AhmadSultan_041034155

CST8921 – Cloud Industry Trends

Lab 3 Report

Title

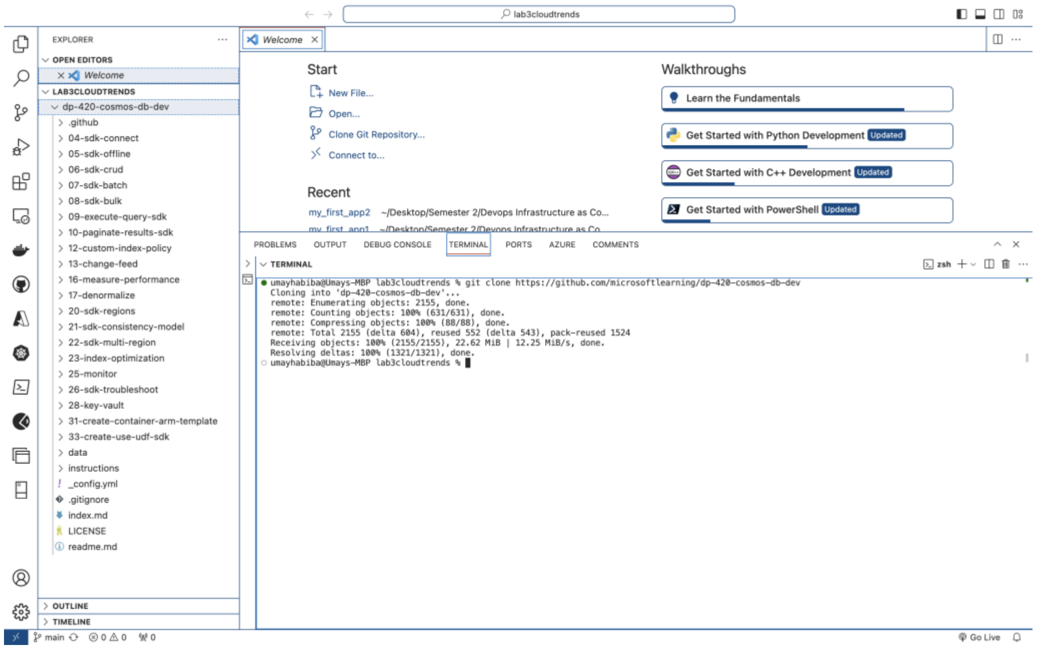
Cosmos DB Change Feed Integration with Azure Functions.

Introduction

In Lab 3, students delve into Azure Cosmos DB's capabilities, exploring its multi-model database with a focus on change feed functionality. Leveraging Azure Functions, this lab guides users through creating an application that reacts to create or update operations on items in a specified container, showcasing real-time data integration.

Steps

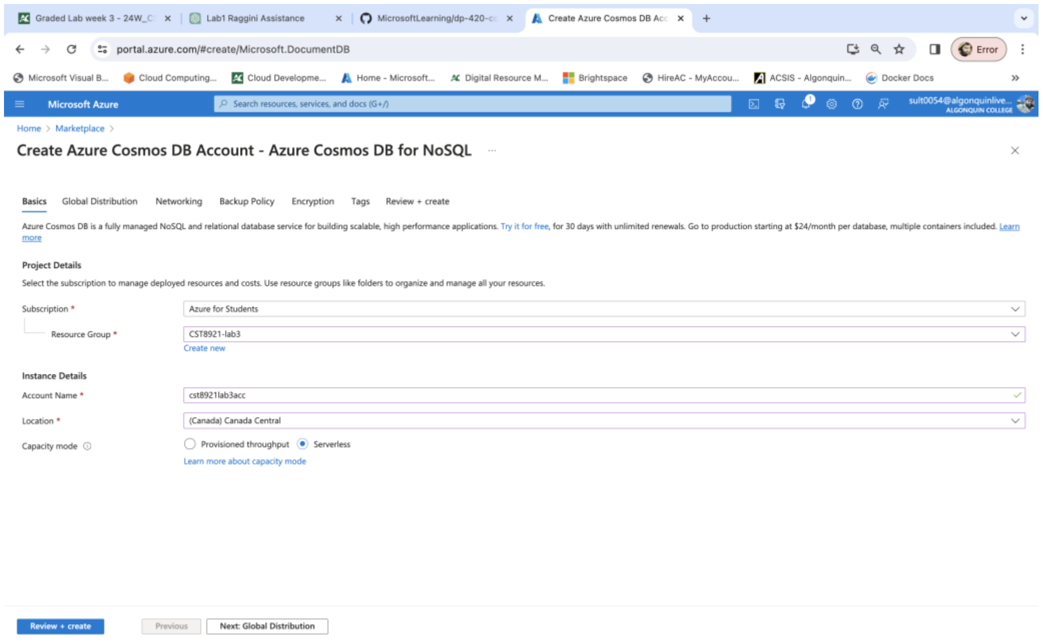
1. **Clone Repository:**
   - Start Visual Studio Code.
   - Open the command palette and run Git: Clone to clone the GitHub repository in a local folder. Open the local folder in Visual Studio Code.



```
umayhabiba@Umays-MBP lab3cloudtrends % git clone https://github.com/microsoftlearning/dp-420-cosmos-db-dev
Cloning into 'dp-420-cosmos-db-dev'...
remote: Enumerating objects: 2155, done.
remote: Counting objects: 100% (631/631), done.
remote: Compressing objects: 100% (88/88), done.
remote: Total 2155 (delta 604), reused 552 (delta 543), pack-reused 1524
Receiving objects: 100% (2155/2155), 22.62 MiB | 12.25 MiB/s, done.
Resolving deltas: 100% (1321/1321), done.
umayhabiba@Umays-MBP lab3cloudtrends %
```

2. **Create Azure Cosmos DB Account:**

   - Create an Azure Cosmos DB for NoSQL account with serverless capacity mode.
   - Review keys pane in Cosmos DB account for connection details.



Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Azure Cosmos DB is a fully managed NoSQL and relational database service for building scalable, high performance applications. Try it for free, for 30 days with unlimited renewals. Go to production starting at $24/month per database, multiple containers included. Learn more

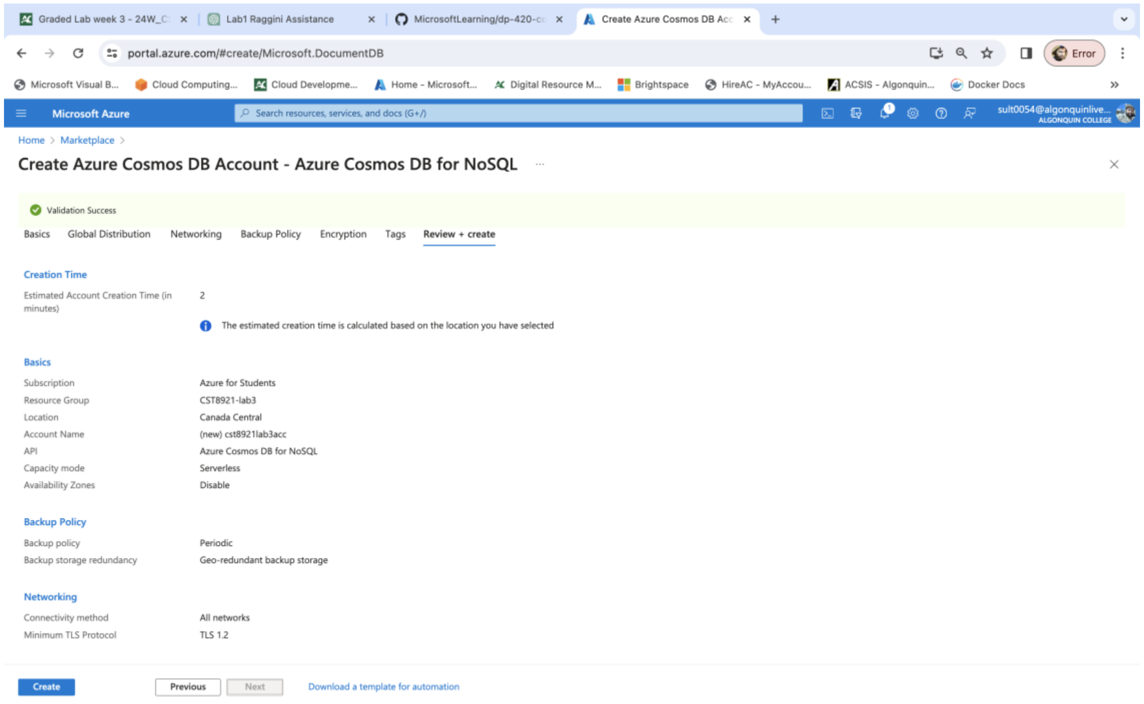
Project Details

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

Subscription *	Azure for Students
Resource Group *	CST8921-lab3

Instance Details

Account Name *	cst8921lab3acc
Location *	(Canada) Canada Central
Capacity mode	Provisioned throughput [✓ Serverless]



Create Azure Cosmos DB Account - Azure Cosmos DB for NoSQL

Validation Success

Creation Time

Estimated Account Creation Time (in minutes)	2

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

Basics

Subscription	Azure for Students
Resource Group	CST8921-lab3
Location	Canada Central
Account Name	(new) cst8921lab3acc
API	Azure Cosmos DB for NoSQL
Capacity mode	Serverless
Availability Zones	Disable

Backup Policy

Backup policy	Periodic
Backup storage redundancy	Geo-redundant backup storage

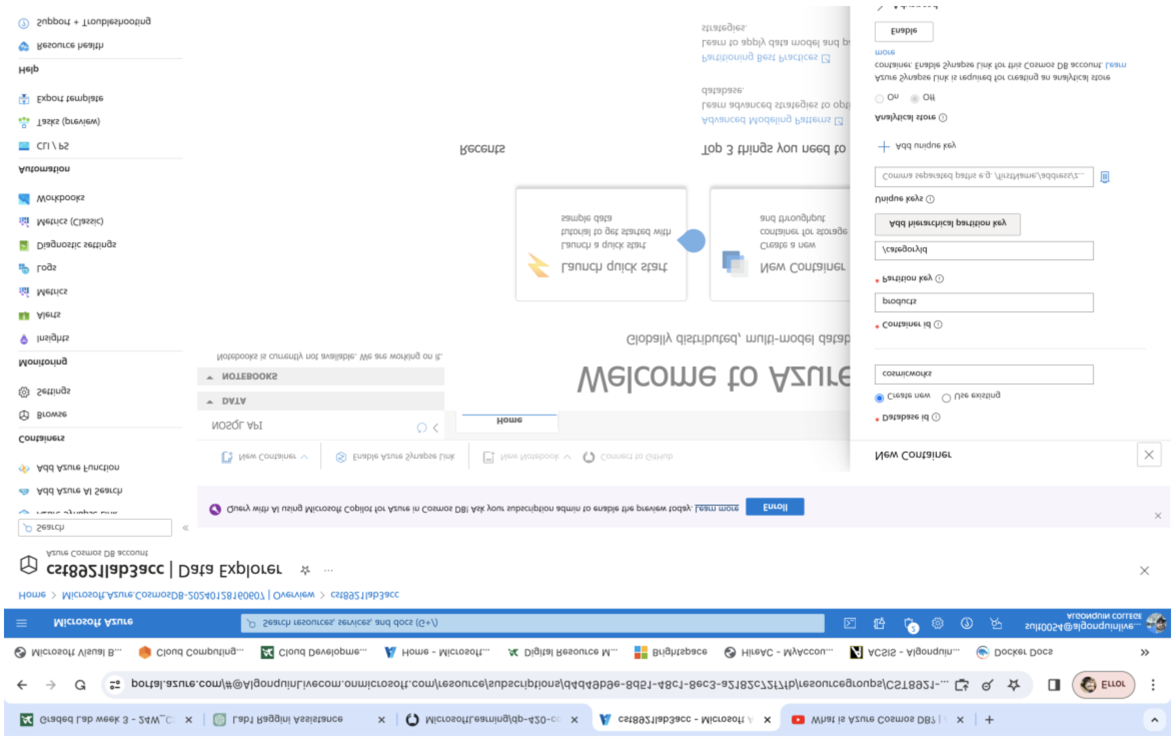
Networking

Connectivity method	All networks
Minimum TLS Protocol	TLS 1.2

3. **Create Containers:**

- In Data Explorer, create a new database "cosmicworks" and containers "products" and "productslease" with specified settings.

Setting	Value
Database id	***Use existing | cosmicworks***
Container id	***products***
Partition key	***/categoryId***



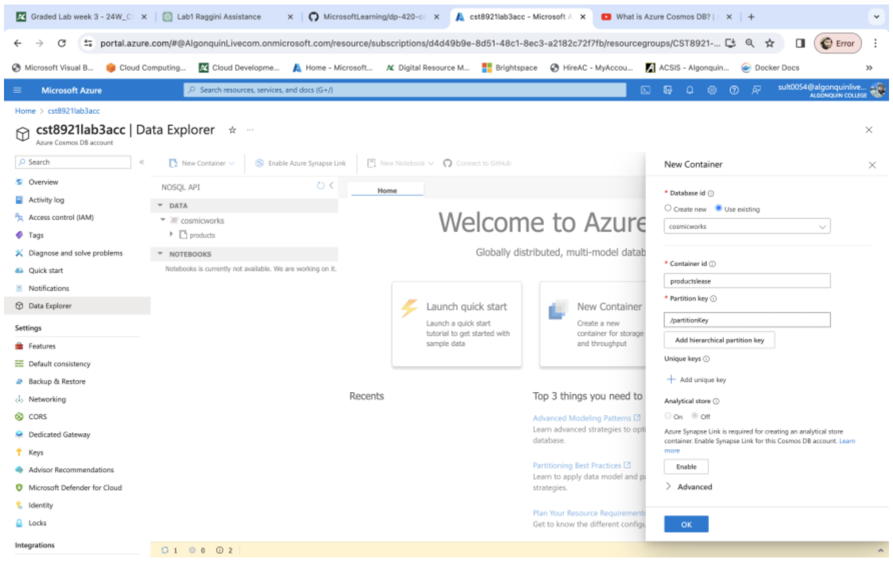
New Container

Database id: [✓ Create new] Use existing — cosmicworks

Container id: products

Partition key: /categoryId

Setting	Value
Database id	*Use existing | cosmicworks*
Container id	`productslease`
Partition key	***/partitionKey***



cst8921lab3acc | Data Explorer

New Container

* Database id
Create new [✓ Use existing]
cosmicworks

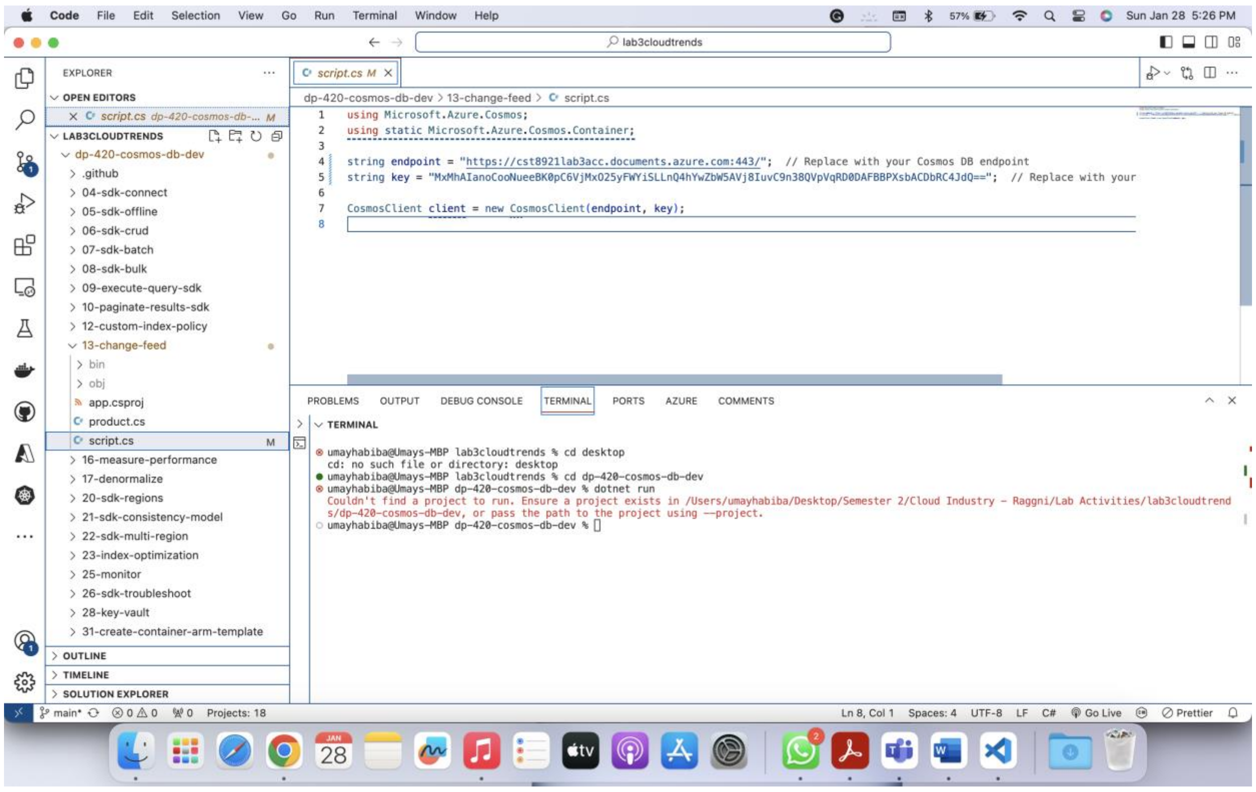
* Container id
productslease

* Partition key
/partitionKey

Analytical store
On [✓ Off]

Azure Synapse Link is required for creating an analytical store container. Enable Synapse Link for this Cosmos DB account. Learn more

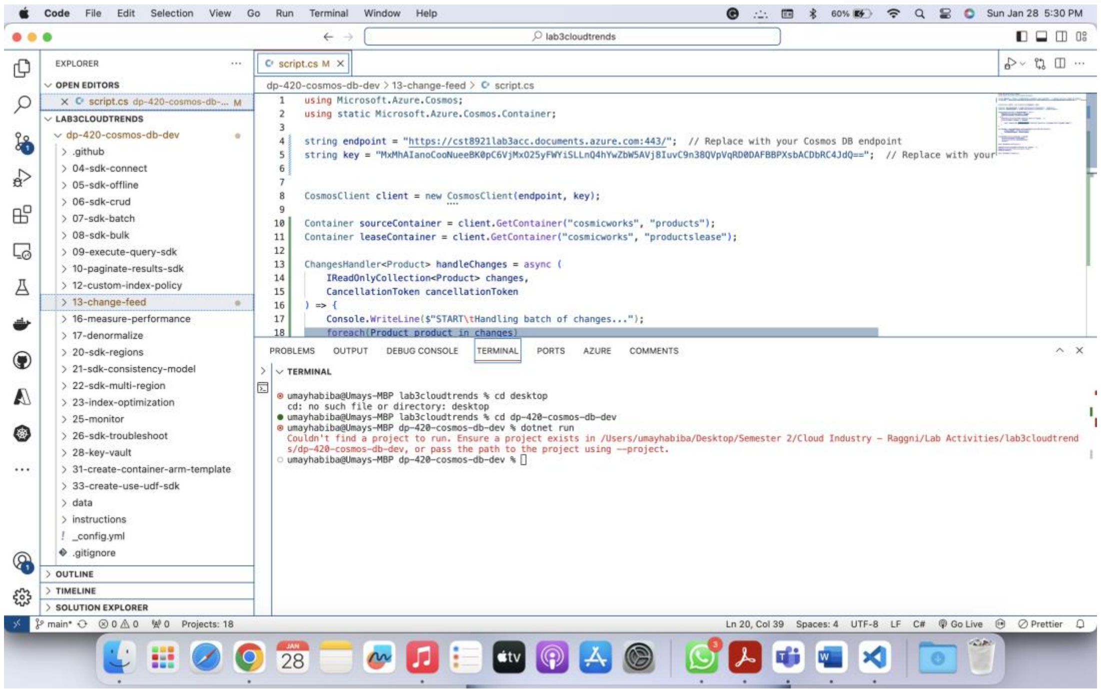
4. **Update Visual Studio Code:**
   - Open the **product.cs** and **script.cs** code files.
   - Update the variables **endpoint** and **key** with Cosmos DB account details.
   - Implement the provided code in **script.cs**.



dp-420-cosmos-db-dev > 13-change-feed > script.cs

```
using Microsoft.Azure.Cosmos;
using static Microsoft.Azure.Cosmos.Container;

string endpoint = "https://cst8921lab3acc.documents.azure.com:443/";  // Replace with your Cosmos DB endpoint
string key = "MxMhAIanoCooNueeBK0pC6VjMx025yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==";  // Replace with your

CosmosClient client = new CosmosClient(endpoint, key);

```

```
umayhabiba@Umays-MBP lab3cloudtrends % cd desktop
cd: no such file or directory: desktop
umayhabiba@Umays-MBP lab3cloudtrends % cd dp-420-cosmos-db-dev
umayhabiba@Umays-MBP dp-420-cosmos-db-dev % dotnet run
Couldn't find a project to run. Ensure a project exists in /Users/umayhabiba/Desktop/Semester 2/Cloud Industry - Raggni/Lab Activities/lab3cloudtrend
s/dp-420-cosmos-db-dev, or pass the path to the project using --project.
umayhabiba@Umays-MBP dp-420-cosmos-db-dev % 
```

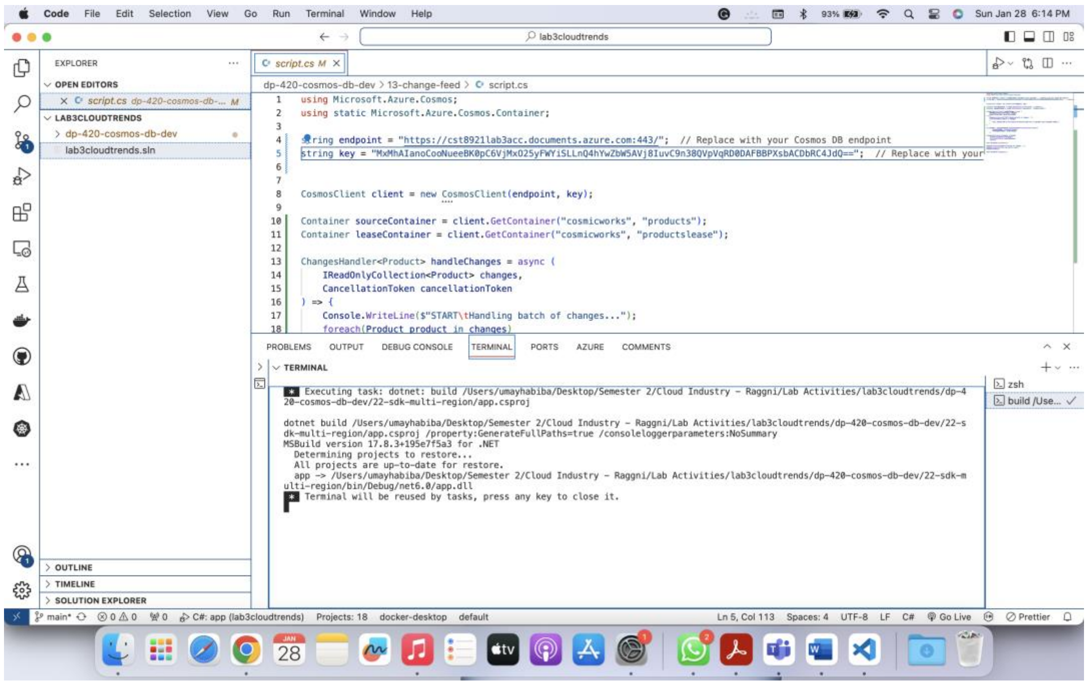


```
using Microsoft.Azure.Cosmos;
using static Microsoft.Azure.Cosmos.Container;

string endpoint = "https://cst8921lab3acc.documents.azure.com:443/";  // Replace with your Cosmos DB endpoint
string key = "MxMhAIanoCooNueeBK0pC6VjMx025yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==";  // Replace with your


CosmosClient client = new CosmosClient(endpoint, key);

Container sourceContainer = client.GetContainer("cosmicworks", "products");
Container leaseContainer = client.GetContainer("cosmicworks", "productslease");

ChangesHandler<Product> handleChanges = async (
    IReadOnlyCollection<Product> changes,
    CancellationToken cancellationToken
) => {
    Console.WriteLine($"START\tHandling batch of changes...");
    foreach(Product product in changes)
```

```
umayhabiba@Umays-MBP lab3cloudtrends % cd desktop
cd: no such file or directory: desktop
umayhabiba@Umays-MBP lab3cloudtrends % cd dp-420-cosmos-db-dev
umayhabiba@Umays-MBP dp-420-cosmos-db-dev % dotnet run
Couldn't find a project to run. Ensure a project exists in /Users/umayhabiba/Desktop/Semester 2/Cloud Industry - Raggni/Lab Activities/lab3cloudtrend
s/dp-420-cosmos-db-dev, or pass the path to the project using --project.
umayhabiba@Umays-MBP dp-420-cosmos-db-dev % 
```

5. **Build and Run Application:**

    - Open an integrated terminal in Visual Studio Code.
    - Build and run the project using the command: bashCopy code dotnet run

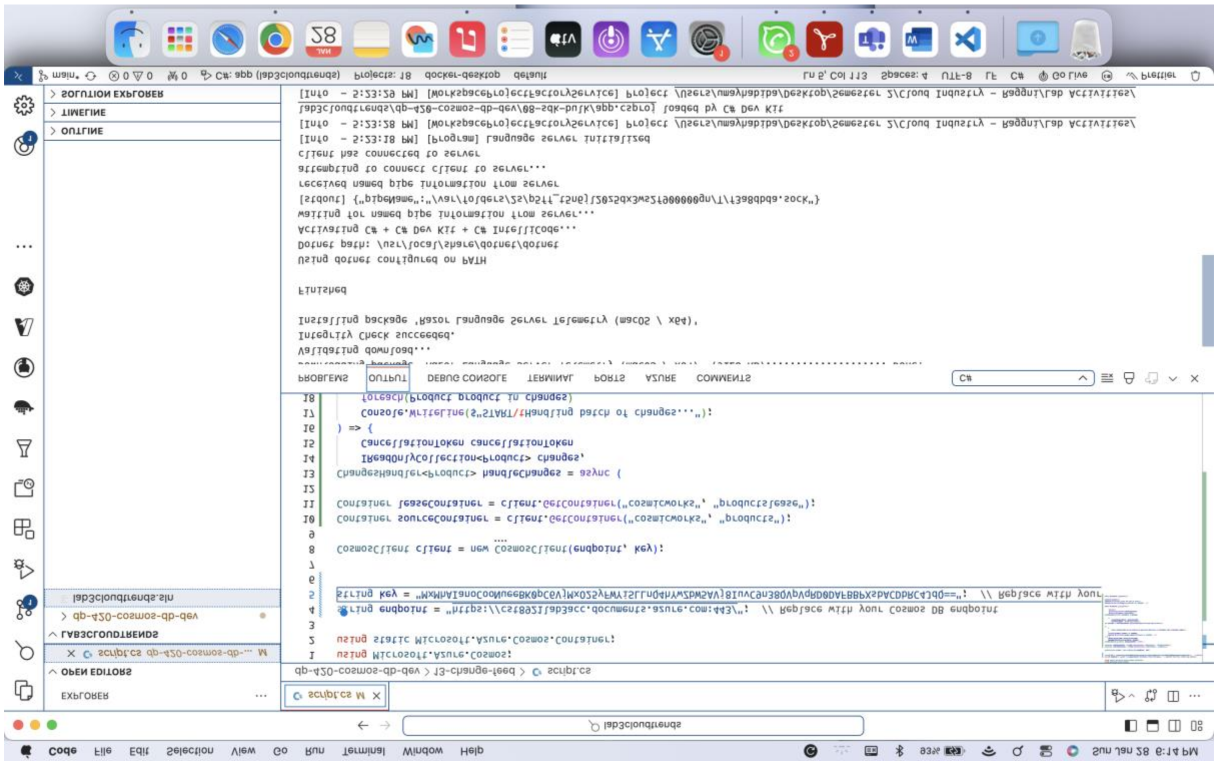


```
using Microsoft.Azure.Cosmos;
using static Microsoft.Azure.Cosmos.Container;

string endpoint = "https://cst8921lab3acc.documents.azure.com:443/";  // Replace with your Cosmos DB endpoint
string key = "MxMhAIanoCooNueeBK0pC6VjMx025yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==";  // Replace with your


CosmosClient client = new CosmosClient(endpoint, key);

Container sourceContainer = client.GetContainer("cosmicworks", "products");
Container leaseContainer = client.GetContainer("cosmicworks", "productslease");

ChangesHandler<Product> handleChanges = async (
    IReadOnlyCollection<Product> changes,
    CancellationToken cancellationToken
) => {
    Console.WriteLine($"START\tHandling batch of changes...");
    foreach(Product product in changes)
```

```
 *  Executing task: dotnet: build /Users/umayhabiba/Desktop/Semester 2/Cloud Industry - Raggni/Lab Activities/lab3cloudtrends/dp-4
20-cosmos-db-dev/22-sdk-multi-region/app.csproj

dotnet build /Users/umayhabiba/Desktop/Semester 2/Cloud Industry - Raggni/Lab Activities/lab3cloudtrends/dp-420-cosmos-db-dev/22-s
dk-multi-region/app.csproj /property:GenerateFullPaths=true /consoleloggerparameters:NoSummary
MSBuild version 17.8.3+195e7f5a3 for .NET
  Determining projects to restore...
  All projects are up-to-date for restore.
  app -> /Users/umayhabiba/Desktop/Semester 2/Cloud Industry - Raggni/Lab Activities/lab3cloudtrends/dp-420-cosmos-db-dev/22-sdk-m
ulti-region/bin/Debug/net6.0/app.dll
 *  Terminal will be reused by tasks, press any key to close it.
```

AhmadSultan_041034155

- Observe the terminal output.



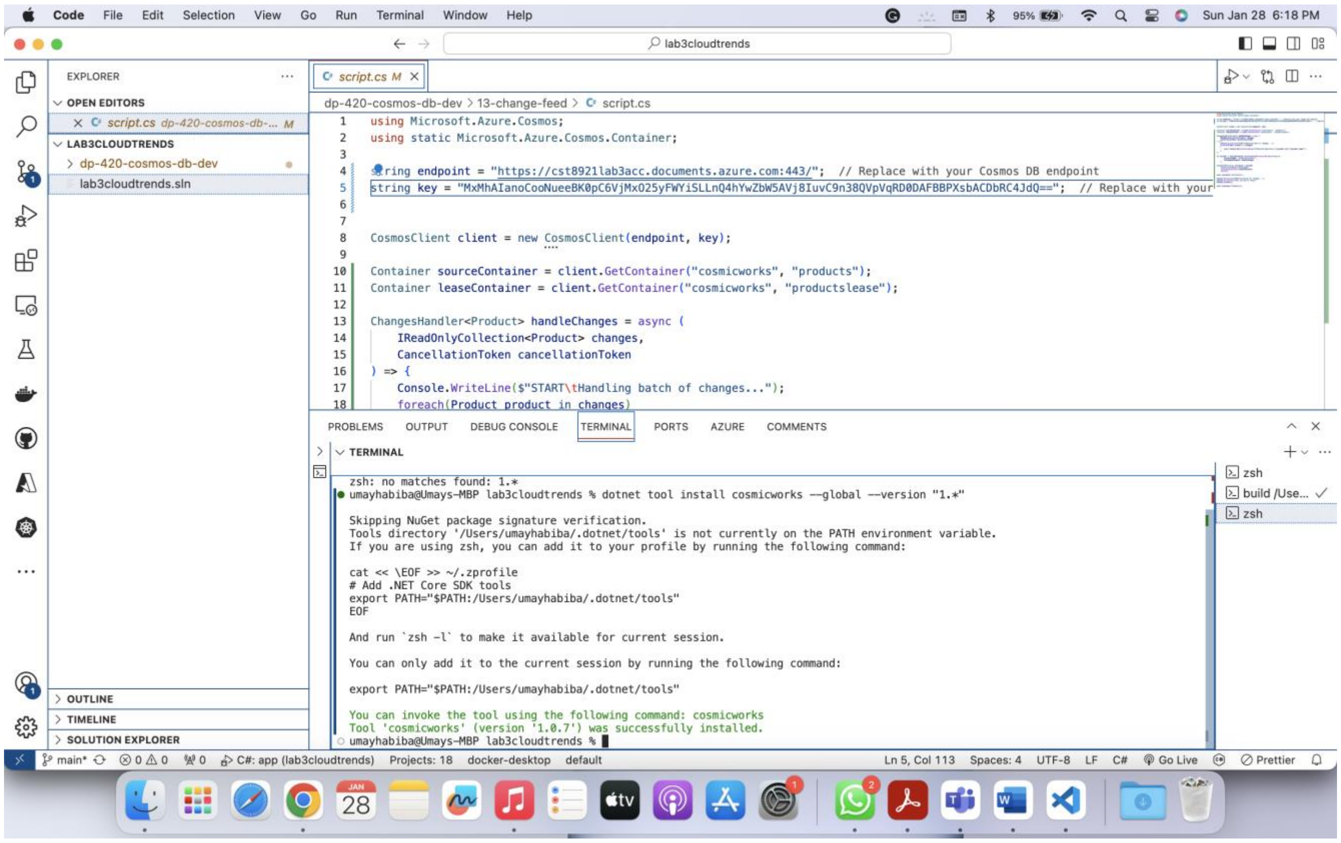
```
Validating download...
Integrity Check succeeded.
Installing package 'Razor Language Server Telemetry (macOS / x64)'

Finished

Using dotnet configured on PATH
Dotnet path: /usr/local/share/dotnet/dotnet
Activating C# + C# Dev Kit + C# IntelliCode...
Waiting for named pipe information from server...
[stdout] {"pipeName":"/var/folders/2s/p5ff_f5nj120z5dx3ws27f000000gn/T/f388dbda.sock"}
Received named pipe information from server
Attempting to connect client to server...
Client has connected to server
[Info  - 5:23:18 PM] [Program] Language server initialized
[Info  - 5:23:28 PM] [WorkspaceProjectFactoryService] Project /Users/umaymabiba/Desktop/Semester 2/Cloud Industry - Raggui/Lab Activities/
lab3cloudtrends/dp-420-cosmos-db-dev/08-sdk-bulk/app.csproj loaded by C# Dev Kit
[Info  - 5:23:29 PM] [WorkspaceProjectFactoryService] Project /Users/umaymabiba/Desktop/Semester 2/Cloud Industry - Raggui/Lab Activities/
```

```csharp
using Microsoft.Azure.Cosmos;
using static Microsoft.Azure.Cosmos.Container;

string endpoint = "https://cst8921lab3acc.documents.azure.com:443/";  // Replace with your Cosmos DB endpoint
string key = "MxMhAIanoCooNueeBK9pC6VjMx0Z5yFWYiSLLn04hYWZbWSAVj8IuvC6n38pVpqRD0DAFBBPXsbACDbRC4JdQ==";  // Replace with your

CosmosClient client = new CosmosClient(endpoint, key);

Container sourceContainer = client.GetContainer("cosmicworks", "products");
Container leaseContainer = client.GetContainer("cosmicworks", "productslease");

ChangesHandler<Product> handleChanges = async (
    IReadOnlyCollection<Product> changes,
    CancellationToken cancellationToken
) => {
    Console.WriteLine($"START\tHandling batch of changes...");
    foreach(Product product in changes)
```

6. **Seed Cosmos DB Account:**

- In a new terminal, install the cosmicworks command-line tool:

bashCopy code : dotnet tool install cosmicworks --global --version 1.*



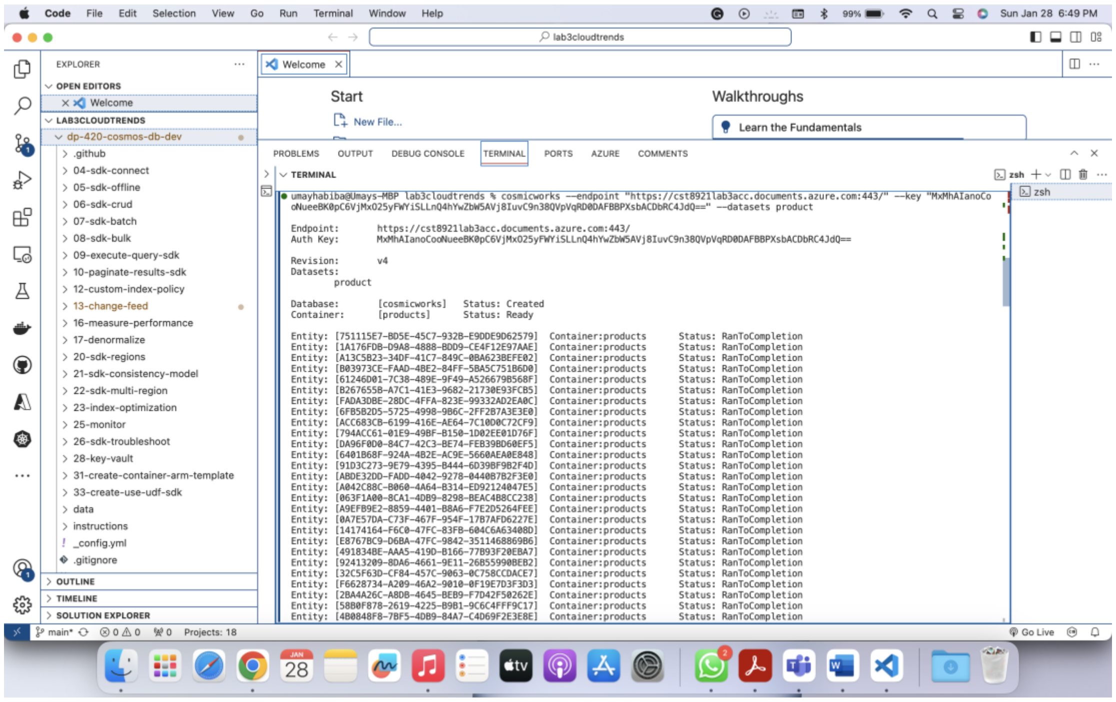
```
using Microsoft.Azure.Cosmos;
using static Microsoft.Azure.Cosmos.Container;

string endpoint = "https://cst8921lab3acc.documents.azure.com:443/";  // Replace with your Cosmos DB endpoint
string key = "MxMhAIanoCooNueeBK0pC6VjMx025yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==";  // Replace with your


CosmosClient client = new CosmosClient(endpoint, key);

Container sourceContainer = client.GetContainer("cosmicworks", "products");
Container leaseContainer = client.GetContainer("cosmicworks", "productslease");

ChangesHandler<Product> handleChanges = async (
    IReadOnlyCollection<Product> changes,
    CancellationToken cancellationToken
) => {
    Console.WriteLine($"START\tHandling batch of changes...");
    foreach(Product product in changes)
```

```
zsh: no matches found: 1.*
umayhabiba@Umays-MBP lab3cloudtrends % dotnet tool install cosmicworks --global --version "1.*"

Skipping NuGet package signature verification.
Tools directory '/Users/umayhabiba/.dotnet/tools' is not currently on the PATH environment variable.
If you are using zsh, you can add it to your profile by running the following command:

cat << \EOF >> ~/.zprofile
# Add .NET Core SDK tools
export PATH="$PATH:/Users/umayhabiba/.dotnet/tools"
EOF

And run `zsh -l` to make it available for current session.

You can only add it to the current session by running the following command:

export PATH="$PATH:/Users/umayhabiba/.dotnet/tools"

You can invoke the tool using the following command: cosmicworks
Tool 'cosmicworks' (version '1.0.7') was successfully installed.
umayhabiba@Umays-MBP lab3cloudtrends %
```

- Seed the Cosmos DB account with sample data:

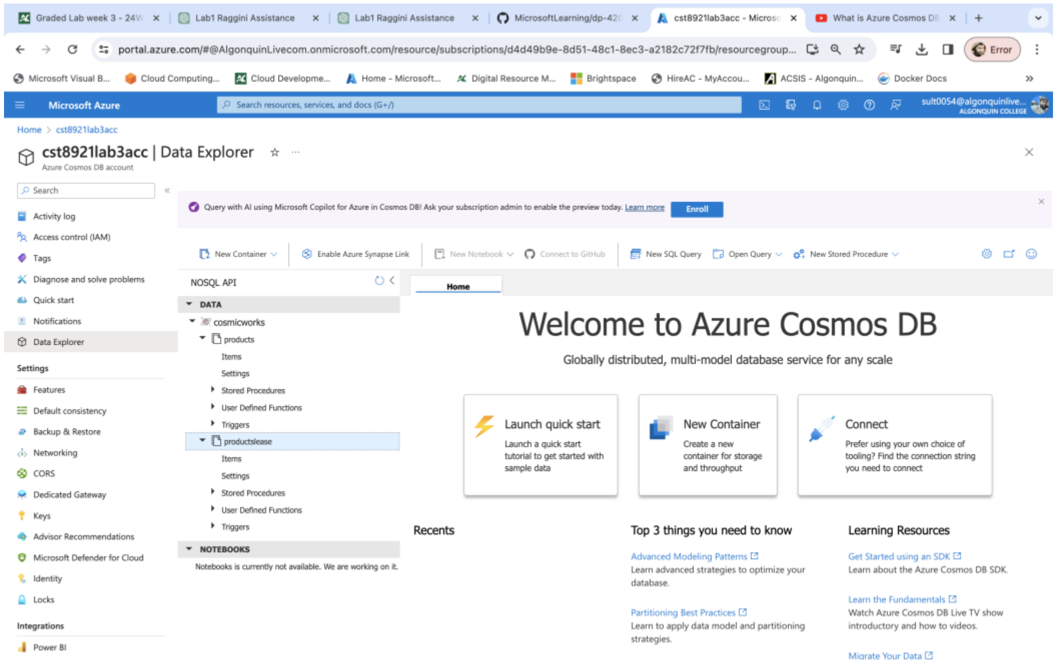
bashCopy code

- cosmicworks --endpoint "https://cst8921lab3acc.documents.azure.com:443/" --key "MxMhAIanoCooNueeBK0pC6VjMxO25yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==" --datasets productWait for the command to finish populating the account.



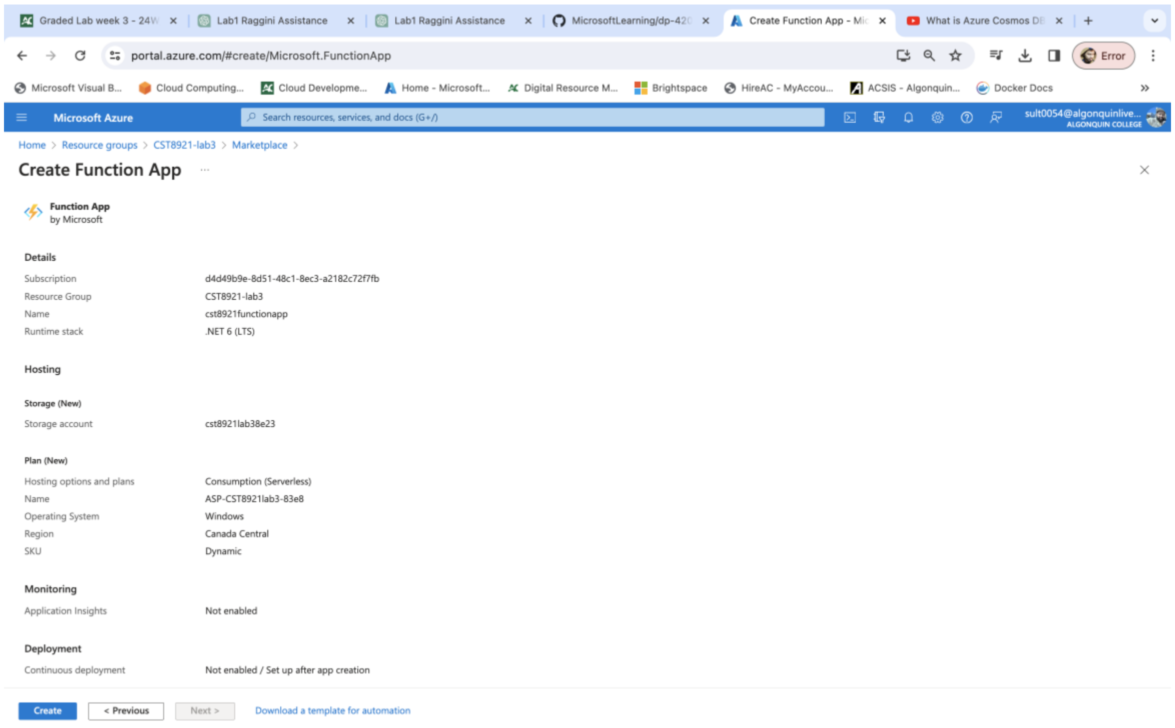
```
umayhabiba@Umays-MBP lab3cloudtrends % cosmicworks --endpoint "https://cst8921lab3acc.documents.azure.com:443/" --key "MxMhAIanoCo
oNueeBK0pC6VjMxO25yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==" --datasets product

Endpoint:       https://cst8921lab3acc.documents.azure.com:443/
Auth Key:       MxMhAIanoCooNueeBK0pC6VjMxO25yFWYiSLLnQ4hYwZbW5AVj8IuvC9n38QVpVqRD0DAFBBPXsbACDbRC4JdQ==

Revision:       v4
Datasets:
        product

Database:       [cosmicworks]   Status: Created
Container:      [products]      Status: Ready

Entity: [751115E7-BD5E-45C7-932B-E9DDE9D62579]  Container:products      Status: RanToCompletion
Entity: [1A176FDB-D9A8-4888-BDD9-CE4F12E97AAE]  Container:products      Status: RanToCompletion
Entity: [A13C5B23-34DF-41C7-849C-0BA623BEFE02]  Container:products      Status: RanToCompletion
Entity: [B03973CE-FAAD-4BE2-84FF-5BA5C751B6D0]  Container:products      Status: RanToCompletion
Entity: [61246D01-7C38-489E-9F49-A526679B568F]  Container:products      Status: RanToCompletion
Entity: [B267655B-A7C1-41E3-9682-21730E93FCB5]  Container:products      Status: RanToCompletion
Entity: [FADA3DBE-28DC-4FFA-823E-99332AD2EA0C]  Container:products      Status: RanToCompletion
Entity: [6FB5B2D5-5725-4998-9B6C-2FF2B7A3E3E0]  Container:products      Status: RanToCompletion
Entity: [ACC683CB-6199-416E-AE64-7C10D0C72CF9]  Container:products      Status: RanToCompletion
Entity: [794ACC61-01E9-49BF-B150-1D02EE01D76F]  Container:products      Status: RanToCompletion
Entity: [DA96F0D0-84C7-42C3-BE74-FEB39BD60EF5]  Container:products      Status: RanToCompletion
Entity: [6401B68F-924A-4B2E-AC9E-5660AEA0E848]  Container:products      Status: RanToCompletion
Entity: [91D3C273-9E79-4395-B444-6D39BF9B2F4D]  Container:products      Status: RanToCompletion
Entity: [ABDE32DD-FADD-4042-9278-0440B7B2F3E0]  Container:products      Status: RanToCompletion
Entity: [A042C88C-B060-4A64-B314-ED92124047E5]  Container:products      Status: RanToCompletion
Entity: [063F1A00-8CA1-4DB9-8298-BEAC4B8CC238]  Container:products      Status: RanToCompletion
Entity: [A9EFB9E2-8859-4401-B8A6-F7E2D5264FEE]  Container:products      Status: RanToCompletion
Entity: [0A7E57DA-C73F-467F-954F-17B7AFD6227E]  Container:products      Status: RanToCompletion
Entity: [14174164-F6C0-47FC-83FB-604C6A63408D]  Container:products      Status: RanToCompletion
Entity: [E8767BC9-D6BA-47FC-9842-3511468869B6]  Container:products      Status: RanToCompletion
Entity: [491834BE-AAA5-419D-B166-77B93F20EBA7]  Container:products      Status: RanToCompletion
Entity: [92413209-8DA6-4661-9E11-26B55990BEB2]  Container:products      Status: RanToCompletion
Entity: [32C5F63D-CF84-457C-9063-0C758CCDACE7]  Container:products      Status: RanToCompletion
Entity: [F6628734-A209-46A2-9010-0F19E7D3F3D3]  Container:products      Status: RanToCompletion
Entity: [2BA4A26C-A8DB-4645-BEB9-F7D42F50262E]  Container:products      Status: RanToCompletion
Entity: [58B0F878-2619-4225-B9B1-9C6C4FFF9C17]  Container:products      Status: RanToCompletion
Entity: [4B0848F8-7BF5-4DB9-84A7-C4D69F2E3E8E]  Container:products      Status: RanToCompletion
```

7. **Verify Changes:**
    - Observe the terminal output from the .NET application for Detected Operation messages.
    - Verify changes in "products" and "productslease" containers in Cosmos DB.


Graded Lab week 3 - 24W
Lab1 Raggini Assistance
Lab1 Raggini Assistance
MicrosoftLearning/dp-420
cst8921lab3acc - Microsoft
What is Azure Cosmos DB
portal.azure.com/#@AlgonquinLivecom.onmicrosoft.com/resource/subscriptions/d4d49b9e-8d51-48c1-8ec3-a2182c72f7fb/resourcegroup...
Microsoft Azure
Search resources, services, and docs (G+/)
sult0054@algonquinlive...
ALGONQUIN COLLEGE
Home > cst8921lab3acc
cst8921lab3acc | Data Explorer
Azure Cosmos DB account
Search
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Quick start
Notifications
Data Explorer
Settings
Features
Default consistency
Backup & Restore
Networking
CORS
Dedicated Gateway
Keys
Advisor Recommendations
Microsoft Defender for Cloud
Identity
Locks
Integrations
Power BI
Query with AI using Microsoft Copilot for Azure in Cosmos DB! Ask your subscription admin to enable the preview today. Learn more
Enroll
New Container
Enable Azure Synapse Link
New Notebook
Connect to GitHub
New SQL Query
Open Query
New Stored Procedure
NOSQL API
DATA
cosmicworks
products
Items
Settings
Stored Procedures
User Defined Functions
Triggers
productslease
Items
Settings
Stored Procedures
User Defined Functions
Triggers
NOTEBOOKS
Notebooks is currently not available. We are working on it.
Home
Welcome to Azure Cosmos DB
Globally distributed, multi-model database service for any scale
Launch quick start
Launch a quick start tutorial to get started with sample data
New Container
Create a new container for storage and throughput
Connect
Prefer using your own choice of tooling? Find the connection string you need to connect
Recents
Top 3 things you need to know
Advanced Modeling Patterns
Learn advanced strategies to optimize your database.
Partitioning Best Practices
Learn to apply data model and partitioning strategies.
Learning Resources
Get Started using an SDK
Learn about the Azure Cosmos DB SDK.
Learn the Fundamentals
Watch Azure Cosmos DB Live TV show introductory and how to videos.
Migrate Your Data

8. **Create Azure Function App:**

- Create a new Azure Function app with specified configurations.


portal.azure.com/#create/Microsoft.FunctionApp
Microsoft Azure
Search resources, services, and docs (G+/)
Home > Resource groups > CST8921-lab3 > Marketplace >
Create Function App
Function App
by Microsoft
Create
< Previous
Next >
Download a template for automation

Details

Subscription	d4d49b9e-8d51-48c1-8ec3-a2182c72f7fb
Resource Group	CST8921-lab3
Name	cst8921functionapp
Runtime stack	.NET 6 (LTS)

Hosting

Storage (New)

Storage account	cst8921lab38e23

Plan (New)

Hosting options and plans	Consumption (Serverless)
Name	ASP-CST8921lab3-83e8
Operating System	Windows
Region	Canada Central
SKU	Dynamic

Monitoring

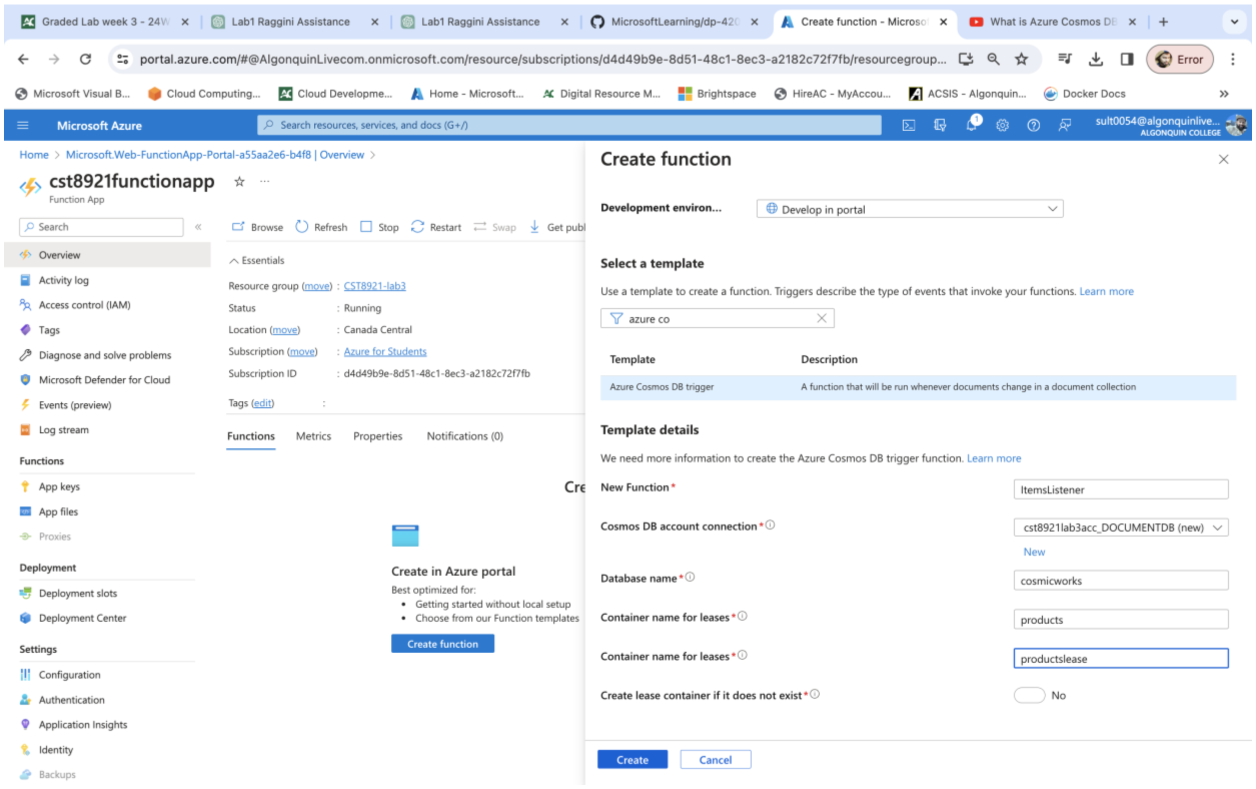
Application Insights	Not enabled

Deployment

Continuous deployment	Not enabled / Set up after app creation

- Navigate to Functions pane and create a new function "ItemsListener" using Azure Cosmos DB trigger template.

Setting	**Value**
Development environment	**Develop in portal**
Select a template	**Azure Cosmos DB trigger**
New Function	**ItemsListener**
Cosmos DB account connection	**Select New | Select Azure Cosmos DB Account | Select the Azure Cosmos DB account you created earlier**
Database name	**cosmicworks**
Collection name	**products**
Collection name for leases	**productslease**
Create lease collection if it does not exist	**No**



Resource group (move) : CST8921-lab3
Status : Running
Location (move) : Canada Central
Subscription (move) : Azure for Students
Subscription ID : d4d49b9e-8d51-48c1-8ec3-a2182c72f7fb

Create function

Development environ... Develop in portal

Select a template

Use a template to create a function. Triggers describe the type of events that invoke your functions. Learn more

azure co

Template	Description
Azure Cosmos DB trigger	A function that will be run whenever documents change in a document collection

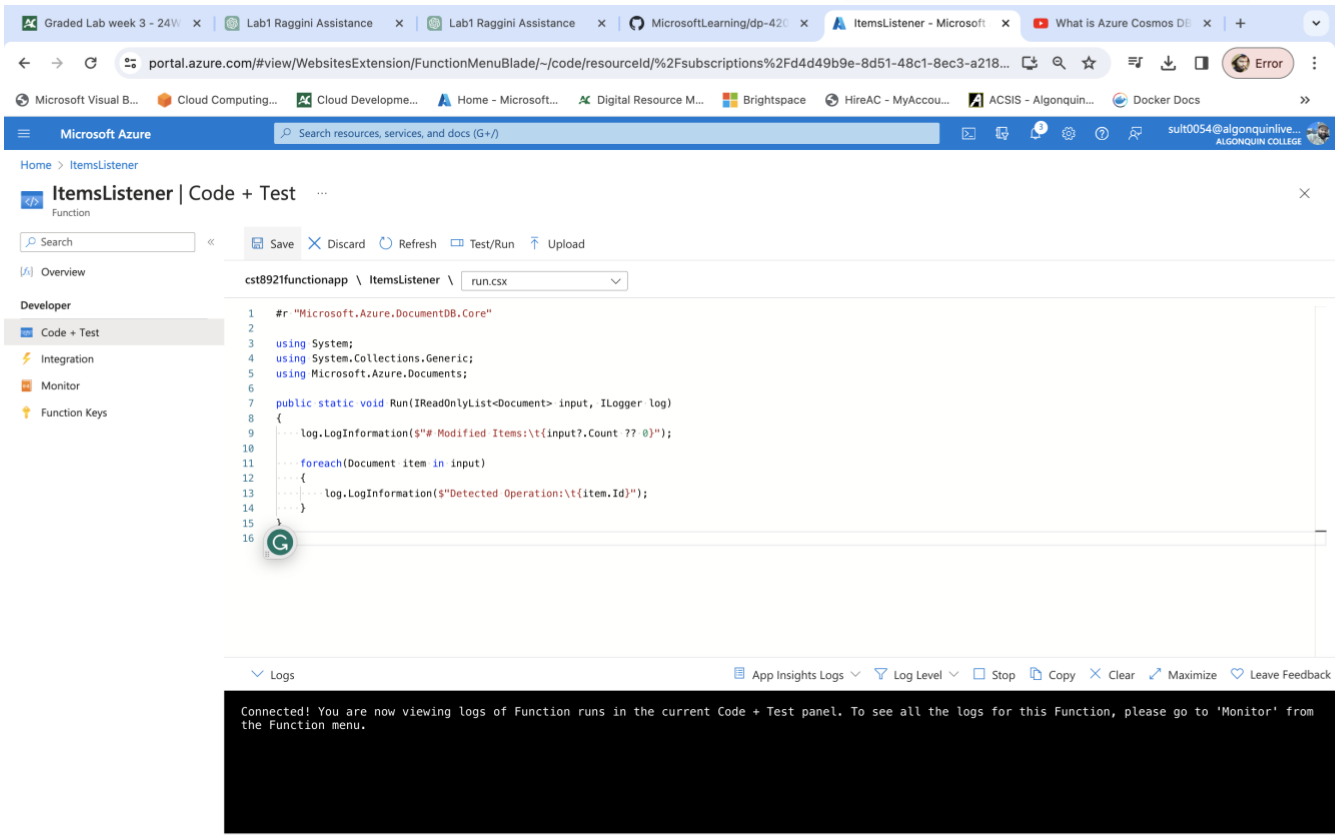
Template details

We need more information to create the Azure Cosmos DB trigger function. Learn more

New Function *	ItemsListener
Cosmos DB account connection *	cst8921lab3acc_DOCUMENTDB (new)
Database name *	cosmicworks
Container name for leases *	products
Container name for leases *	productslease
Create lease container if it does not exist *	No

9. **Implement Function Code:**
   - In the Function pane, navigate to Code + Test.
   - Delete the existing code and use the provided C# code.

```
#r "Microsoft.Azure.DocumentDB.Core"

using System;
using System.Collections.Generic;
using Microsoft.Azure.Documents;

public static void Run(IReadOnlyList<Document> input, ILogger log)
{
   log.LogInformation($"# Modified Items:\t{input?.Count ?? 0}");

   foreach(Document item in input)
   {
      log.LogInformation($"Detected Operation:\t{item.Id}");
   } }
```



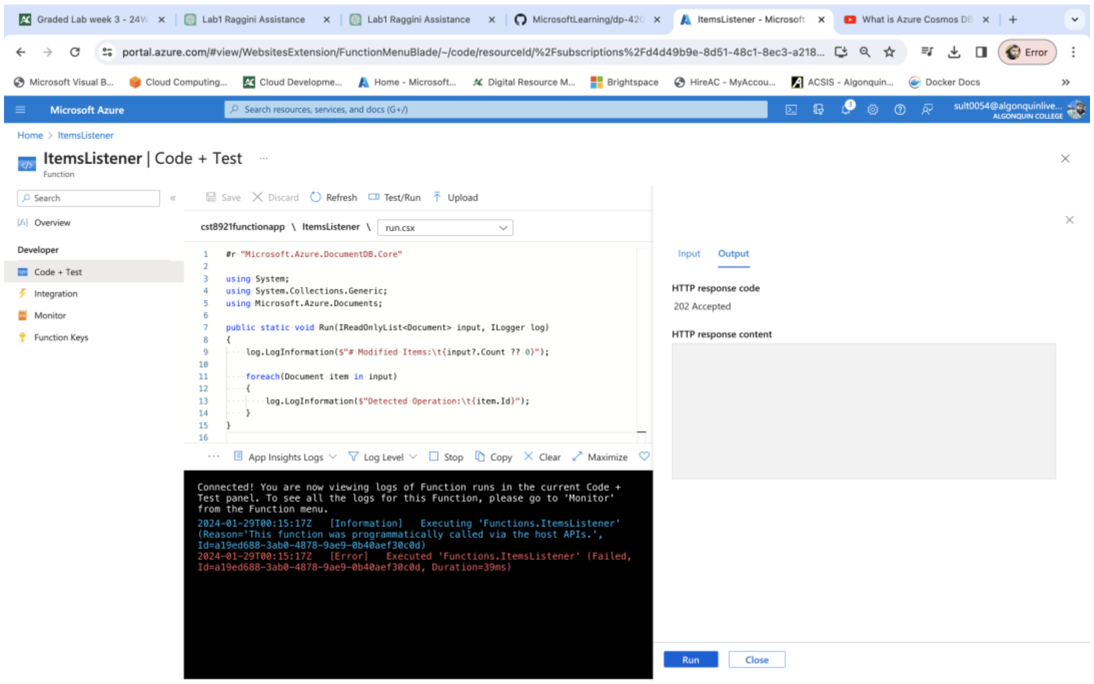
```
#r "Microsoft.Azure.DocumentDB.Core"

using System;
using System.Collections.Generic;
using Microsoft.Azure.Documents;

public static void Run(IReadOnlyList<Document> input, ILogger log)
{
    log.LogInformation($"# Modified Items:\t{input?.Count ?? 0}");

    foreach(Document item in input)
    {
        log.LogInformation($"Detected Operation:\t{item.Id}");
    }
}
```

```
Connected! You are now viewing logs of Function runs in the current Code + Test panel. To see all the logs for this Function, please go to 'Monitor' from the Function menu.
```

- Save the function code.

10. **Generate Items and Observe:**

- Use the cosmicworks command-line tool to seed Cosmos DB again.
- Observe the log output from the Azure Functions application.
- Detected Operation messages should appear in the log



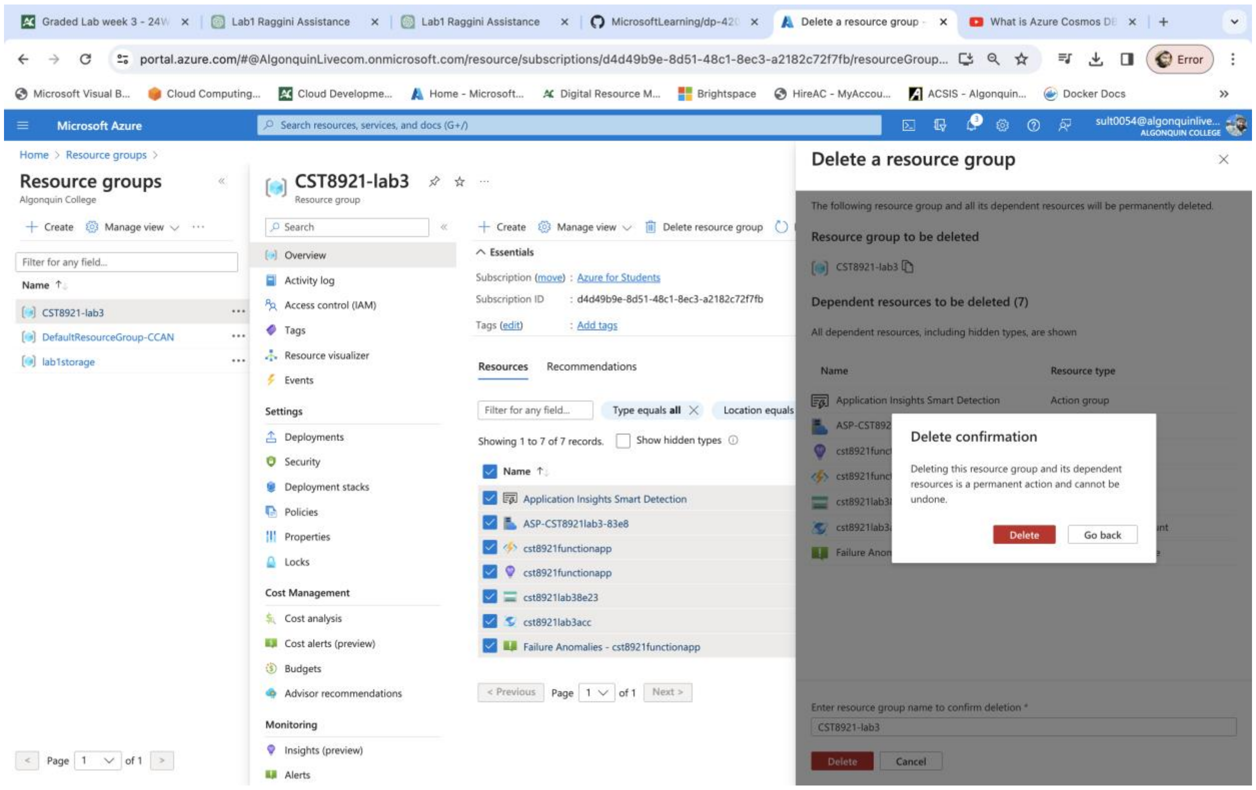
```
#r "Microsoft.Azure.DocumentDB.Core"

using System;
using System.Collections.Generic;
using Microsoft.Azure.Documents;

public static void Run(IReadOnlyList<Document> input, ILogger log)
{
    log.LogInformation($"# Modified Items:\t{input?.Count ?? 0}");

    foreach(Document item in input)
    {
        log.LogInformation($"Detected Operation:\t{item.Id}");
    }
}
```

```
Connected! You are now viewing logs of Function runs in the current Code +
Test panel. To see all the logs for this Function, please go to 'Monitor'
from the Function menu.
2024-01-29T00:15:17Z   [Information]   Executing 'Functions.ItemsListener'
(Reason='This function was programmatically called via the host APIs.',
Id=a19ed688-3ab0-4878-9ae9-0b40aef30c0d)
2024-01-29T00:15:17Z   [Error]   Executed 'Functions.ItemsListener' (Failed,
Id=a19ed688-3ab0-4878-9ae9-0b40aef30c0d, Duration=39ms)
```

Added another library `using Microsoft.Extensions.Logging;`

11. **Clean Up:**

- Delete all resources created during the lab.



Subscription (move) : Azure for Students
Subscription ID : d4d49b9e-8d51-48c1-8ec3-a2182c72f7fb
Tags (edit) : Add tags

Showing 1 to 7 of 7 records.

Name
Application Insights Smart Detection
ASP-CST8921lab3-83e8
cst8921functionapp
cst8921functionapp
cst8921lab38e23
cst8921lab3acc
Failure Anomalies - cst8921functionapp

Delete a resource group

The following resource group and all its dependent resources will be permanently deleted.

Resource group to be deleted

CST8921-lab3

Dependent resources to be deleted (7)

All dependent resources, including hidden types, are shown

Delete confirmation

Deleting this resource group and its dependent resources is a permanent action and cannot be undone.

Delete | Go back

Enter resource group name to confirm deletion *

CST8921-lab3

Delete | Cancel

Results

Students successfully implemented a .NET application using Cosmos DB's change feed processor, demonstrating asynchronous and incremental processing of persistent changes.

Azure Functions were seamlessly integrated with Cosmos DB triggers, creating an "ItemsListener" function that reacts to modifications in the "products" container, showcasing dynamic response capabilities.

Through comprehensive testing, users observed the synchronization of change feed events, gaining practical insights into managing distributed, globally-available databases with high SLAs in a serverless capacity mode.