

Cosmos DB

console application in C#

(quickStart)

1. Login in the Azure portal and select **Azure CosmosDB**

Microsoft Azure

Search resources, services, and docs (G+)

Azure services

- Create a resource
- Azure Cosmos DB
- Resource groups
- All resources
- Cost Management ...
- Storage accounts
- Azure Databricks
- Azure Active Directory
- Virtual machines
- More services

Resources

Recent Favorite

Name	Type	Last Viewed
luiscococosmosdb	Azure Cosmos DB account	23 minutes ago
cosmosdbRG	Resource group	25 minutes ago
Azure subscription 1	Subscription	7 days ago

[See all](#)

Navigate

- Subscriptions
- Resource groups
- All resources
- Dashboard

2. Press the “Create” button in the Azure CosmosDB

The screenshot shows the Azure portal interface for an Azure Cosmos DB resource. The left sidebar contains navigation options: 'Create a resource', 'Home', 'Dashboard', 'All services', and a 'FAVORITES' section with 'All resources', 'Resource groups', and 'Quickstart Center'. The main content area is titled 'Azure Cosmos DB' and shows the resource 'particular (luiscocoenriquezhotmail.onmicrosoft.com)'. Below the title are action buttons: '+ Create', 'Restore', 'Manage view', 'Refresh', 'Export to CSV', 'Open query', and 'Assign tags'. A filter bar is present with a search input and several filter buttons: 'Subscription equals all', 'Type equals all', 'Resource group equals all', and 'Location equals all'. Below the filters, it says 'Showing 1 to 1 of 1 records.' and provides options for 'No grouping' and 'List view'. A table displays the resource details:

<input type="checkbox"/> Name ↑↓	Status ↑↓	Subscription ↑↓	Write region ↑↓	Read Region ↑↓	
<input type="checkbox"/> luisccocosmosdb	Online	Azure subscription 1	France Central	France Central	...

3. Press the “Create” button in the Azure CosmosDB for NoSQL

The screenshot shows the Azure portal interface for creating a new Azure Cosmos DB account. The browser address bar displays `portal.azure.com/#create/Microsoft.DocumentDB`. The left-hand navigation pane includes links to 'Create a resource', 'Home', 'Dashboard', 'All services', and a 'FAVORITES' section with various resource categories. The main content area is titled 'Create an Azure Cosmos DB account' and features the heading 'Which API best suits your workload?'. Below this heading, a descriptive paragraph states: 'Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. [Learn more](#)'. A sub-note reads: 'To start, select the API to create a new account. The API selection cannot be changed after account creation.' Six API options are presented in a grid, each with a description and a 'Create' button:

- 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.
- 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.
- 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.
- 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.
- 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.
- 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.

4. Set the Subscription, the resource group, the account name, the location, select the capacity mode, apply the free tier discount, limit total account throughput and press the button “Review + create”

The screenshot shows the 'Create Azure Cosmos DB Account' page in the Azure portal. The page is titled 'Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL'. The left sidebar shows the 'Create a resource' button and a list of services. The main content area contains the following fields and options:

- Subscription ***: Azure subscription 1
- Resource Group ***: cosmosdbRG (with a 'Create new' link)
- Instance Details**
 - Account Name ***: cosmosdbluiscoenriquez (with a green checkmark)
 - Location ***: (Europe) France Central
 - Capacity mode**: ☒ Provisioned throughput ☐ Serverless (with a 'Learn more about capacity mode' link)
- Free Tier Discount**: ☐ Apply ☒ Do Not Apply
- Limit total account throughput**: ☒ Limit the total amount of throughput that can be provisioned on this account (with an information icon and text: 'This limit will prevent unexpected charges related to provisioned throughput. You can update or remove this limit after your account is created.'

At the bottom, there are three buttons: 'Review + create' (blue), 'Previous' (grey), and 'Next: Global Distribution' (white with a border).

5. Press the CosmosDB link “luiscococosmosdb” to navigate to the **Azure CosmosDB account**

The screenshot shows the Azure portal interface. The left sidebar contains navigation links: 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES', 'All resources', 'Resource groups', and 'Quickstart Center'. The main content area displays the 'Azure Cosmos DB' resource page for the account 'luiscococosmosdb'. The page includes a search bar, a filter bar with 'Subscription equals all', 'Type equals all', 'Resource group equals all', and 'Location equals all', and a table of resources. The table shows one record for 'luiscococosmosdb' with status 'Online', subscription 'Azure subscription 1', and location 'France Central'.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

Azure Cosmos DB

particular (luiscocoenriquezhotmail.onmicrosoft.com)

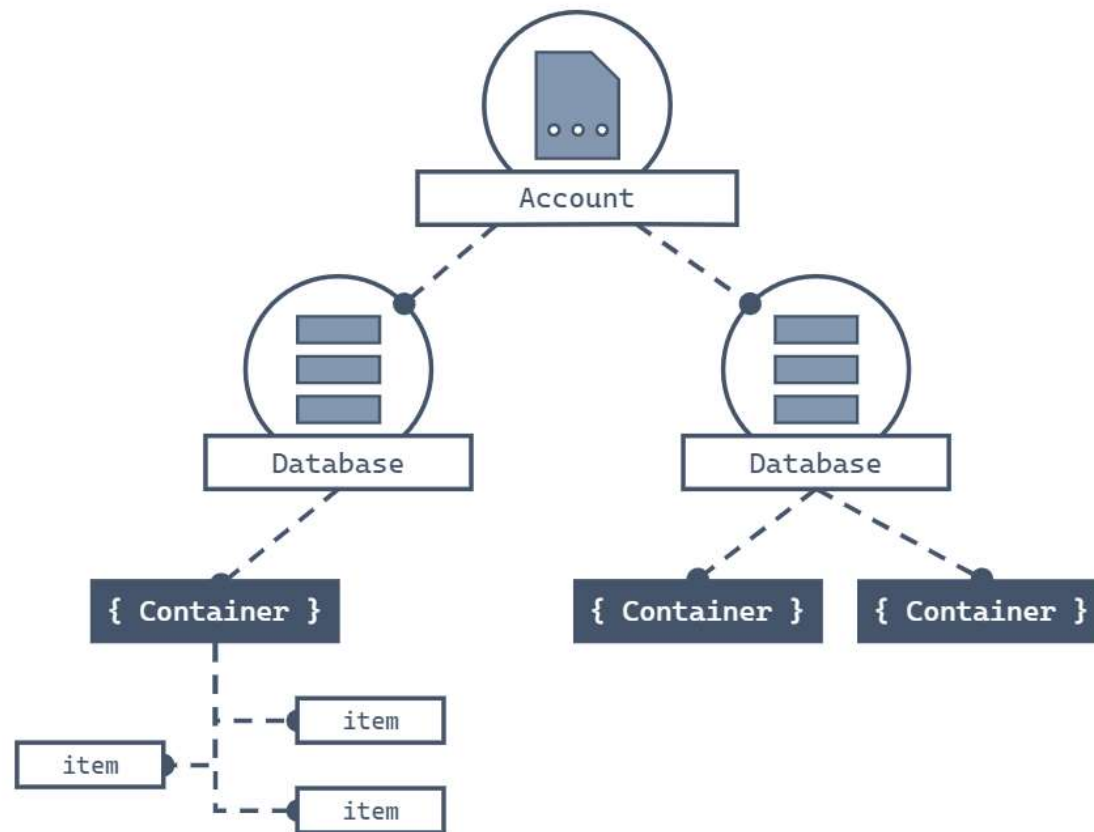
+ Create Restore Manage view Refresh Export to CSV Open query Assign tags

Filter for any field... Subscription equals all Type equals all Resource group equals all Location equals all Add filter

Showing 1 to 1 of 1 records.

<input type="checkbox"/> Name ↑↓	Status ↑↓	Subscription ↑↓	Write region ↑↓	Read Region ↑↓	
<input type="checkbox"/> luiscococosmosdb	Online	Azure subscription 1	France Central	France Central	...

Cosmos DB. Azure Cosmos DB has a specific object model used to create and access resources. The Azure Cosmos DB creates resources in a hierarchy that consists of accounts, databases, containers, and items.



6. Press the CosmosDB link “luiscococosmosdb” to navigate to the **Azure CosmosDB account**

The screenshot shows the Azure portal interface. The left sidebar contains navigation links: 'Create a resource', 'Home', 'Dashboard', 'All services', 'FAVORITES', 'All resources', 'Resource groups', 'Quickstart Center', 'App Services', 'Function App', 'SQL databases', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Azure Active Directory', 'Monitor', 'Advisor', and 'Microsoft Defender for Cloud'. The main content area is titled 'Azure Cosmos DB' and shows the account 'luiscococosmosdb'. The account details include: Status (Online), Read Locations (France Central), Write Locations (France Central), URI (https://luiscococosmosdb.documents.azure.com...), Subscription ID (846901e6-da09-45c8-98ca-7cca2353ff0e), Total throughput limit (1000 RU/s), and Free Tier Discount (Opted In). The 'Containers' section shows a table with one container named 'products' in the 'cosmicworks' database, with a throughput of 400 RU/s. The 'Monitoring' section shows a dropdown for 'Show data for last' with options: 1 hour, 24 hours (selected), 7 days, and 30 days.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Azure Cosmos DB >

Azure Cosmos DB

particular (luiscocoenriquezhotmail.onmicrosoft.c...

+ Create Restore ...

Filter for any field...

Name ↑↓

luiscococosmosdb

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Cost Management

Quick start

Notifications

Data Explorer

Settings

Features

Replicate data globally

Default consistency

Backup & Restore

Networking

CORS

Dedicated Gateway

luiscococosmosdb

Azure Cosmos DB account

Search

+ Add Container Refresh Move Data Explorer Enable geo-redundancy

Welcome to your Azure Cosmos DB Free Tier account! Your first 1000 RU/s and 25 GB of storage will be free for the lifetime of this account. Click here to learn more.

Essentials

Status: Online

Read Locations: France Central

Write Locations: France Central

URI: https://luiscococosmosdb.documents.azure.com...

Subscription ID: 846901e6-da09-45c8-98ca-7cca2353ff0e

Total throughput limit: 1000 RU/s

Free Tier Discount: Opted In

Capacity mode: Provisioned throughput

Containers

ID	Database	Throughput (RU/s)
products	cosmicworks	400

Monitoring

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

7. In the **Azure CosmosDB account** go to the “Keys” option

The screenshot displays the Azure portal interface. The left-hand navigation pane includes options like 'Create a resource', 'Home', 'Dashboard', 'All services', and a 'FAVORITES' section with 'All resources', 'Resource groups', 'Quickstart Center', 'App Services', 'Function App', 'SQL databases', 'Azure Cosmos DB', 'Virtual machines', 'Load balancers', 'Storage accounts', 'Virtual networks', 'Azure Active Directory', 'Monitor', and 'Advisor'. The main content area is titled 'Azure Cosmos DB' and shows a list of resources with 'luisccoccosmosdb' selected. A secondary pane on the right lists settings such as 'Features', 'Replicate data globally', 'Default consistency', 'Backup & Restore', 'Networking', 'CORS', 'Dedicated Gateway', 'Keys', 'Advisor Recommendations', 'Microsoft Defender for Cloud', 'Identity', and 'Locks'. The 'Keys' option is highlighted, leading to a detailed view of the account's keys. This view includes the 'URI' (https://luisccoccosmosdb.documents.azure.com:443/), 'Read-write Keys' and 'Read-only Keys' tabs, 'PRIMARY KEY' and 'SECONDARY KEY' sections (both last regenerated on 8/14/2023), and 'PRIMARY CONNECTION STRING' and 'SECONDARY CONNECTION STRING' sections. Each key and connection string is displayed in a masked format with a copy icon and a toggle to show the full value.

Microsoft Azure

Search resources, services, and docs (G+)

Home > Azure Cosmos DB > luisccoccosmosdb

Azure Cosmos DB
particular (luisccoccosmosdb.onmicrosoft.com)

+ Create Restore ...

Filter for any field...

Name ↑

luisccoccosmosdb

luisccoccosmosdb | Keys

Search Refresh

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

Read-write Keys Read-only Keys

PRIMARY KEY
Last regenerated: 8/14/2023 (0 days ago). [Learn more](#)

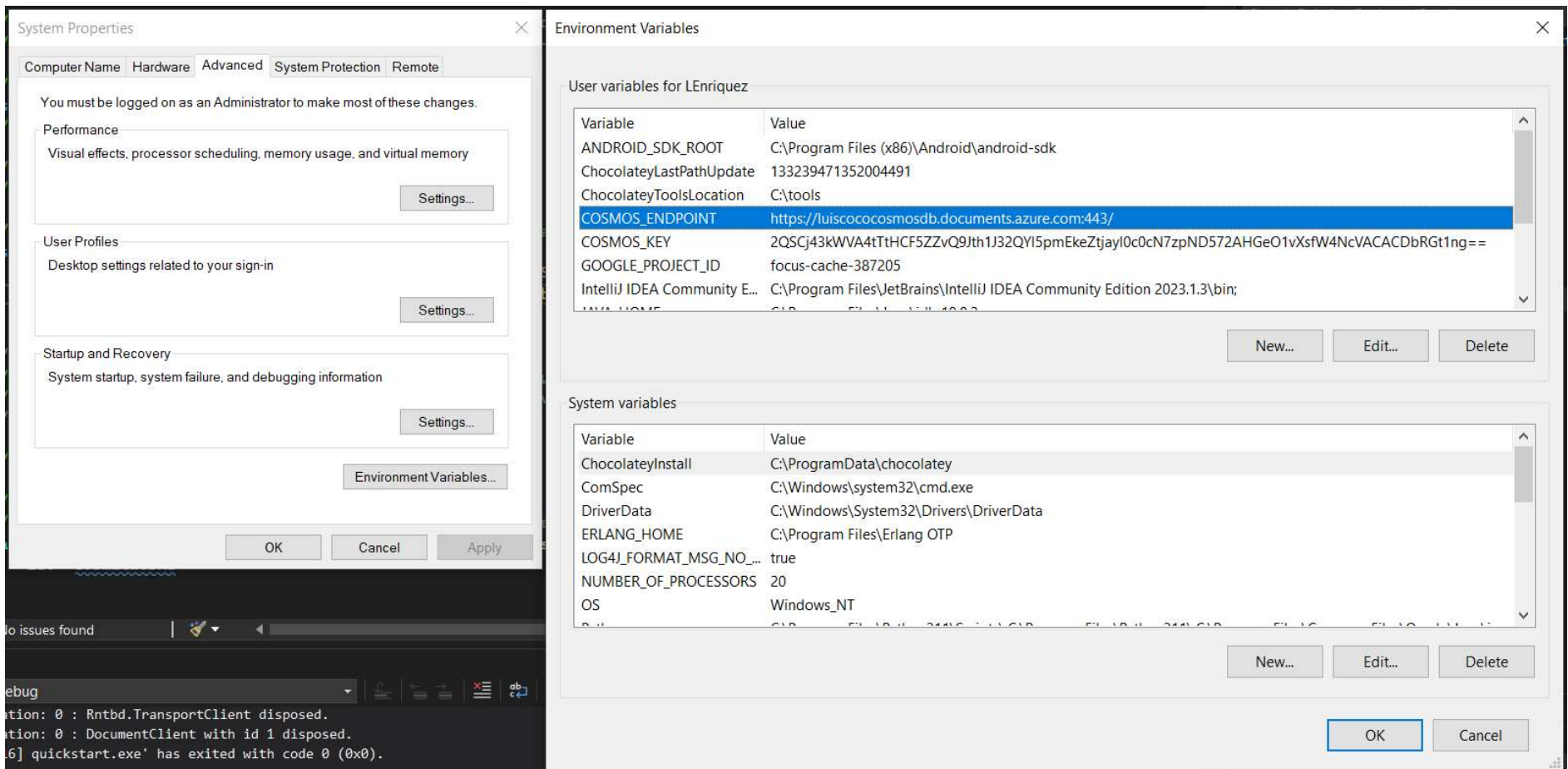
SECONDARY KEY
Last regenerated: 8/14/2023 (0 days ago). [Learn more](#)

PRIMARY CONNECTION STRING

SECONDARY CONNECTION STRING

Page 1 of 1

8. Set the environmental variables values: the **COSMOS_ENDPOINT** and the **COSMOS_KEY**



```
// -----  
// Copyright (c) Microsoft Corporation. All rights reserved.  
// -----  
  
// <using_directives>  
using Microsoft.Azure.Cosmos;  
// </using_directives>  
  
// <client_credentials>  
// New instance of CosmosClient class  
  
// We need to set these two variable in the Environment Variables...  
using CosmosClient client = new(  
    accountEndpoint: Environment.GetEnvironmentVariable("COSMOS_ENDPOINT")!,  
    authKeyOrResourceToken: Environment.GetEnvironmentVariable("COSMOS_KEY")!  
);  
// </client_credentials>  
  
// <new_database>  
// Database reference with creation if it does not already exist  
Database database = await client.CreateDatabaseIfNotExistsAsync(  
    id: "cosmicworks"  
);
```

```
Console.WriteLine($"New database:\t{database.Id}");
// </new_database>

// <new_container>
// Container reference with creation if it does not already exist
Container container = await database.CreateContainerIfNotExistsAsync(
    id: "products",
    partitionKeyPath: "/"categoryId",
    throughput: 400
);

Console.WriteLine($"New container:\t{container.Id}");
// </new_container>

// <new_item>
// Create new object and upsert (create or replace) to container
Product newItem = new(
    id: "70b63682-b93a-4c77-aad2-65501347265f",
    categoryId: "61dba35b-4f02-45c5-b648-c6badc0cbd79",
    categoryName: "gear-surf-surfboards",
    name: "Yamba Surfboard",
    quantity: 12,
    sale: false
);

Product createdItem = await container.CreateItemAsync<Product>(
    item: newItem,
    partitionKey: new PartitionKey("61dba35b-4f02-45c5-b648-c6badc0cbd79")
);
```

```
// <query_items>
// Create query using a SQL string and parameters
var query = new QueryDefinition(
    query: "SELECT * FROM products p WHERE p.categoryId = @categoryId"
)
    .WithParameter("@categoryId", "61dba35b-4f02-45c5-b648-c6badc0cbd79");

using FeedIterator<Product> feed = container.GetItemQueryIterator<Product>(
    queryDefinition: query
);

while (feed.HasMoreResults)
{
    FeedResponse<Product> response = await feed.ReadNextAsync();
    foreach (Product item in response)
    {
        Console.WriteLine($"Found item:\t{item.name}");
    }
}
// </query_items>
```