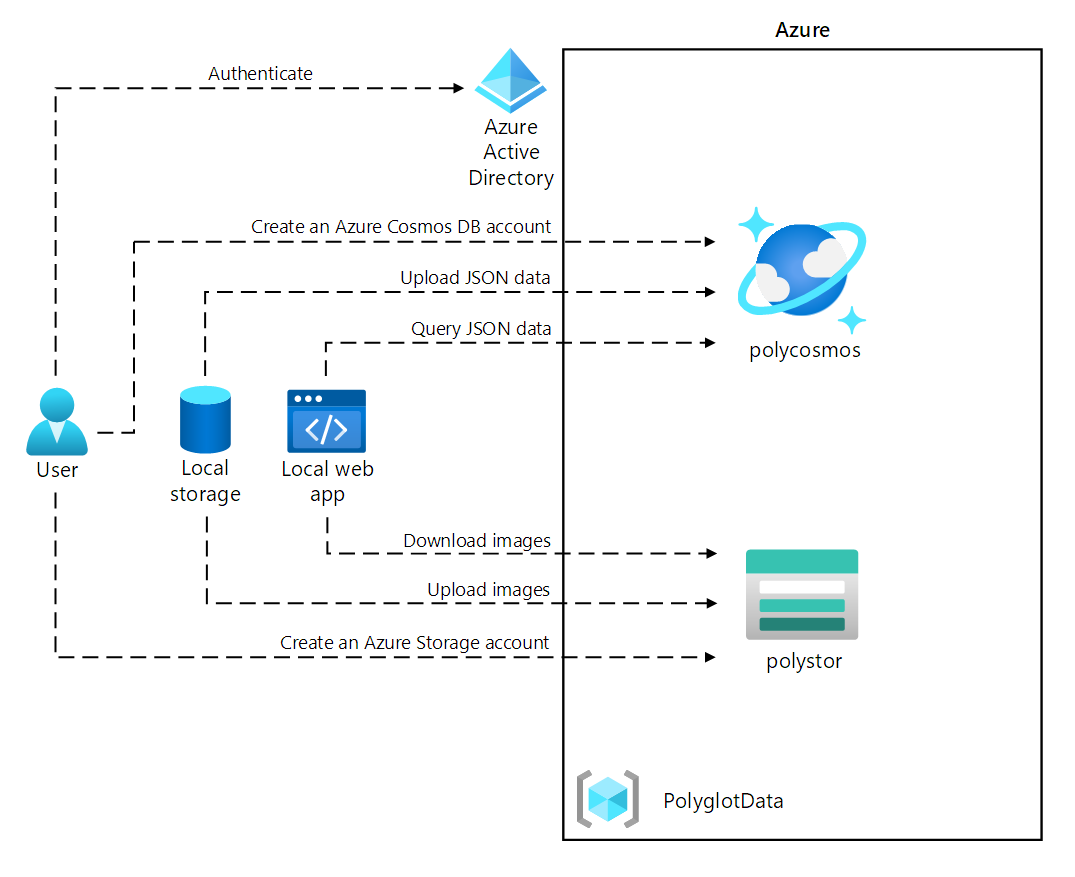
1. Develop solutions that use Cosmos DB storage

# Lab 04: Construct a polyglot data solution

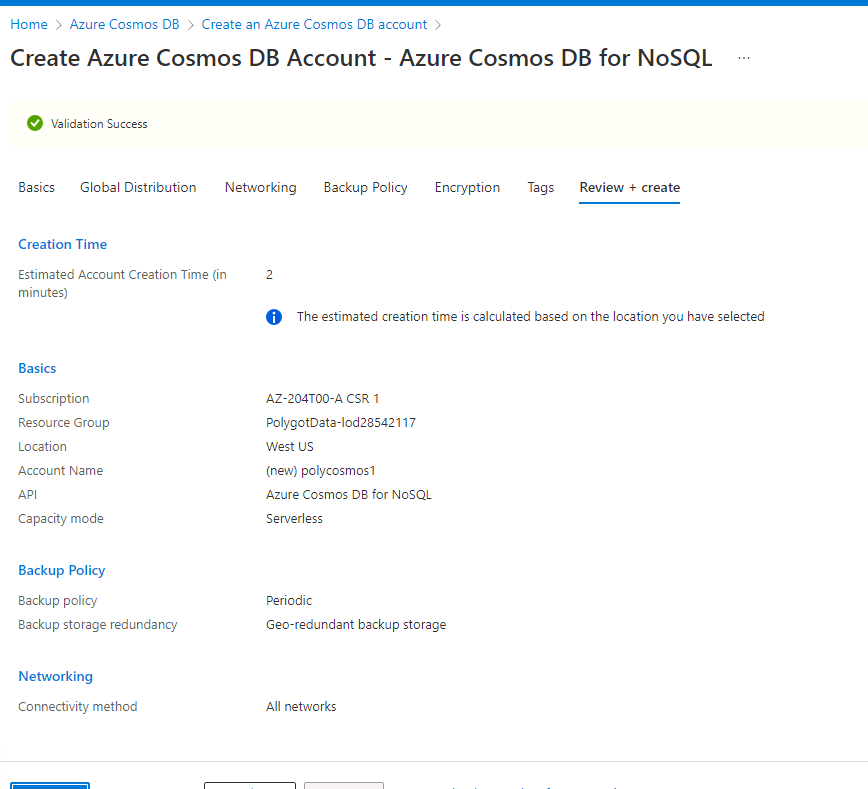
Загрузили через портал 42 картинки в контейнер Images, посмотрели JSON на диске, вставили его в БД, с портала сделали запросы и проверили что вставилось,

написали приложение, которое подключается к этой бд, на локалхост, выводит данные



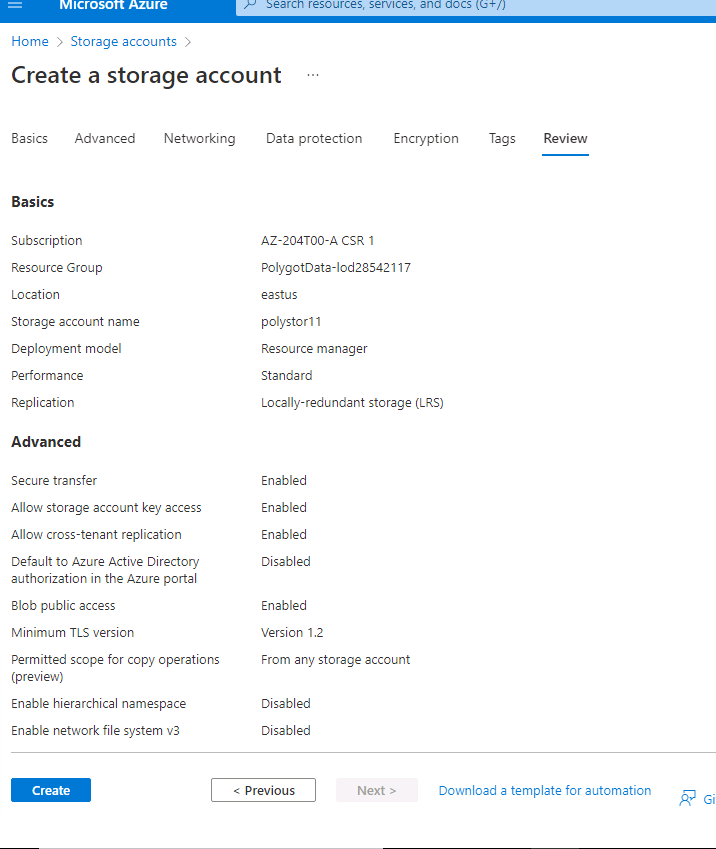
### Exercise 1: Creating data store resources in Azure

#### Create an Azure Cosmos DB account resource



#### 

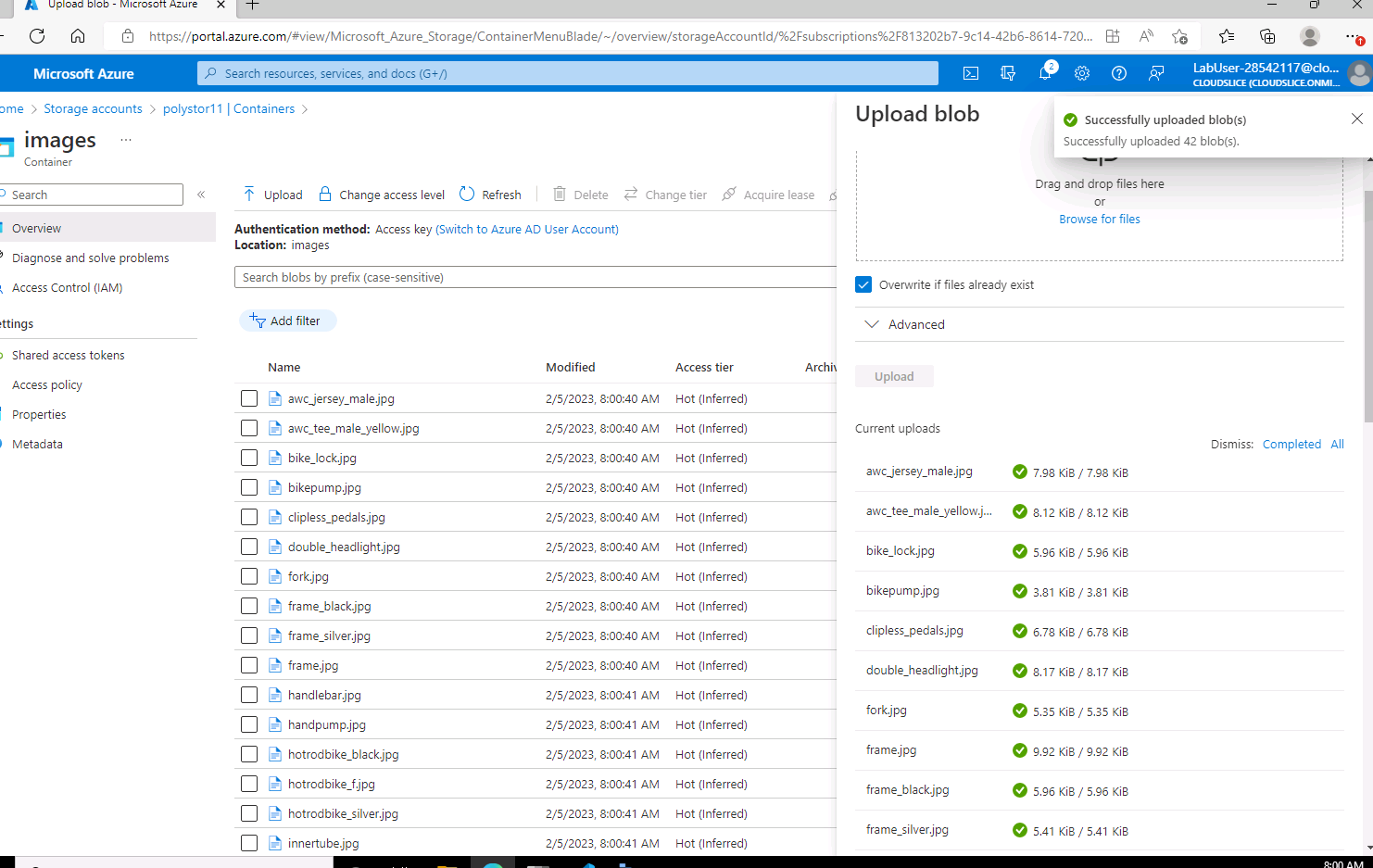
#### Create an Azure Storage account resource



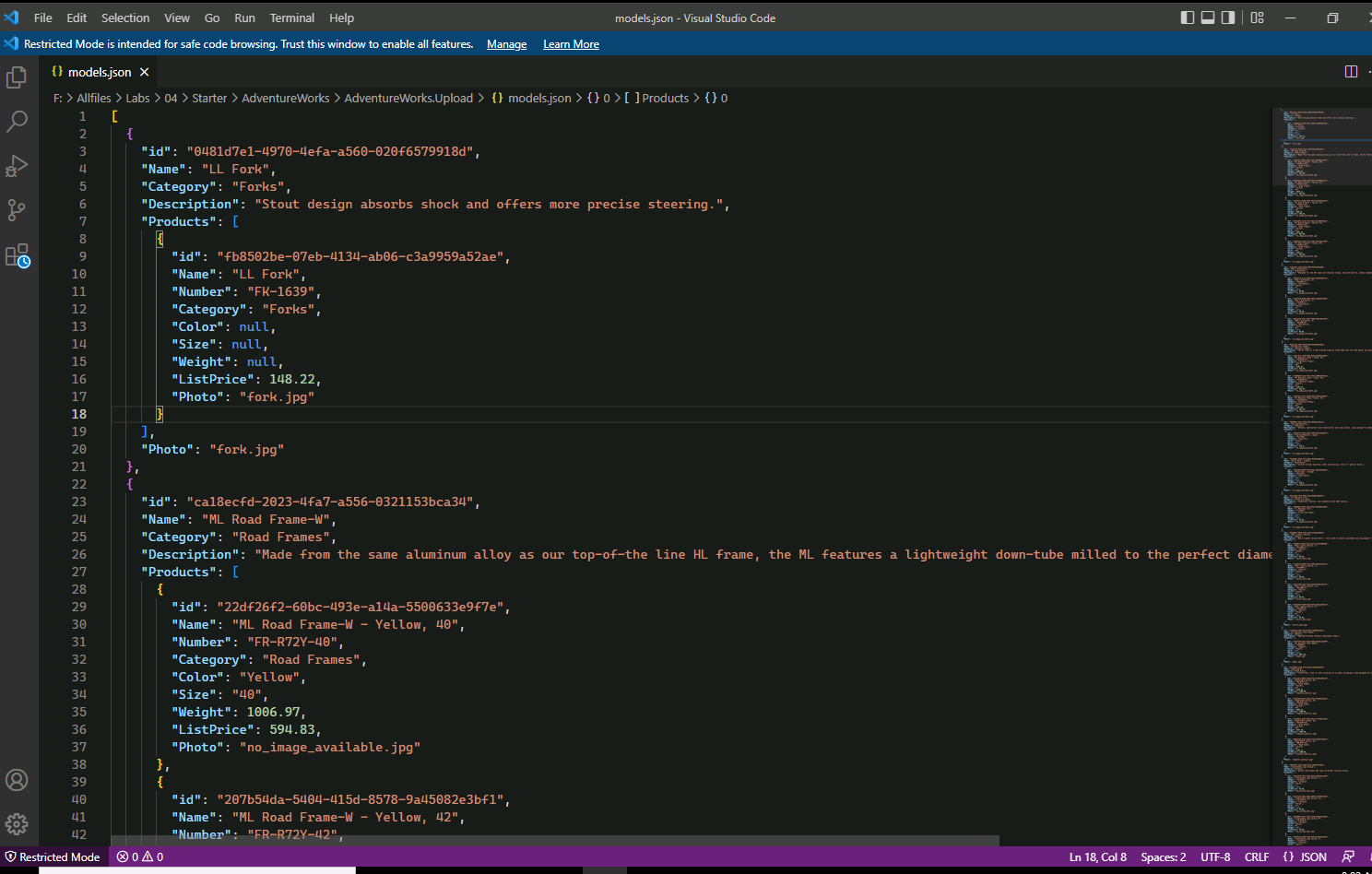
In this exercise, you created the Azure resources that you'll need for the polyglot data solution you'll implement in this lab. The Azure resources you created include an Azure Cosmos DB account and an Azure Storage account.

### Exercise 2: Review and upload data

#### Upload images to Azure Blob Storage

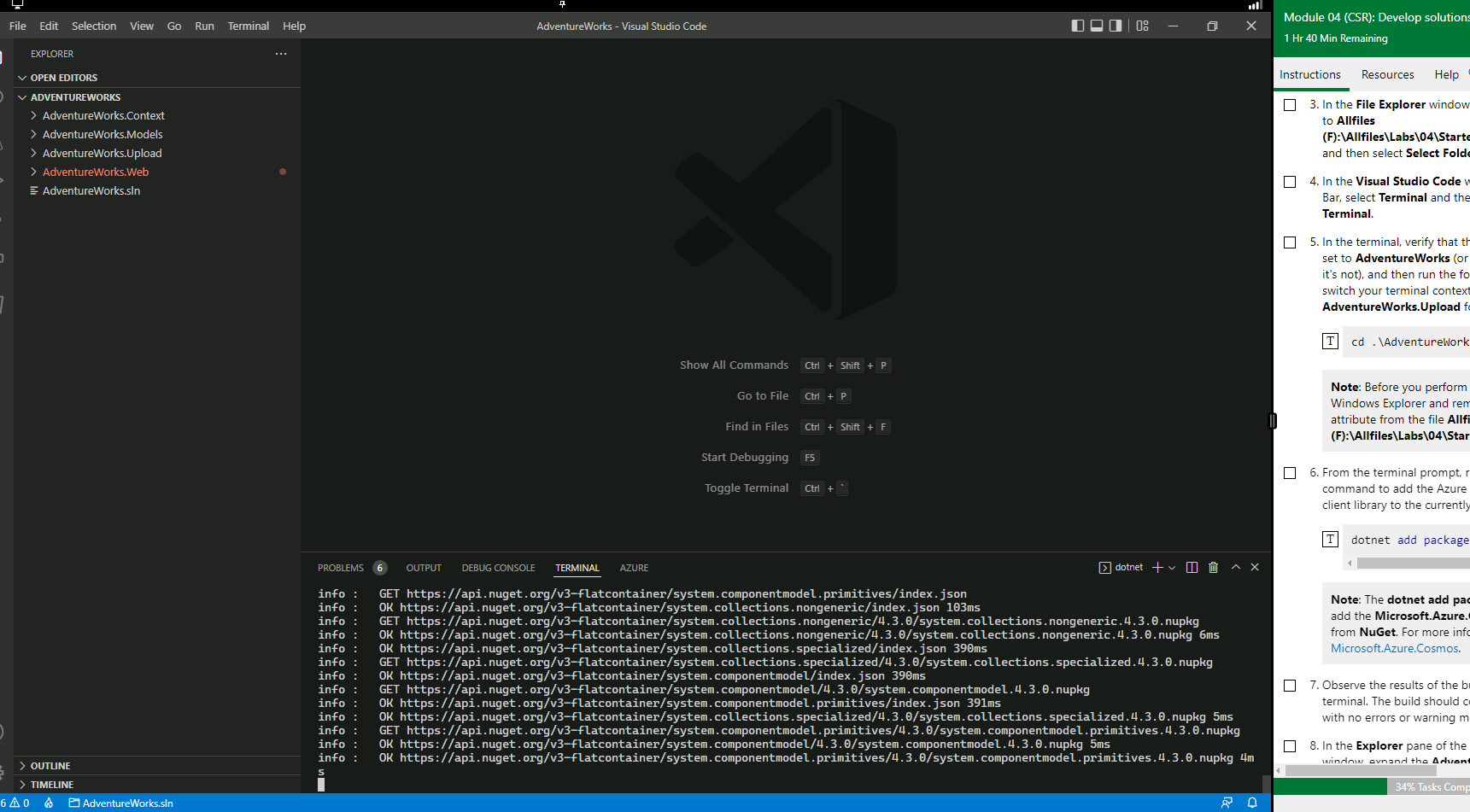


#### Review JSON data



This will determine the classes you'll define to deserialize the JSON file's contents before uploading it to a Cosmos DB collection.

 You'll use the **Category** property to define partitioning of the target Cosmos DB collection.

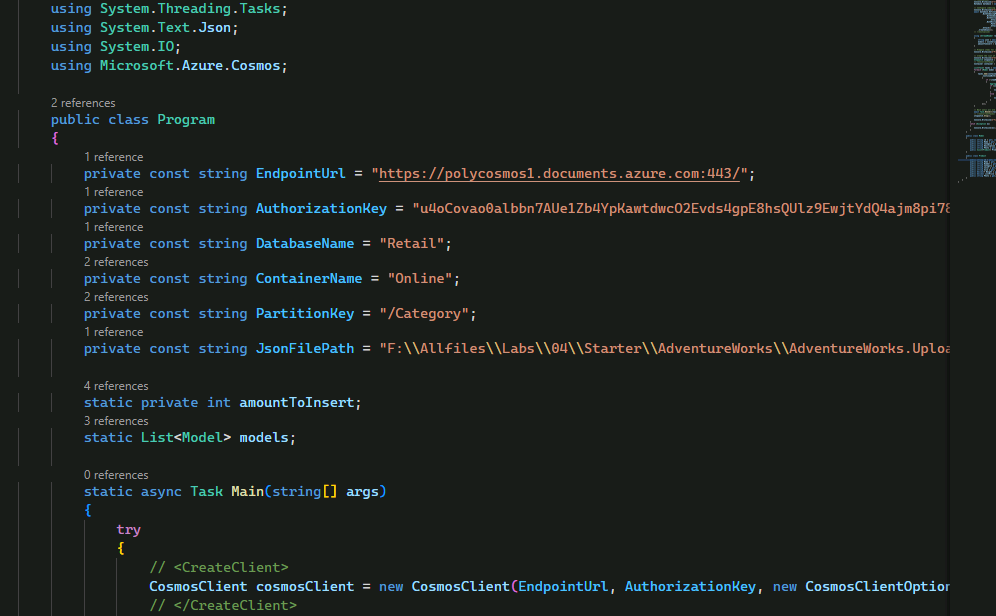


Установили пакет нагет

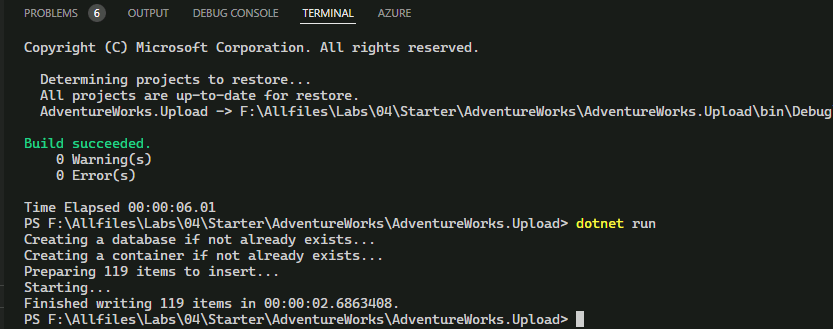
Within the try block, note the invocation of the **CreateDatabaseIfNotExistsAsync** method of the **CosmosClient** class. This will create a databa

Note the invocation of the **DefineContainer** method of the **Database** class. This will create a container that will host the JSON items if one doesn't already exist.

**Model** and **Product**

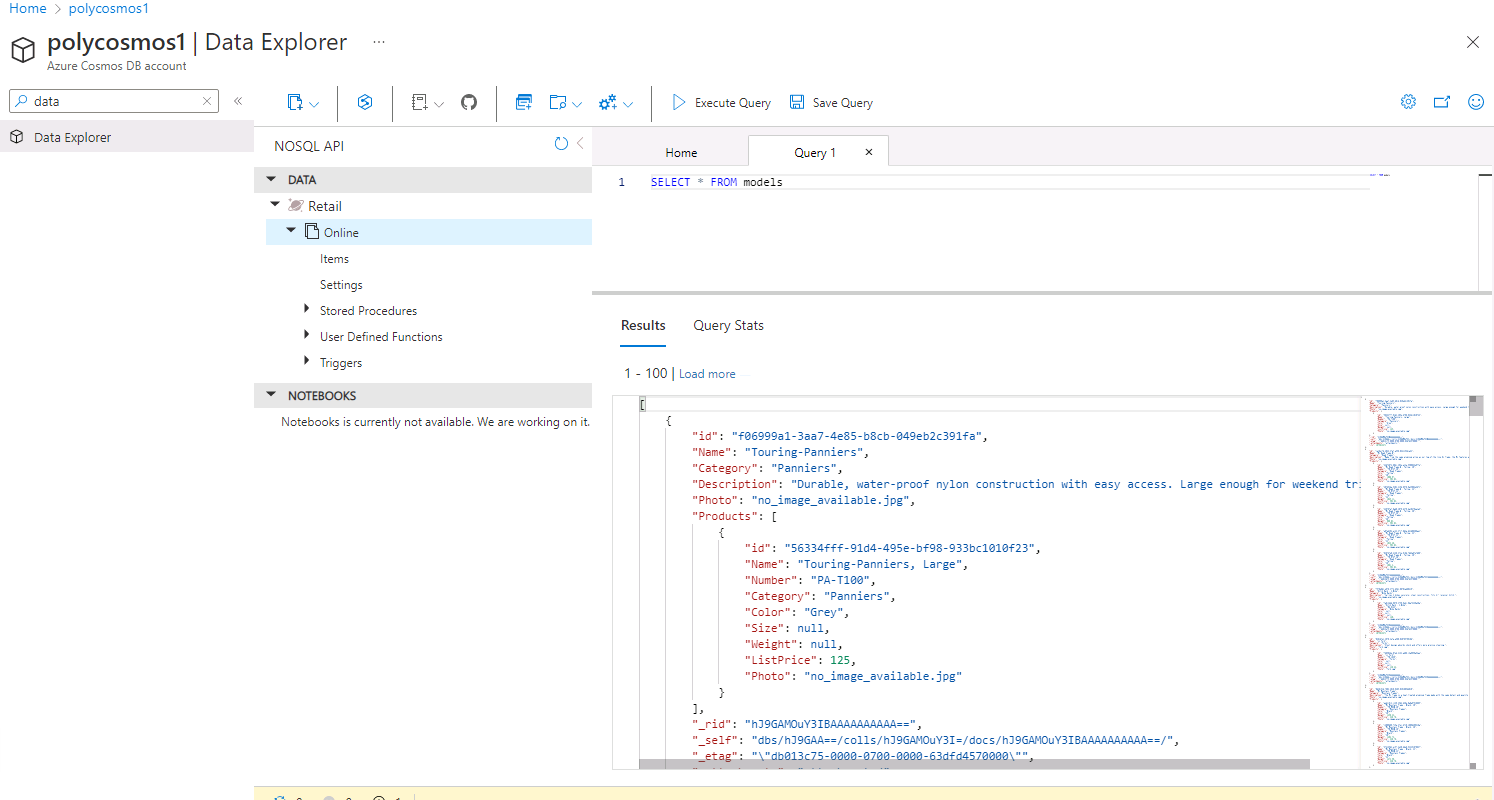


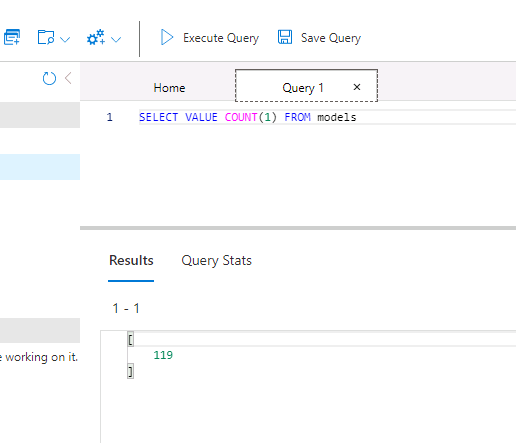
Run



l. The run should complete successfully, displaying the message about there being 119 items inserted into the target Cosmos DB collection.

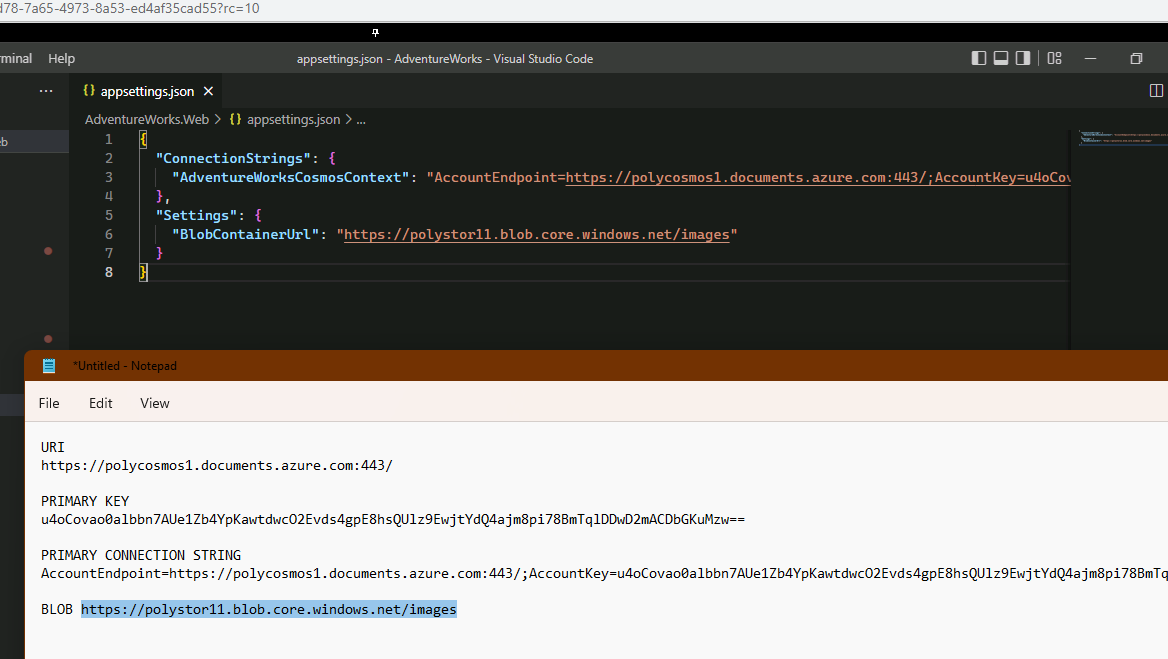
#### Validate JSON data upload



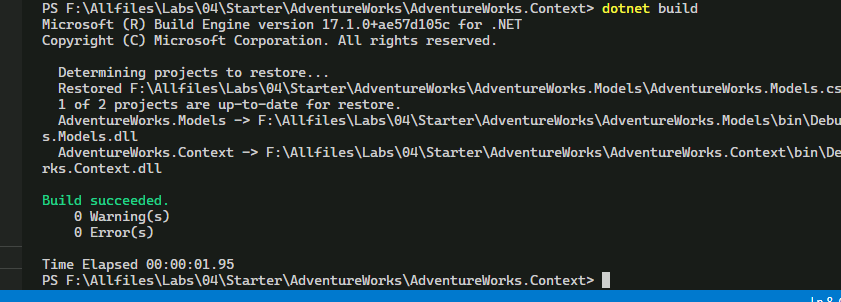


In this exercise, you used the .NET SDK for Azure Cosmos DB to insert data into Azure Cosmos DB. The web application that you implement next will use this data.

### Exercise 3: Configure a .NET web application



Buidl

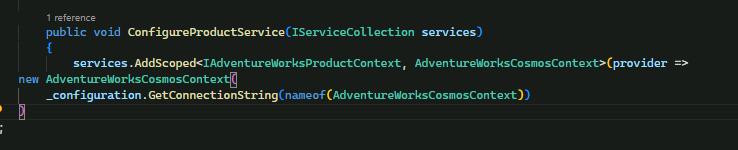


#### Configure connectivity to Azure Cosmos DB



Build

#### Review the .NET application startup logic



#### Validate that the .NET application successfully connects to data stores

Dotnet run

