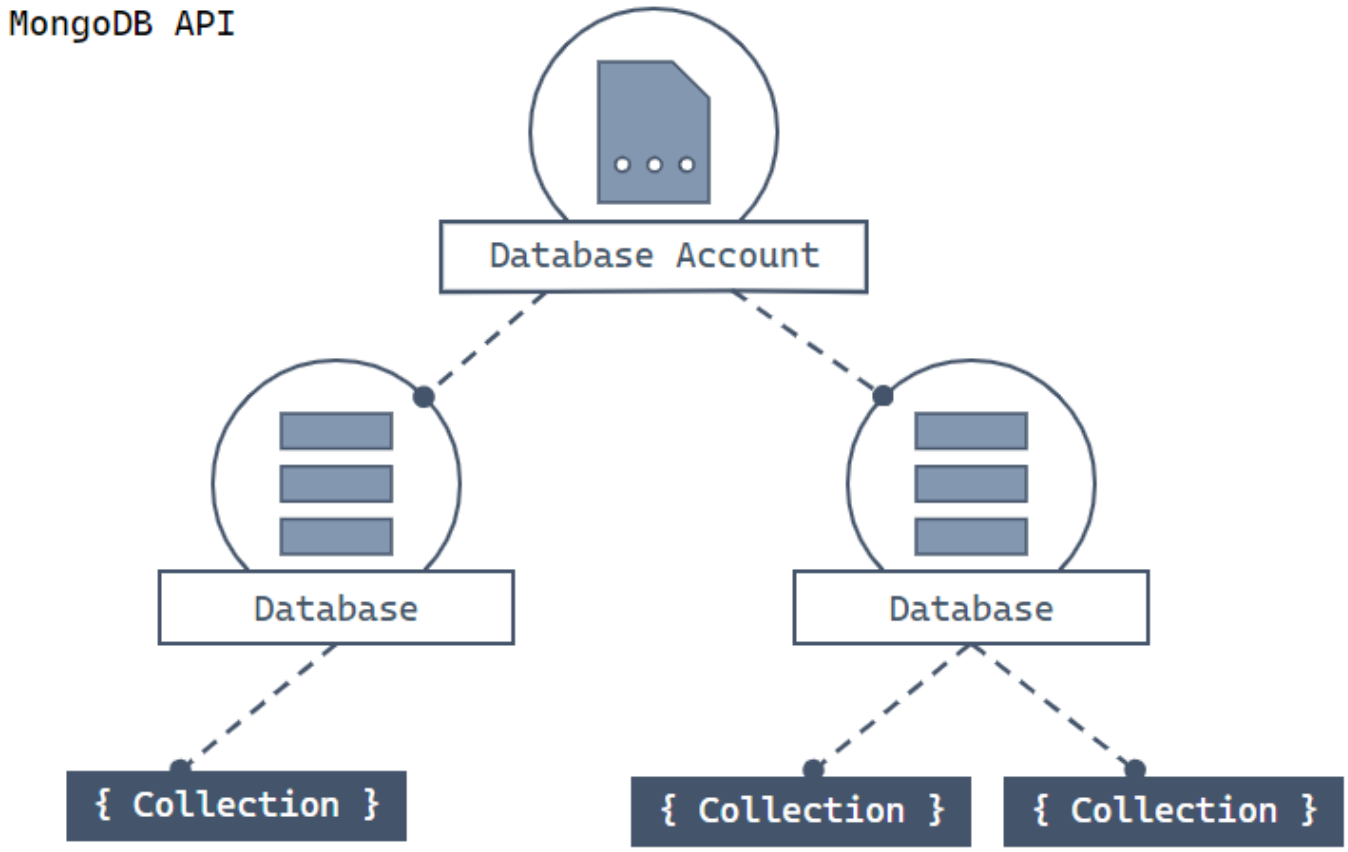
# Azure SDK for .NET Cosmos DB for MongoDB

https://learn.microsoft.com/en-us/azure/cosmos-db/mongodb/quickstart-dotnet




# 1. Prerequisites

## 1.1. Install .NET 8 SDK

For installing .NET 8 navigate to the URL: https://dotnet.microsoft.com/en-us/download/dotnet/thank-you/sdk-8.0.100-windows-x64-installer

To check the installation was successful run the command:

```
dotnet --list-sdks
```

## 1.2. Install the Azure CLI or Azure PowerShell

Navigate to the following URL to install Azure CLI: https://learn.microsoft.com/en-us/cli/azure/install-azure-cli-windows?tabs=azure-cli

Navigate to the following URL to install Azure PowerShell: https://learn.microsoft.com/en-us/powershell/azure/install-azure-powershell?view=azps-11.1.0

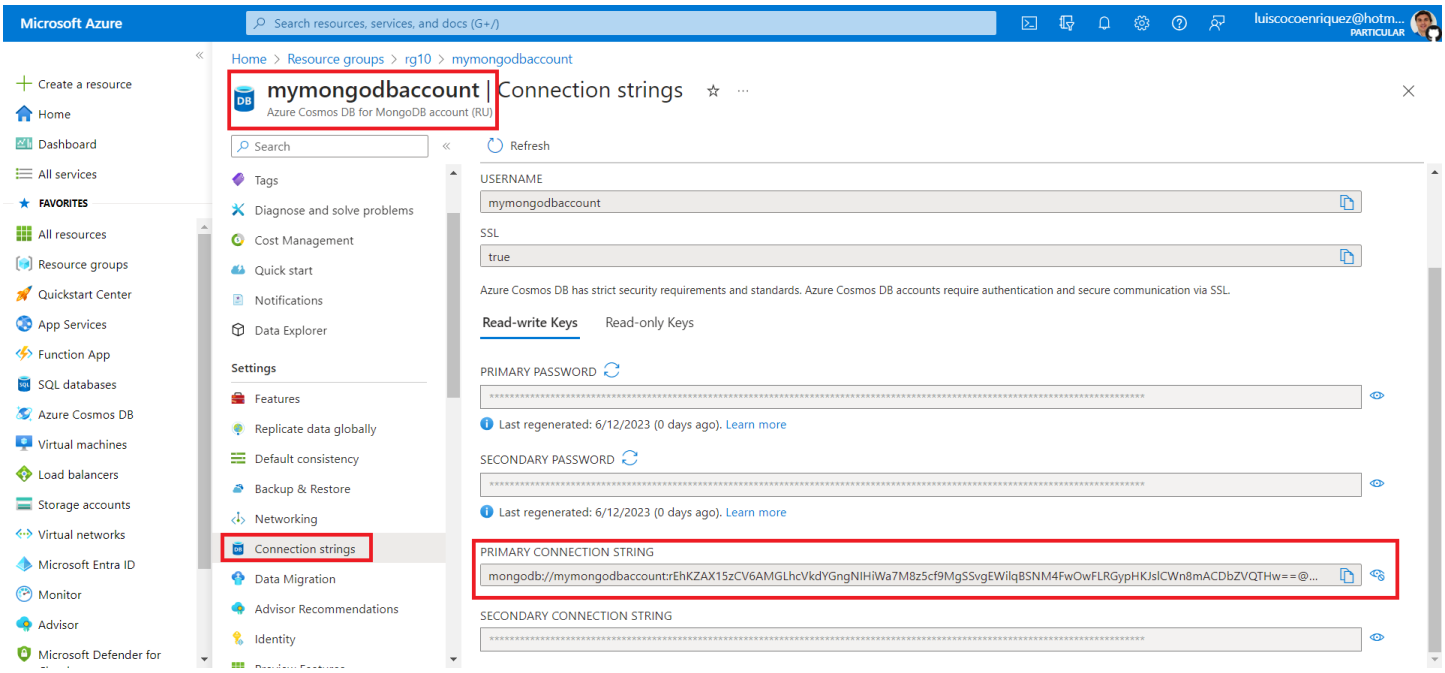
Check the installation was succesful running the commands:

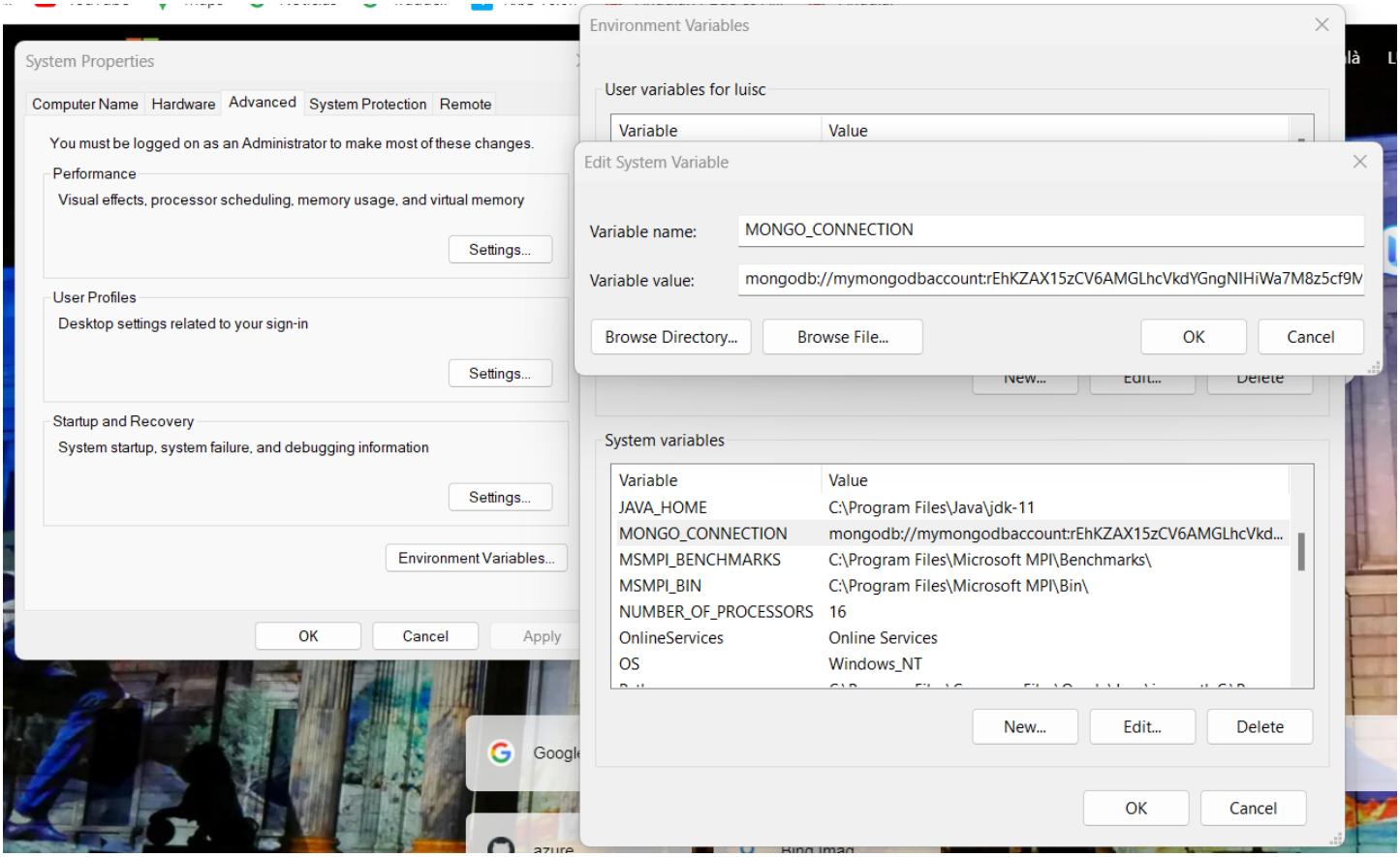
```
az --version
```

```
Get-Module -ListAvailable AzureRM
```

# 2. Create a new CosmosDB MongoDB account in the Azure portal

Copy the connection string and create a new environmental variable to store it






# 3. Create in VSCode a C# .NET 8 console application

Open VSCode in the folder where we would like to create the new application run the following command

```
dotnet new console --framework net8.0
```

Add the "Mongo.Driver" library

```
dotnet add package MongoDB.Driver
```

# 4. Input the application source code

```
using System;
using System.Security.Authentication;
using System.Threading.Tasks;
using MongoDB.Driver;

//https://learn.microsoft.com/en-us/azure/cosmos-db/mongodb/quickstart-dotnet
```

```
//1. New instance of CosmosClient class
//string connectionString = @"mongodb://mymongodbaccount:rEhKZAX15zCV6AMGLhcVkdYGngNIHiWa7M8z5

string connectionString = Environment.GetEnvironmentVariable("MONGO_CONNECTION");

MongoClientSettings settings = MongoClientSettings.FromUrl(new MongoUrl(connectionString));

settings.SslSettings = new SslSettings() { EnabledSslProtocols = SslProtocols.Tls12 };

var MongoDBclient = new MongoClient(settings);

//2. Database reference with creation if it does not already exist
var db = MongoDBclient.GetDatabase("adventure");

//3. Container reference with creation if it does not alredy exist
var _products = db.GetCollection<Product>("products");

//4. Create new object and upsert (create or replace) to container
_products.InsertOne(new Product(
    Guid.NewGuid().ToString(),
    "gear-surf-surfboards",
    "Yamba Surfboard",
    12,
    false
));

//5. Read a single item from container
var product = (await _products.FindAsync(p => p.Name.Contains("Yamba"))).FirstOrDefault();
Console.WriteLine("Single product:");
Console.WriteLine(product.Name);

//6. Read multiple items from container
_products.InsertOne(new Product(
    Guid.NewGuid().ToString(),
    "gear-surf-surfboards",
    "Sand Surfboard",
    4,
    false
));

var products = _products.AsQueryable().Where(p => p.Category == "gear-surf-surfboards");

Console.WriteLine("Multiple products:");
foreach (var prod in products)
{
    Console.WriteLine(prod.Name);
}

//7. We define the "Product" class

public record Product(
    string Id,
    string Category,
```

```
        string Name,
        int Quantity,
        bool Sale
    );
```

# 5. Build and run the application

For running the application type this command:

```
dotnet run
```

See the output in the terminal window