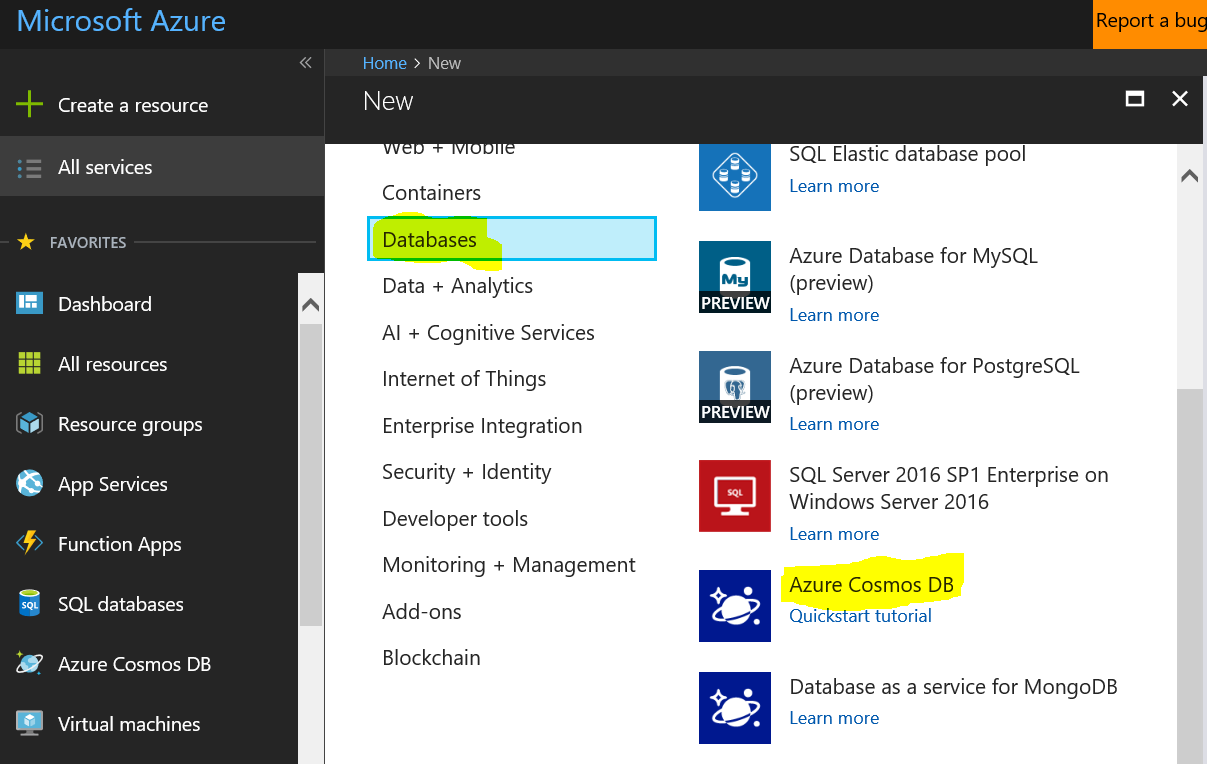
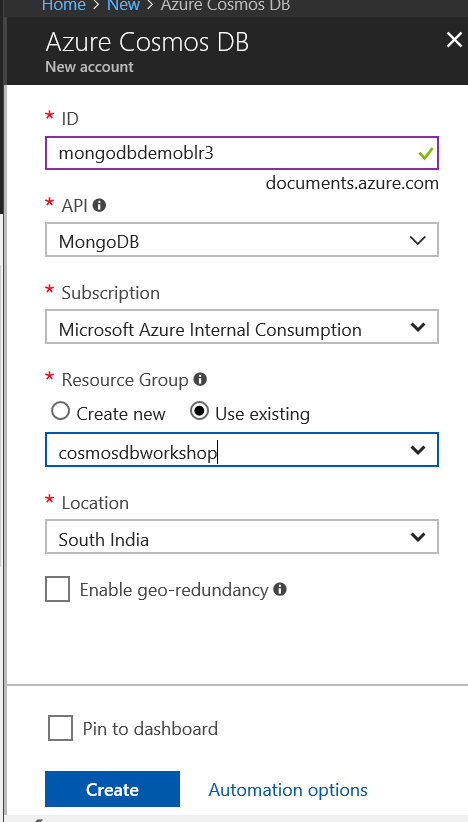
# Migrate data from MongoDB on local to Azure Cosmos DB

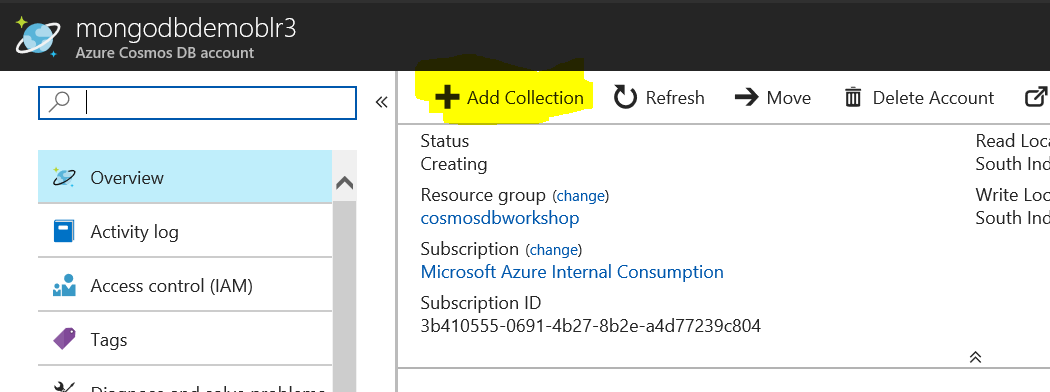
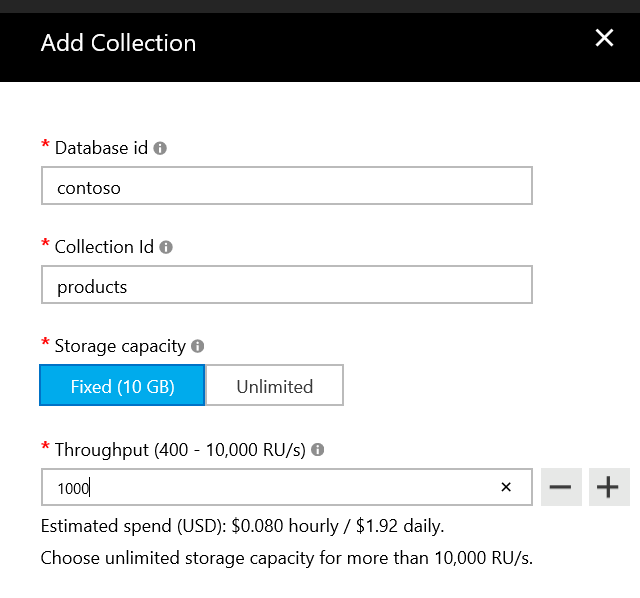
# Objective

Objective of this hands-on-lab is to help you to migrate data from on premise MongoDB Databa server to Azure Cosmos DB, and a sample Web Application which will be able to switch it’s database connection string, pointing from local MongoDB to Azure Cosmos and start communicating with Azure cosmos without any code change in the application.

# Creating Azure Cosmos Mongo DB

1. Go to <https://portal.azure.com> and click on “+ Create a resource” on the top left corner of the portal and go to “Databases -> Azure Cosmos DB” 
2. In the next page, provide ID, API (**Select MongoDB**), Subscription, Resource Group, location and click create

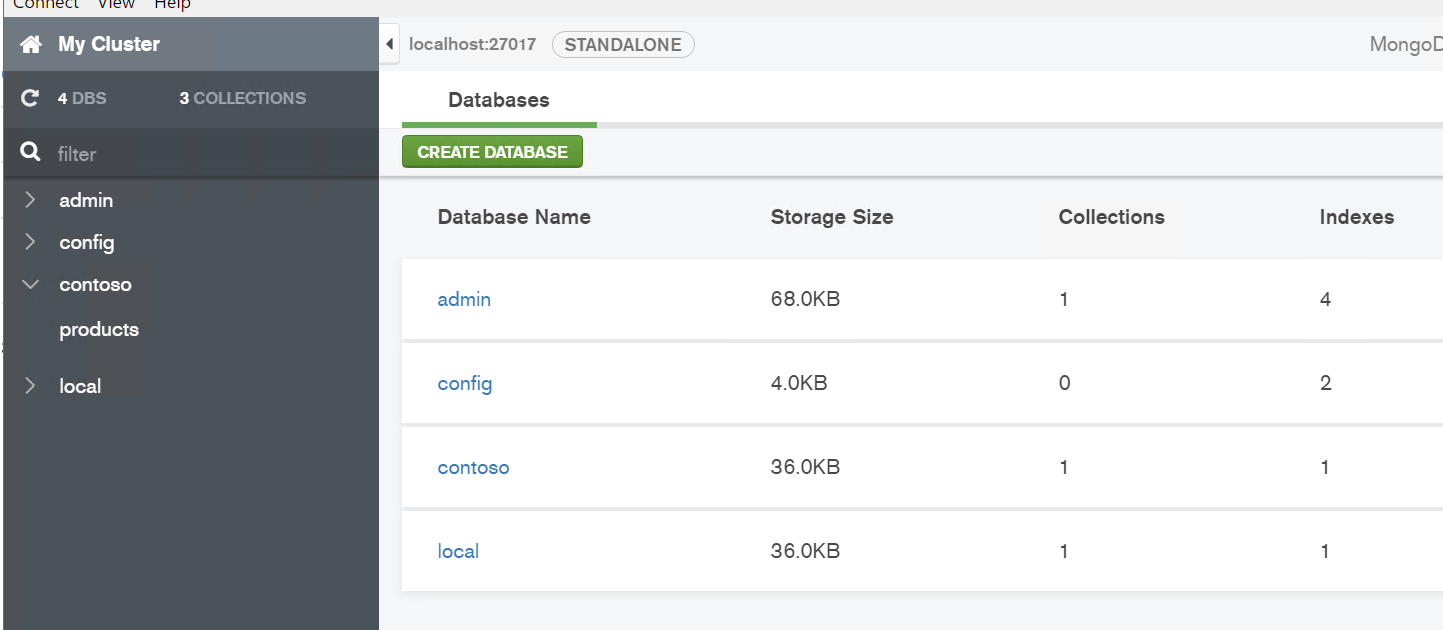


1. In the next page, Go to “Overview” and click on “Add Collection” 
2. Provide details for **Database ID, Collection ID, select storage capacity as “Fixed 10GB”** as this is testing purpose, and throughput about 1000 and click “OK”

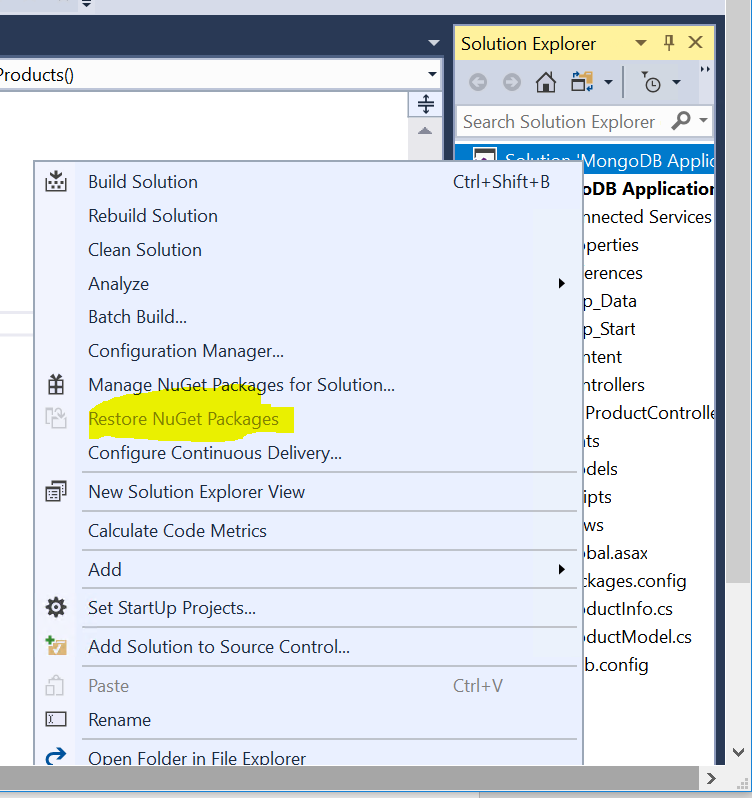
# Installing Mongo DB & Client in your computer

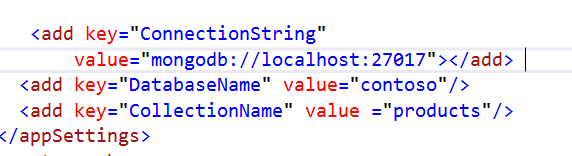
*[You may skip this step if you have already MongoDB installed or accessible from your computer ]*

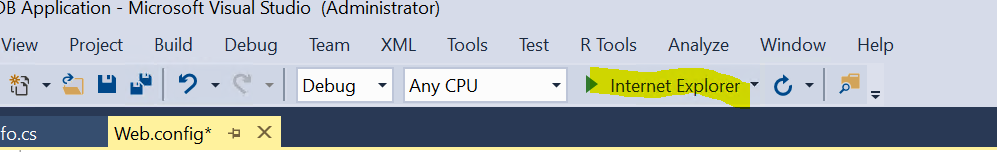
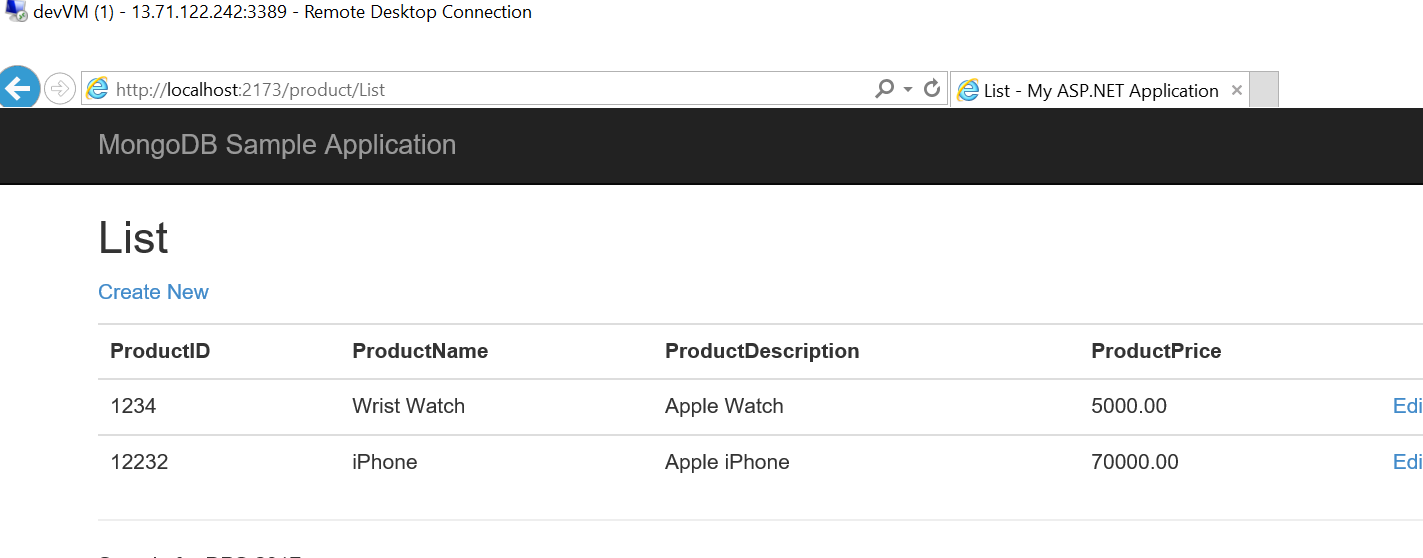
1. Follow the instructions given in <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>
2. Go to this page <https://www.mongodb.com/download-center?_ga=2.261004942.2078043922.1512711227-1658917347.1512711227#community> to download and install “Community Server”, as part of this installation, you may also install “Mongo Compass” client tool
3. Follow the steps given in this MongoDB documentation to configure MongoDB <https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/>
4. If you have followed all the steps correctly, then when you launch “Mongo Compass” then you should see databases and collections like the one shown below. Now go ahead and create database called “contoso” and within that create a collection called “products”. We will be using this to connect form the ASP.NET web application



# Build a sample ASP.NET Application

1. Go to the GIT Repo and download/clone the ASP.NET Web Application <https://github.com/mannu2050/MongoDB-CosmosDB>
2. Open the solution in visual Studio and do “Restore NuGet packages” at the solution level
3. Open the “Web.Config” file and update the connection string, to make the application to point to the local MongoDB instance. In general MongoDB connection string is in the form mongodb://<<UserName>>:<<Password>>@localhost:27017/<<Database>>. In the below scenario, user name and password is not present as “authentication mode” is not enable in the demo DB.



1. Run the application 
2. Enter the URL - <http://localhost:2173/product> in the browser and hit enter and you should see the application page similar to the one shown (without data) , This indicates that your application. You can go ahead and click “Create New” button and populate few sample products

# Migrating local MongoDB to Azure Cosmos DB

Go to MongoDB install folder “C:\Program Files\MongoDB\Server\3.6\bin” in command prompt and type the following Mongo Import command

1. Start the Mongo DB server with binding to 127.0.0.1 using the command “**mongod --bind\_ip 127.0.0.1”**
2. Use the following command to export the documents to JSON file

Sample Command – “mongoexport --host 127.0.0.1 --db contoso --collection products –out products.json”

1. Use the following command to import the documents to Azure Cosmos MongoDB

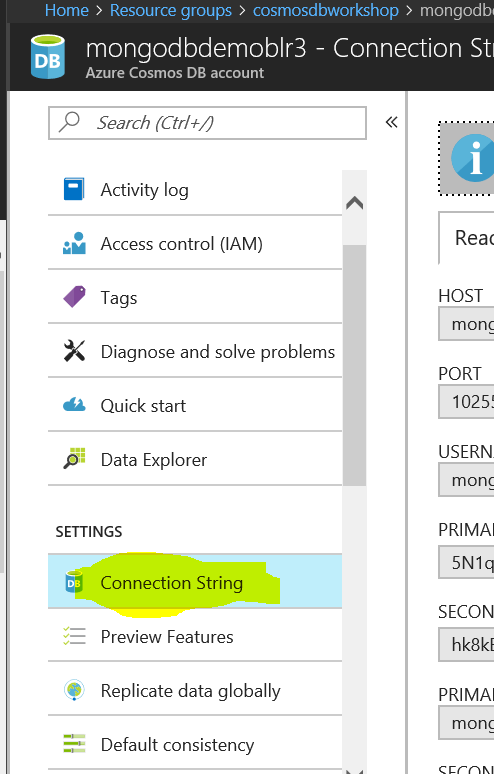
**Template**

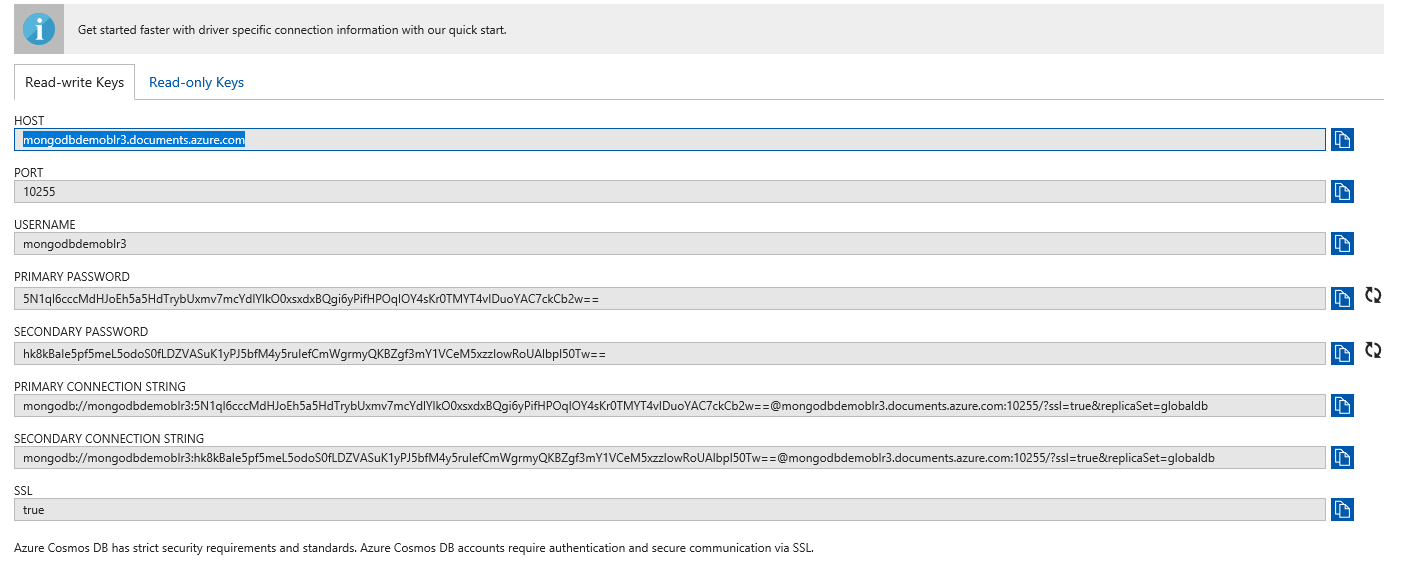
mongoimport.exe --host <your\_hostname>:10255 -u <your\_username> -p <your\_password> --db <your\_database> --collection <your\_collection> --ssl --sslAllowInvalidCertificates --type json --file C:\sample.json

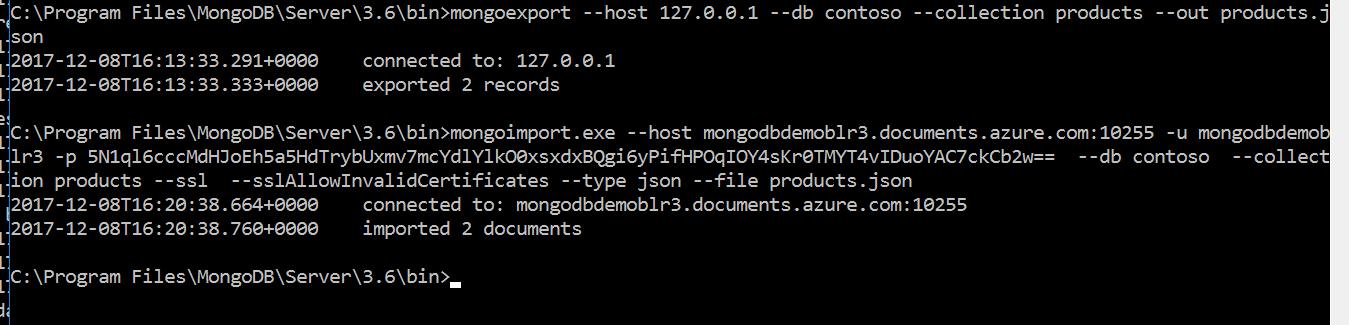
**Sample**

mongoimport.exe --host ***mongodbdemoblr3.documents.azure.com:10255*** -u ***mongodbdemoblr3*** -p ***5N1ql6cccMdHJoEh5a5HdTrybUxmv7mcYdlYlkO0xsxdxBQgi6yPifHPOqIOY4sKr0TMYT4vIDuoYAC7ckCb2w==*** --db ***contoso*** --collection ***products*** --ssl --sslAllowInvalidCertificates --type json --file ***products.json***

you can obtain the connection string, user name, port number etc.. from Azure Cosmos Mongo DB account setting as shown below



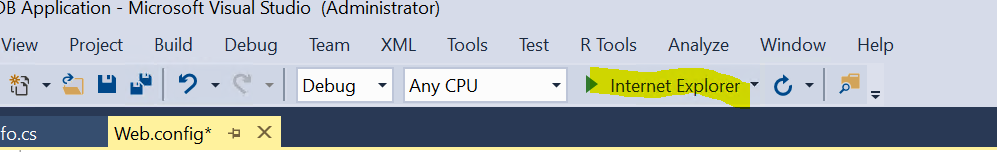




# Make Website to connect to Azure Cosmos MongoDB

1. Open the “Web.Config” file and update the connection string, to make the application to point to the Azure MongoDB instance.

<add key="ConnectionString" value="mongodb://<<mongodbusername>>:<<Primary Password>>==@<<Host>>:10255/?ssl=true"/>

1. Run the application 
2. Enter the URL - <http://localhost:2173/product> in the browser and hit enter and you should see the application page similar to the one shown and you should also see that data you had in your local MongoDB got migrated to Azure Cosmos MongoDB and your application now able to pull the data as well. 