Bookstore assignment

**Technology Stack: -** python 3.7, json, fastAPI, Azure Cosmos DB(DB), request

Repository: - https://github.com/devendrajha/bookstore.git

**Steps to execute this framework: -**

cd bookstore\app

python3 main.py

Project Structure :-

Bookstore 🡪app – All serverside component(producing API’s)

🡪client -- client code(consuming api’s)

🡪monitoring --- add jmetrix script and serverside response matrix

**Created Below List of API :-**

|  |  |  |
| --- | --- | --- |
|  | Get bookdetails | http://127.0.0.1:8000/bookstore/bookdetails/{id}, |
|  | Get all books | http://127.0.0.1:8000/bookstore/getallbooks/ |
|  | Get books based on count | http://127.0.0.1:8000/bookstore/page/{limit} |
|  | Add new book entry | http://127.0.0.1:8000/bookstore/addbook/ |
|  | Update books from | http://127.0.0.1:8000/bookstore/updatebook/{id} |
|  | search books | http://127.0.0.1:8000/bookstore/search/{search} |
|  | Delete book | http://127.0.0.1:8000/bookstore/remove/{id} |

And For all api create consumer(client) code in client dir

Sample json for post and put call.

{

    "title": "Data structure",

    "author": "Devendra kumar Jha",

    "book\_description": " Data structure algo. engineering",

    "added\_time": 2021,

    "cost": 153

}

Note :- While Implementing search API ( "http://127.0.0.1:8000/bookstore/search/{search})

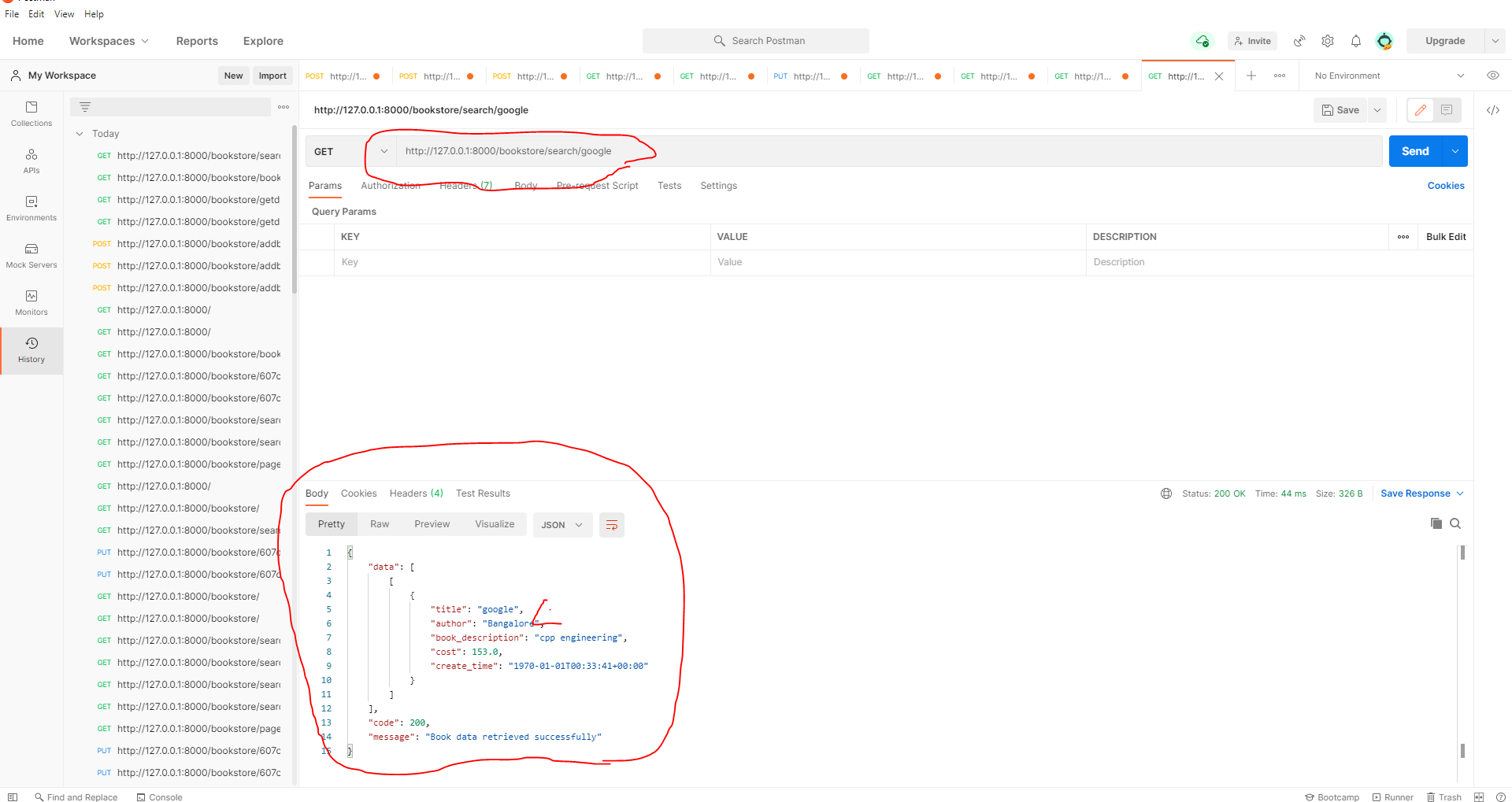
After some search, I found out that MongoDB provided on Azure (DOCUMENTDB...) **does not support text indexing.**

Refrence <https://docs.microsoft.com/en-us/azure/documentdb/documentdb-indexing-policies>

**My Implemented API Works fine using mongodb I tested in my locale to implement this I created text Index**

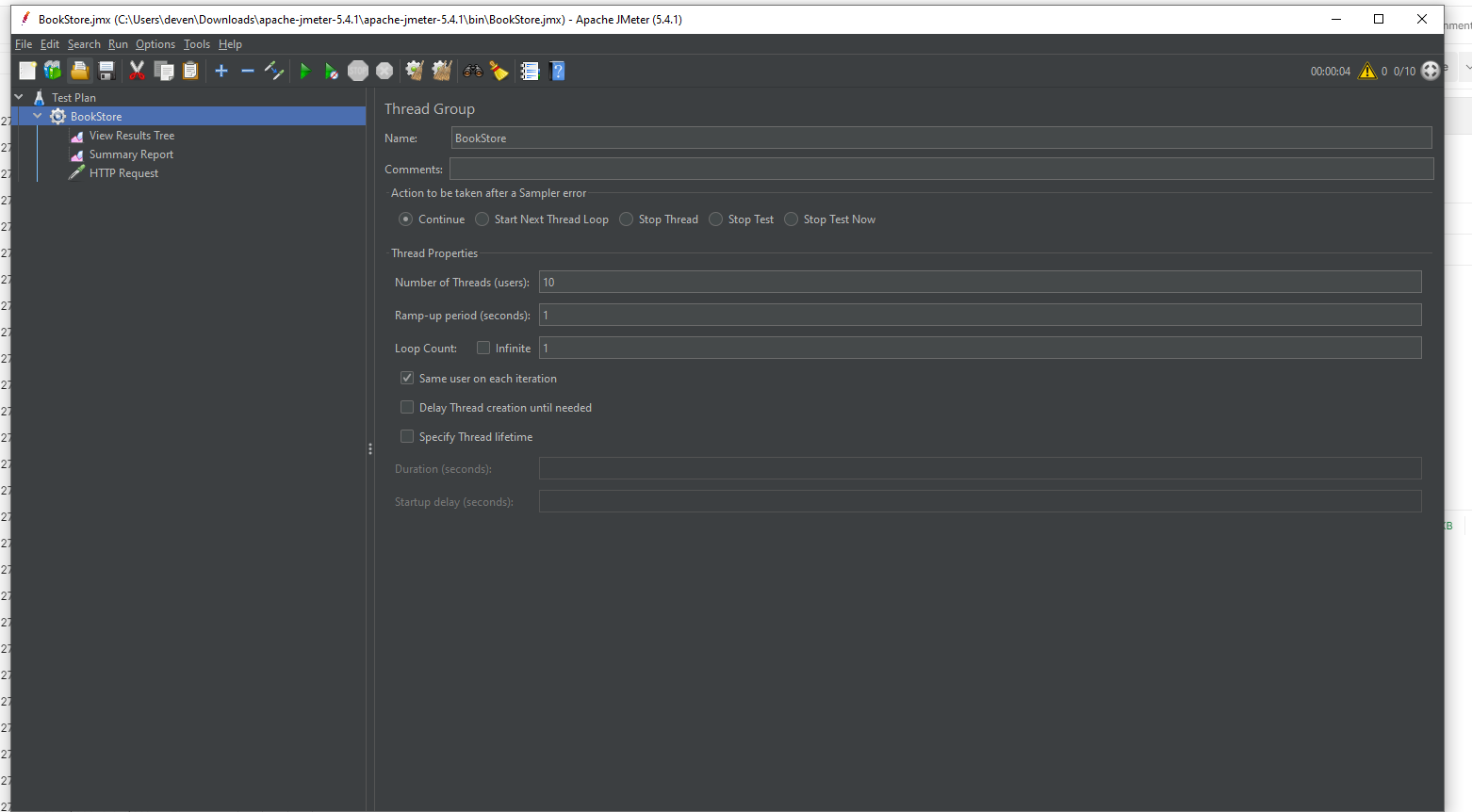
**db.book\_collection.createIndex( { title: "text", author: "text"} )**

below screen of working search API in using mongodb in my locale system.

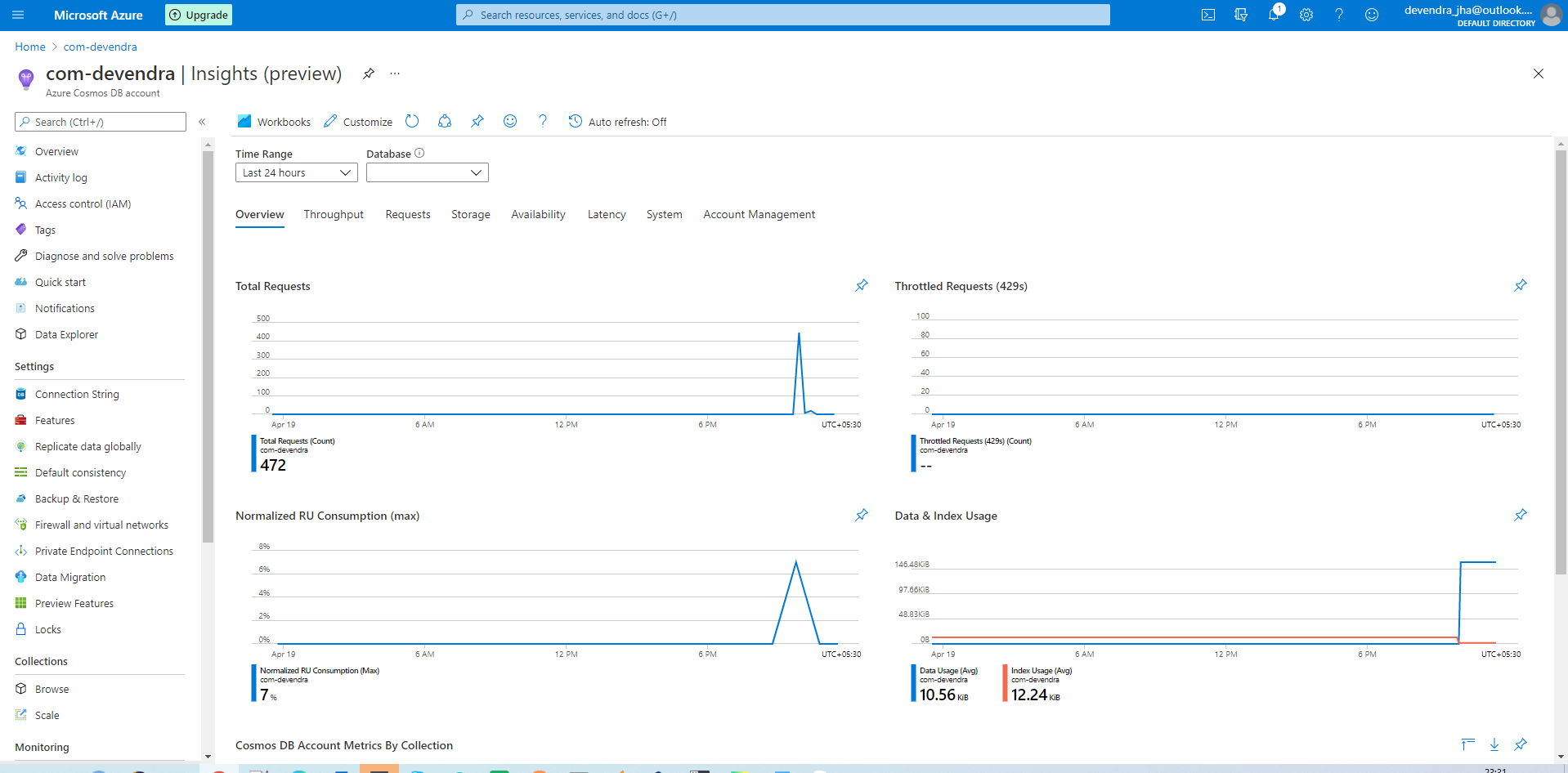


API Monitoring :-

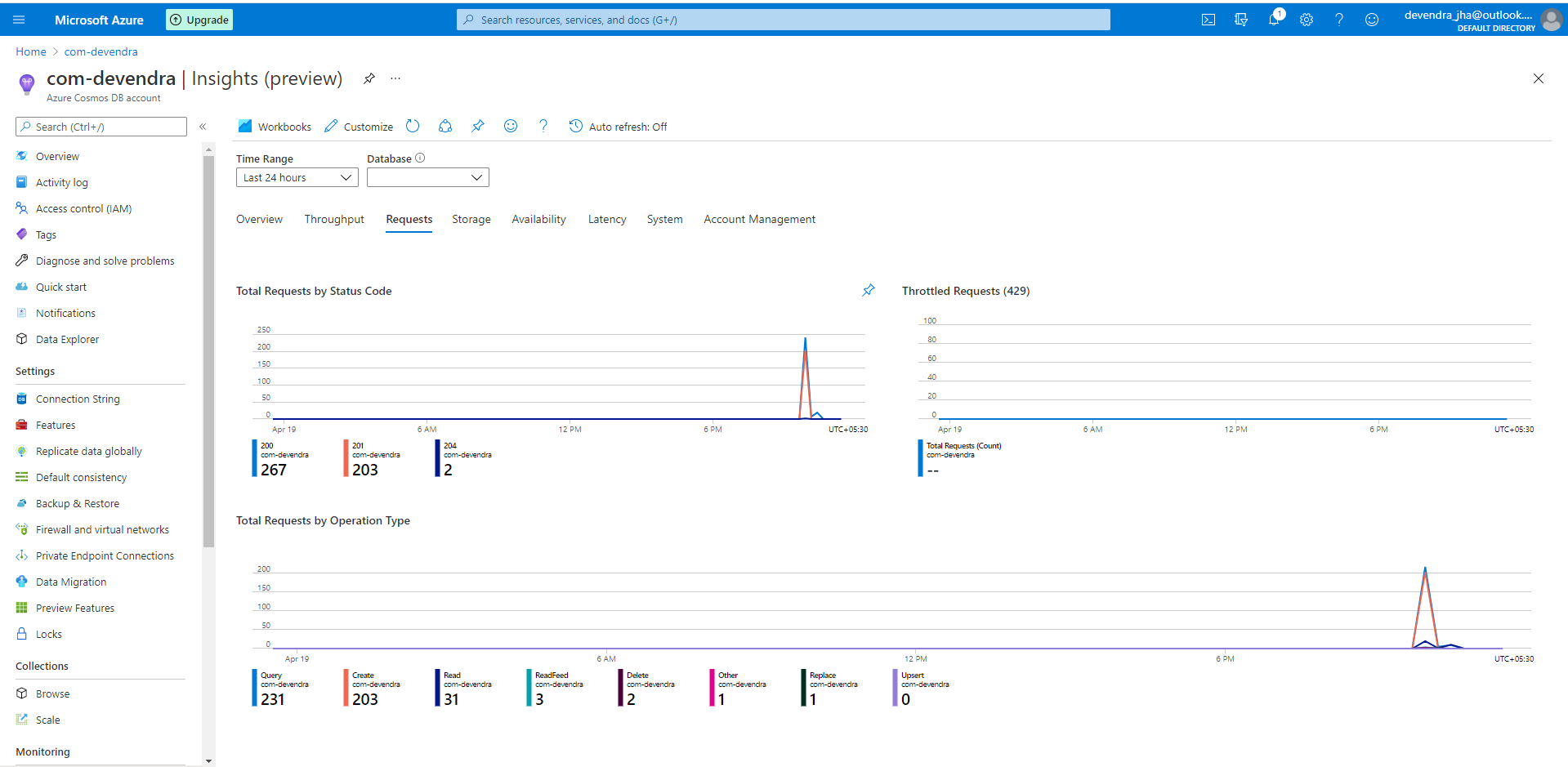
For API Monitoring I have used **Jmeter .** using JMeter I am sending 10 request and capturing matrix



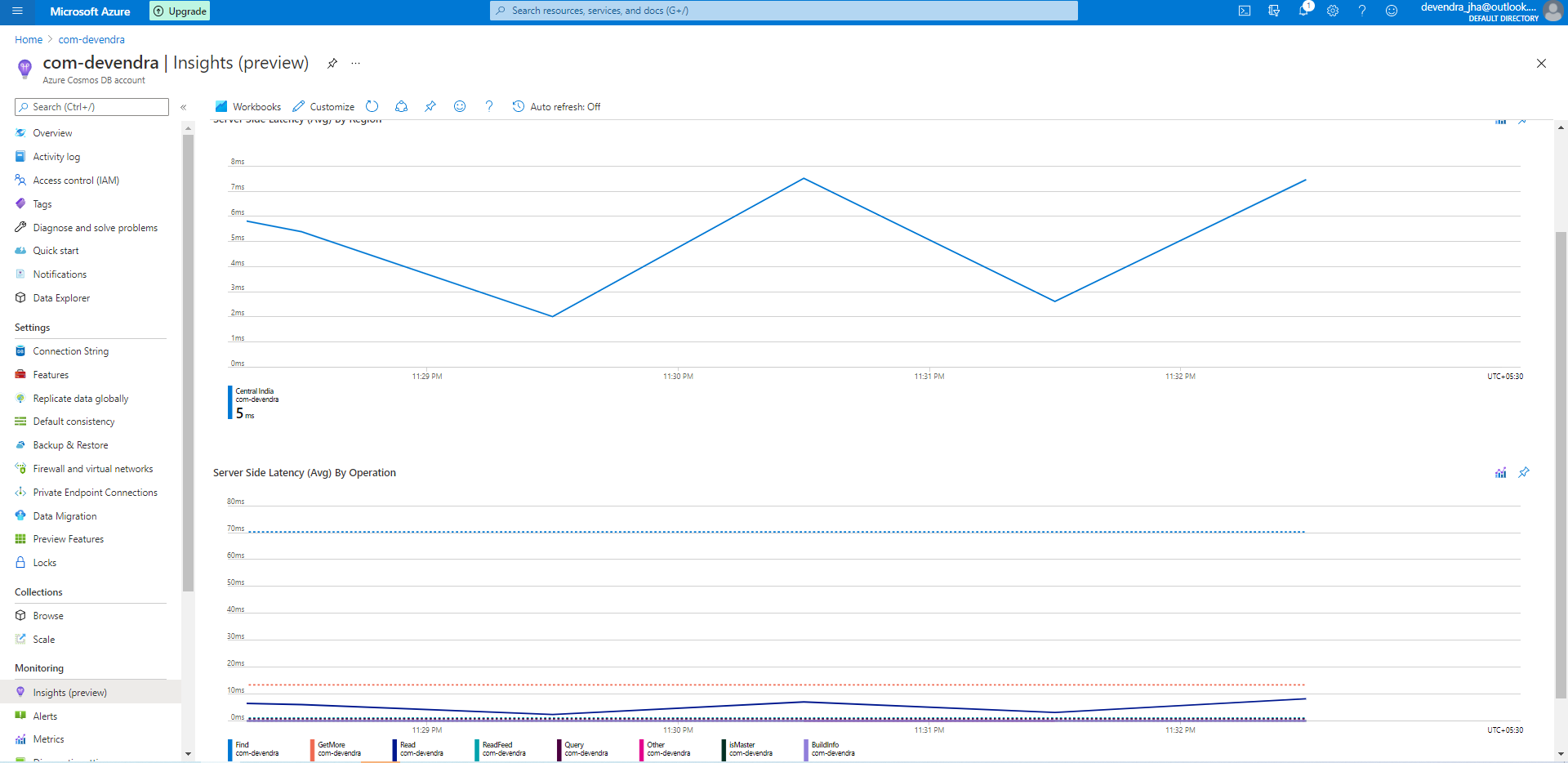
Cloud Usage overview: -



Requests



Server-side Latency: -



All Monitoring reports added in monitoring folder in project