "EmpDep": "HR"
    }
]
```

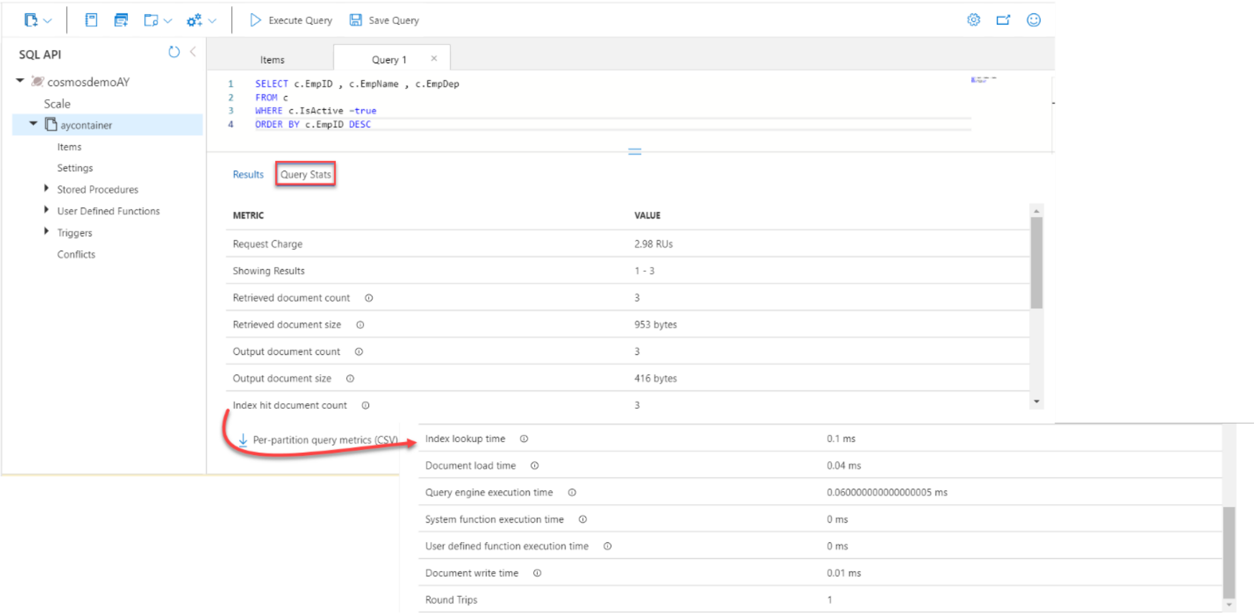
You can use your SQL development skills to write more complex queries, such as selecting specific columns, provide a condition to filter the data in WHERE clause and sort the data using the ORDER BY clause, as shown below:



```
SELECT c.EmpID , c.EmpName , c.EmpDep
FROM c
WHERE c.IsActive =true
ORDER BY c.EmpID DESC
```

```
[
    {
        "EmpID": "3",
        "EmpName": "David",
        "EmpDep": "HR"
    },
    {
        "EmpID": "2",
        "EmpName": "Jon",
        "EmpDep": "IT"
    },
    {
        "EmpID": "1",
        "EmpName": "Mike",
        "EmpDep": "Finance"
    }
]
```

Besides the result of the executed query, you can review the execution performance statistics of that query, such as the number of RU consumed, storage and time, that helps in comparing the different queries and enhance the queries performance, as shown below:



```
SELECT c.EmpID , c.EmpName , c.EmpDep
FROM c
WHERE c.IsActive -true
ORDER BY c.EmpID DESC
```

METRIC	VALUE
Request Charge	2.98 RUs
Showing Results	1 - 3
Retrieved document count	3
Retrieved document size	953 bytes
Output document count	3
Output document size	416 bytes
Index hit document count	3
Index lookup time	0.1 ms
Document load time	0.04 ms
Query engine execution time	0.060000000000000005 ms
System function execution time	0 ms
User defined function execution time	0 ms
Document write time	0.01 ms
Round Trips	1

5. Conclusion

It is clear from the previous examples how easy we can configure and use the Azure Cosmos DB to store and query the data and take advantage of the great performance provided by it in processing and storing your data