Mini Project

DOTNET+AZURE

Create a **Web API Project** to store Product Information. Use Entity Framework to store the product information in the database. The user should be able to perform all the CRUD Operations. Configure **GET, POST, PUT and DELETE**.

The Product Entity should have the following properties: -

- Product ID
- Product Name
- Price
- Brand
- Manufacturing Date
- Expiration Date

Use Data Annotations to: -

- Mark the Primary Key
- Make ProductName Mandatory
- Make Price a Number

Create a jQuery and AJAX Client to consume the Web API and show the result.

1. Start a new project in visual studio, with project type as ASP.NET Core Web API.

2. Create a new Folder called Models with class Products with the given specifications, add all the validation attributes on to the fields of the Product class.

3. Added required entity framework libraries using NuGet package manager.

4. Performed database migration operations on PS console.

5. Created a controller by right clicking on Controller and selecting the option Controller, create a controller with the created model class by using scaffolding feature of visual studio.

6. Tested the API with the default end point weather forecast by publishing it into the Azure App Services.

7. The jQuery AJAX calls (REST Client) or Index.html page is served on the same host, in the folder on Server at the path Static Files/index.html. The html page is designed to perform all the CRUD operations on the Created WEB API.

Azure Hosting: ·

Host the WEB API in azure and consume the same using jQuery Client.

·Configure Scale out by adding rules for custom scaling

·Configure Deployment slots for staging and production

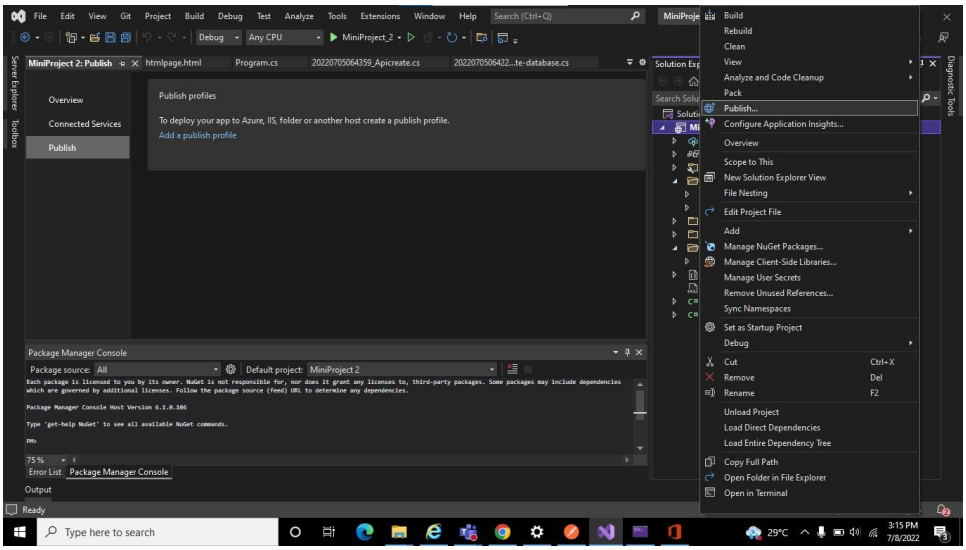
·Configure Application Insights for the project

·Configure Swagger for the Api

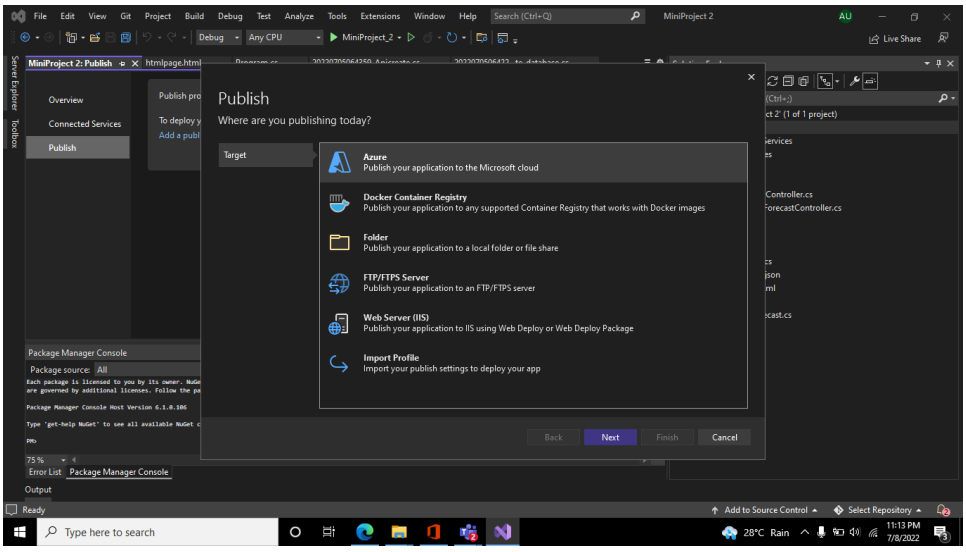
·Work with Log Analytics with the sample logs available

1.Host the Web API in azure and consume the same using jQuery Client.

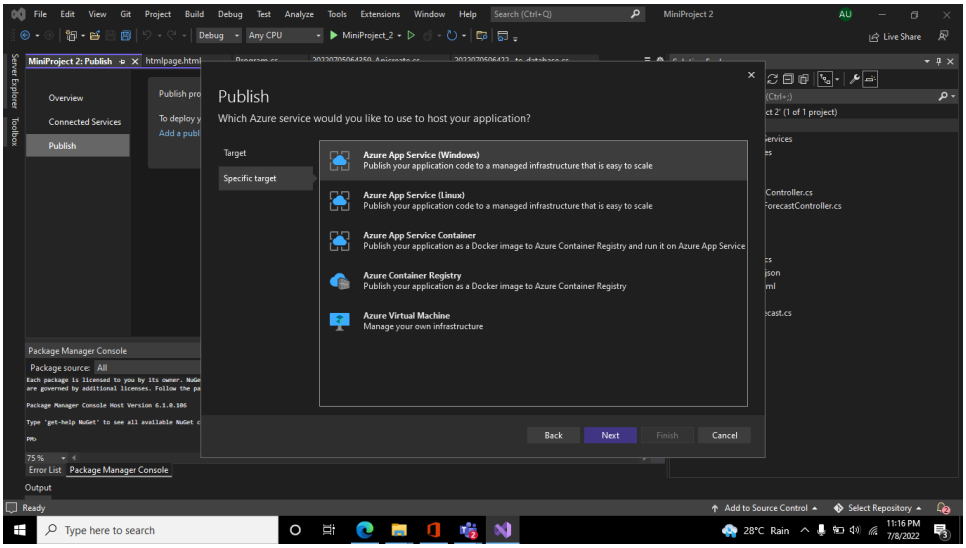
- In Solution **Explorer**, right-click the web site project and select **Publish**.



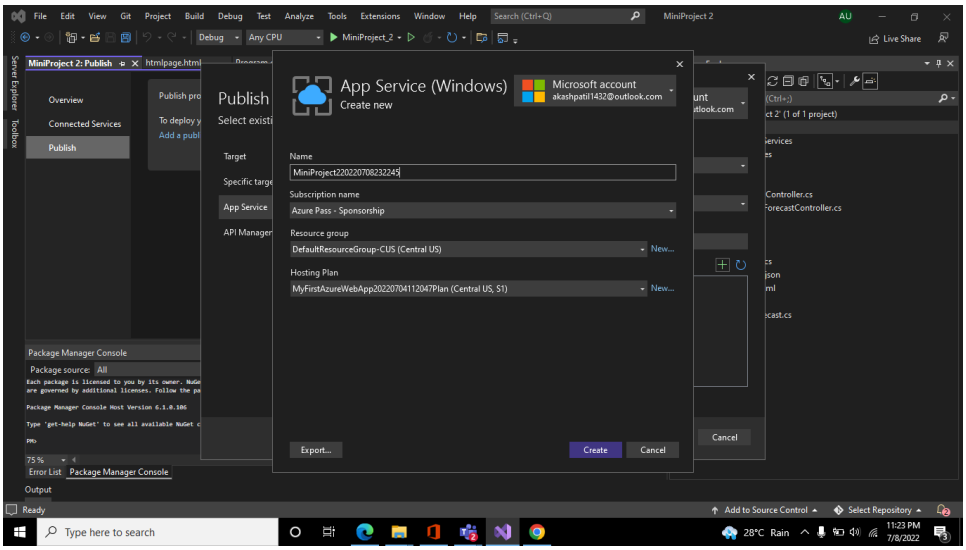
- ▪In the Publish dialog, select Azure and select the Next button



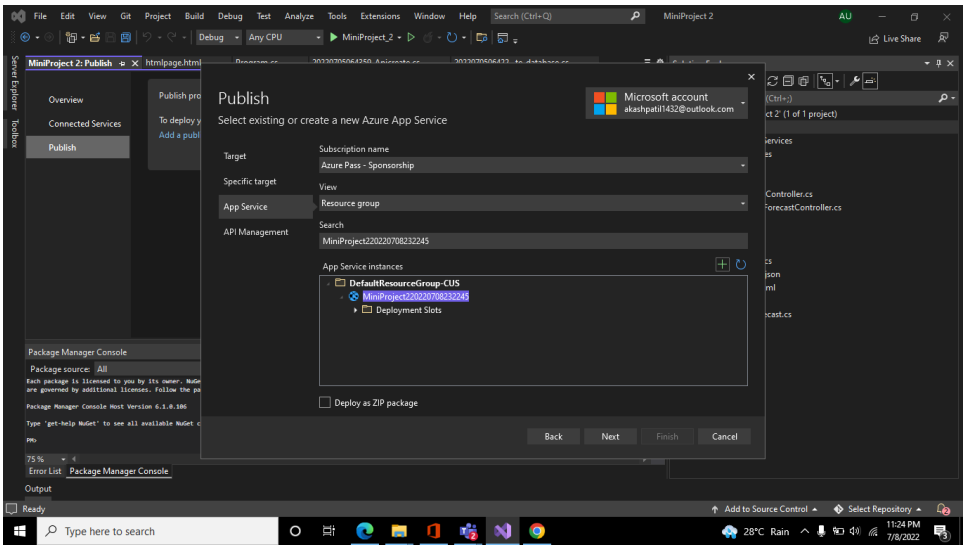
- Select Azure App Service (Windows) and select the Next button



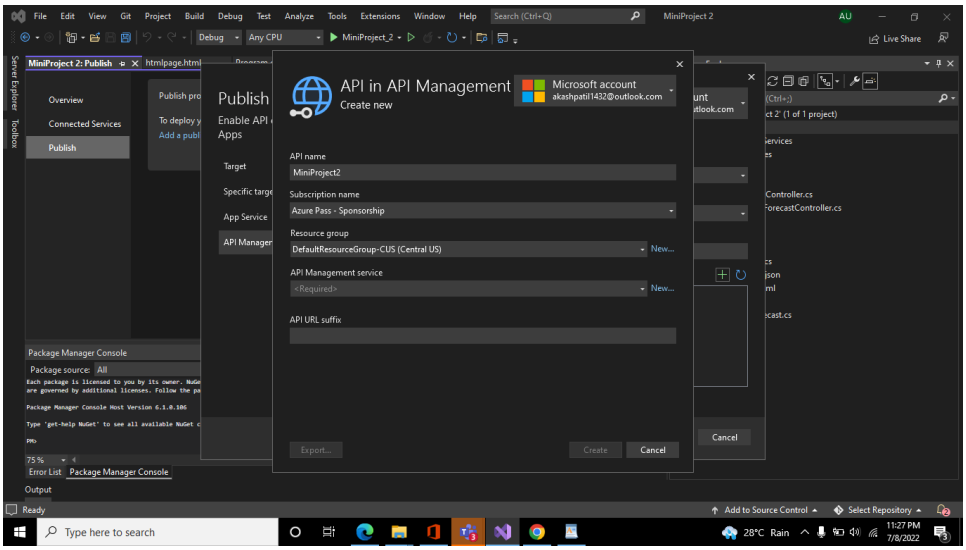
- Select Create a new Azure App Service.



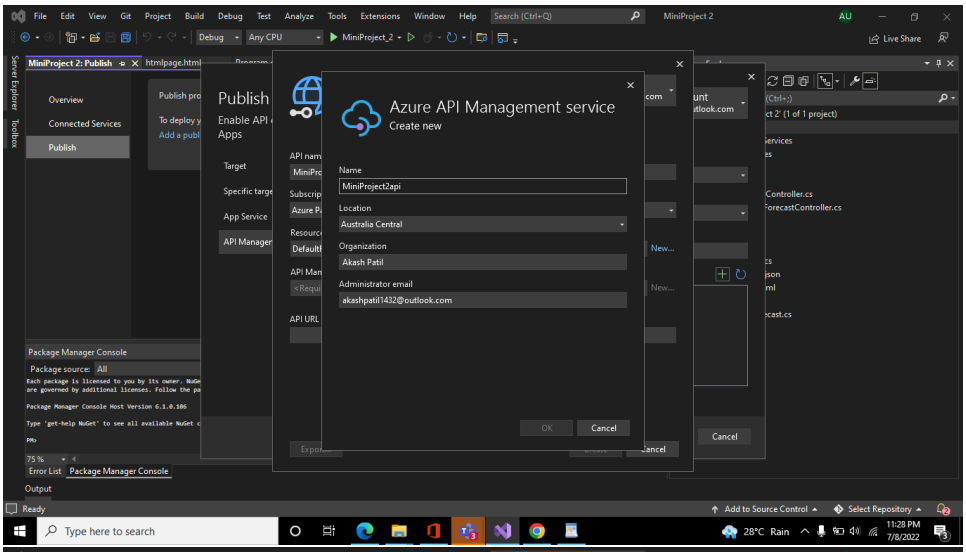
- The Create App Service dialog appears. The App Name, Resource Group, and App Service Plan entry fields are populated. You can keep these names or change them. Select the Create button.
- After creation is completed, the dialog is automatically closed and the Publish dialog gets focus again. The instance that was created is automatically selected.

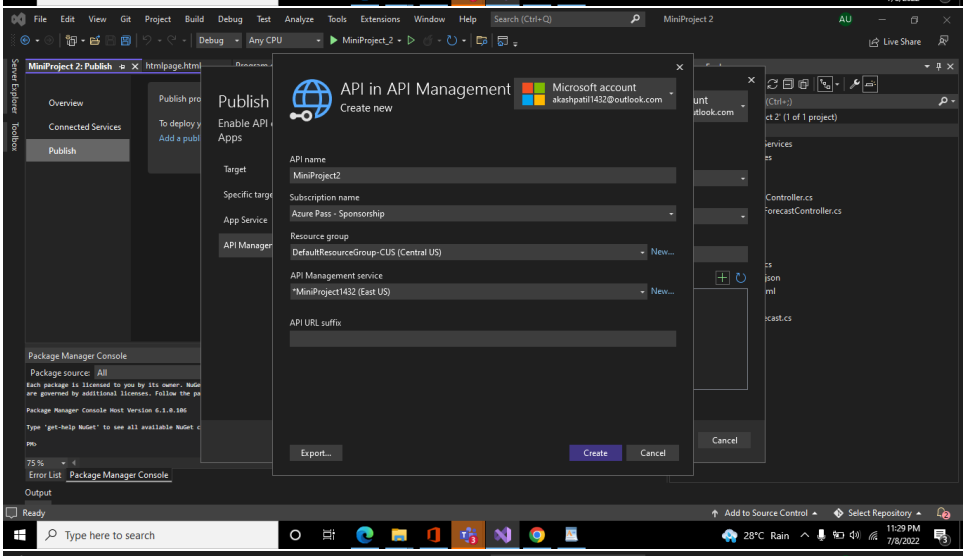


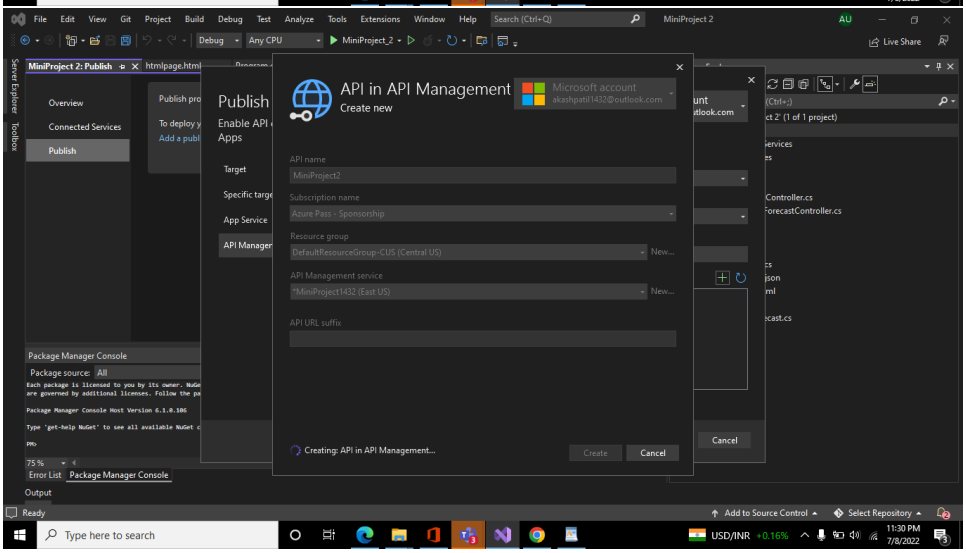
- The dialog now shows the Azure API Management service to create.



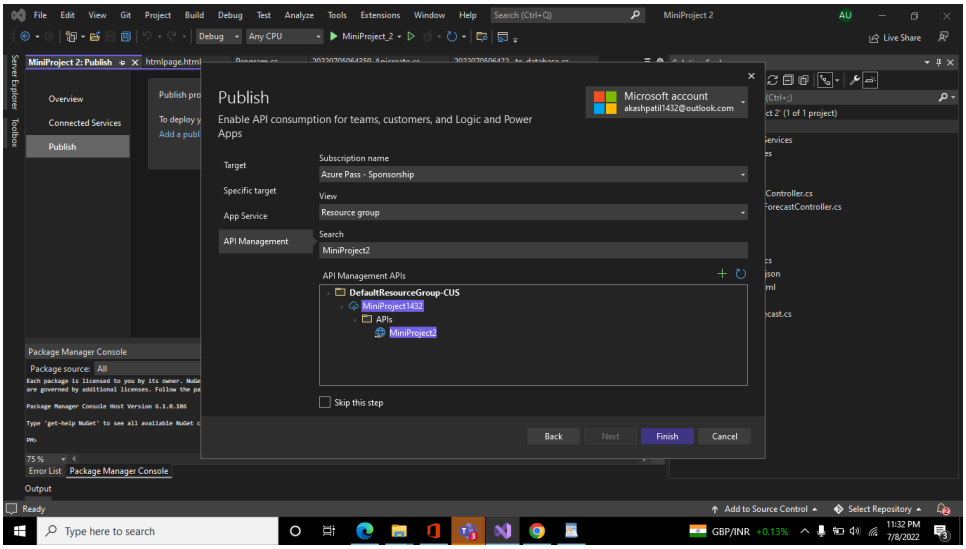
- The Create the Azure API Management service dialog appears. The App Name, Resource Group, and API Management service entry fields are populated. You can keep these names or change them. Select the Create button


Azure API Management service
Create new
Name
MiniProject2api
Location
Australia Central
Organization
Akash Patil
Administrator email
akashpatil1432@outlook.com
OK
Cancel

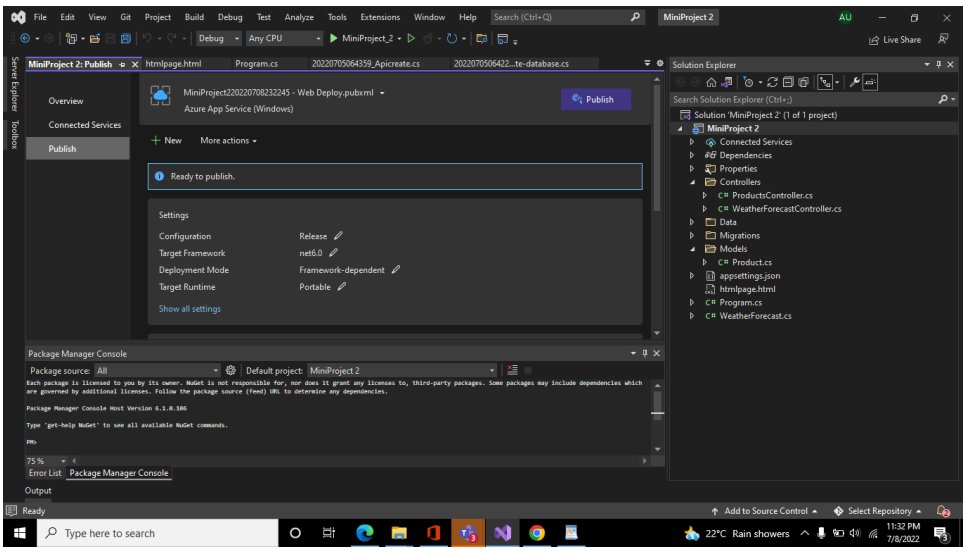

API in API Management
Create new
Microsoft account
akashpatil1432@outlook.com
API name
MiniProject2
Subscription name
Azure Pass - Sponsorship
Resource group
DefaultResourceGroup-CUS (Central US)
API Management service
*MiniProject1432 (East US)
API URL suffix
Export...
Create
Cancel


API in API Management
Create new
Microsoft account
akashpatil1432@outlook.com
Creating: API in API Management...
Create
Cancel

- After creation is completed, the dialog is automatically closed and the Publish dialog gets focus again. The instance that was created is automatically selected.



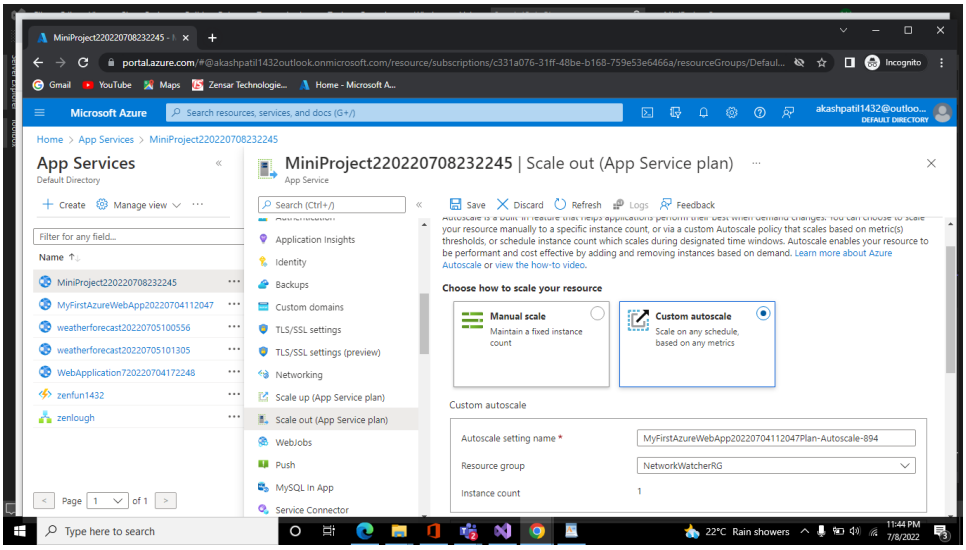
- The dialog closes and a summary screen appears with information about the publish. Select the Publish button



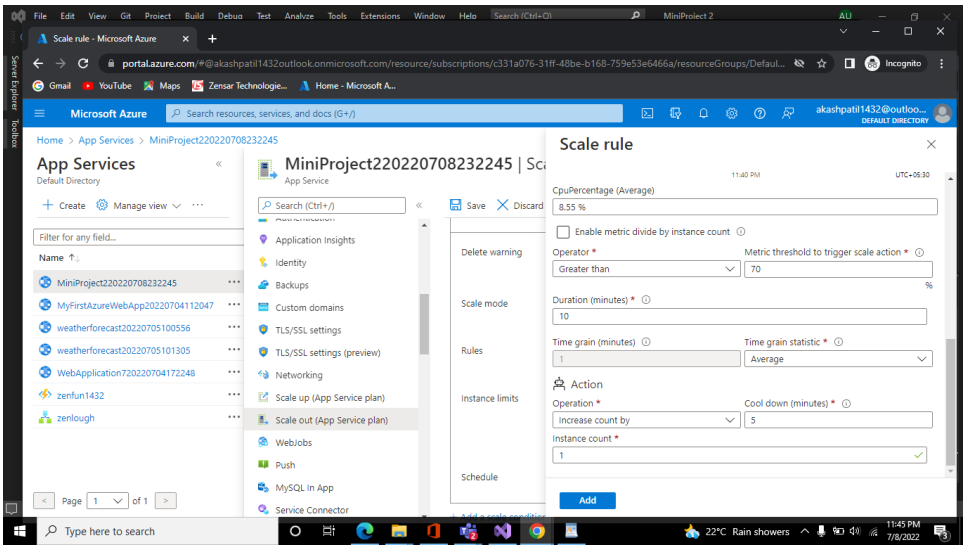
☺ Once the publishing process finishes, your default browser will open the published web site.

2. Configure Scale out by adding rules for custom scaling

- Switch back to the Azure API Management instance in the Azure portal. Refresh the browser window. Select the API you created in the preceding steps. It's now populated and you can explore around.
- Search and select Auto scale in the search bar and Select Custom Auto scale In the Rules section of the default scale condition, select Add a rule. From the Metric source dropdown, select current resource. From Resource Type, select Application Insights.

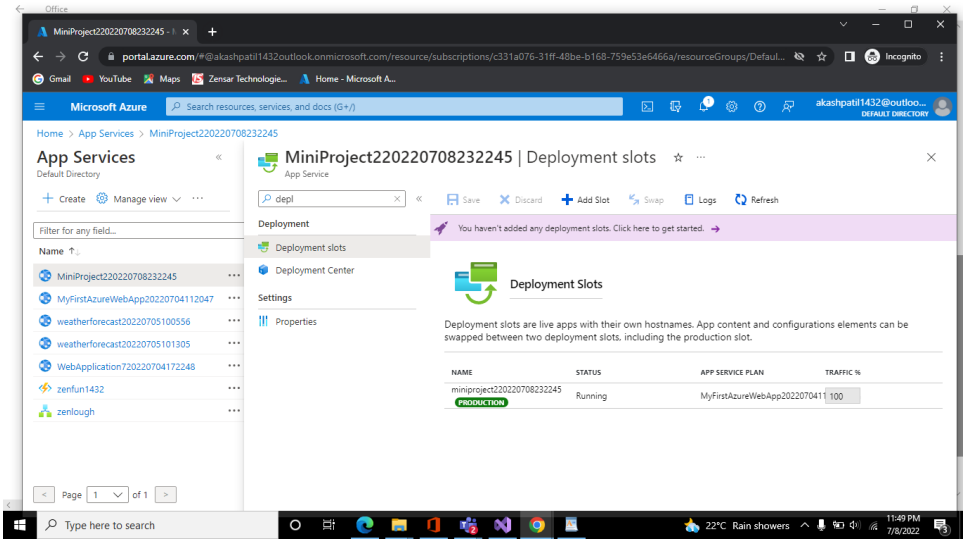


- From the Resource dropdown, select your App services plan standard metrics. Select a Metric name to CPU Percentage and Select Enable metric divide by instance count so that the number of sessions per instance is measured.From the Operator dropdown, select Greater than.
- Enter the Metric threshold to trigger the scale action, for example, 70 and Under Actions, set the Operation to Increase count and set the Instance count to 1 and Cool down by 5minutes and then click Add. Set the maximum number of instances that can be spun up in the Maximum field of the Instance limits section and Select Add

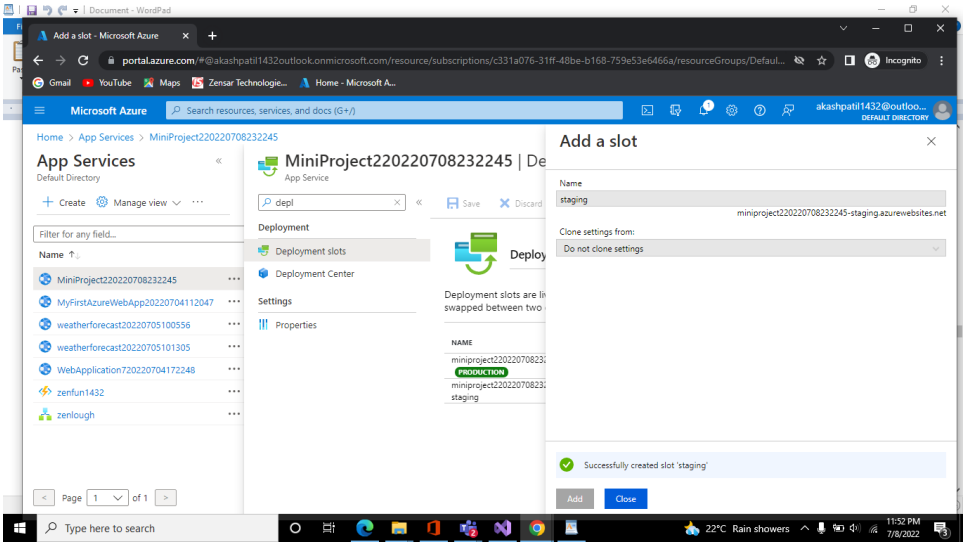


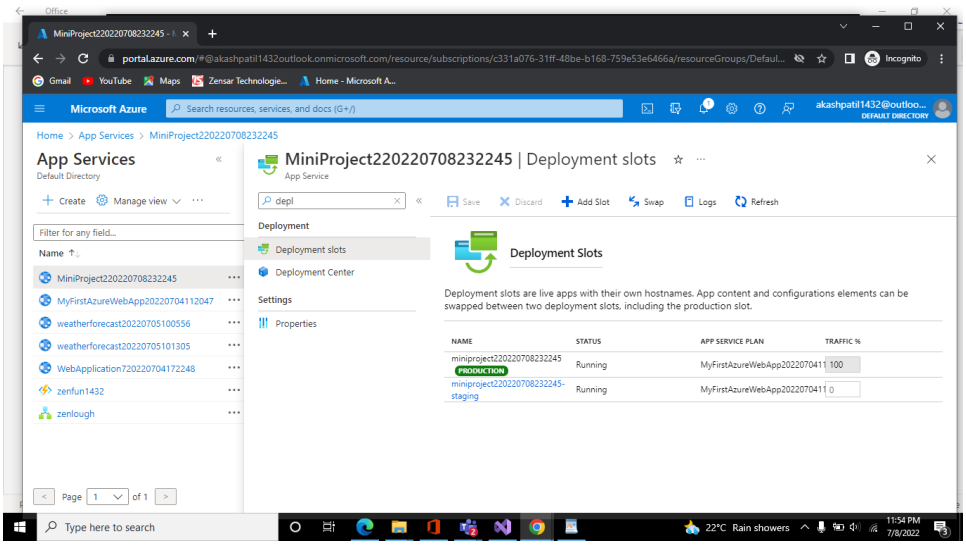
☺ These two images show an App Service plan of Scale rule

3.Configure Deployment slots for staging and production

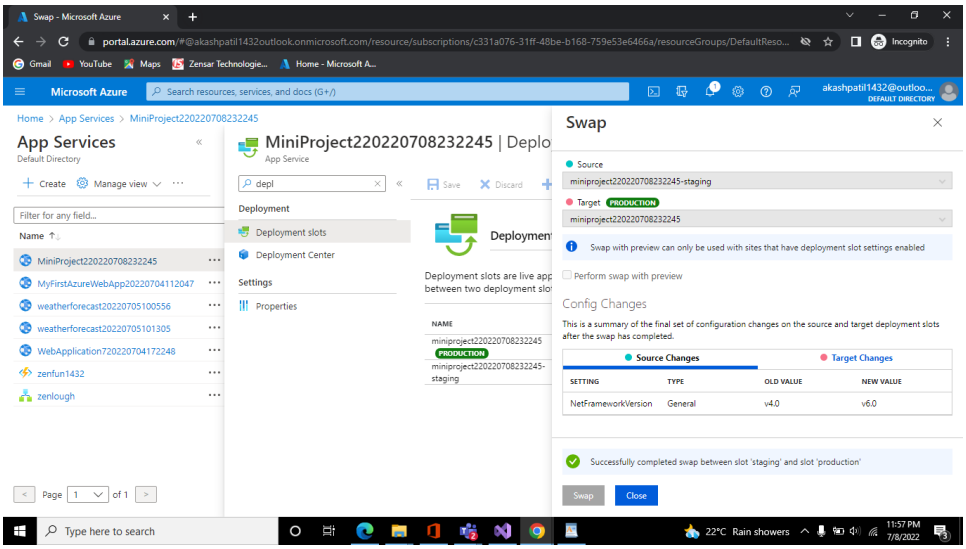


- ⍰Add the Slot and Name as Staging and then Add the Slot

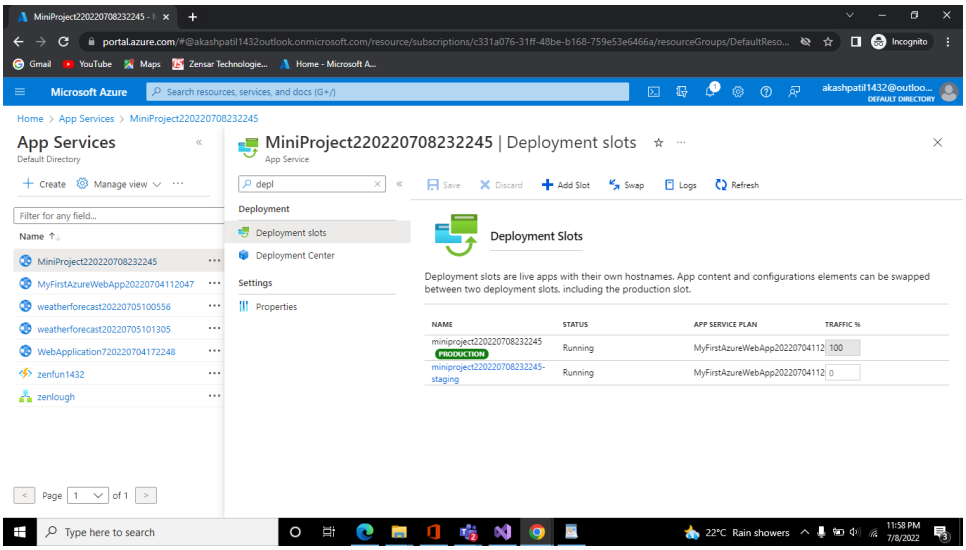




- Select Deployment slots, and then select Swap and Verify the configuration settings for your swap and select Swap

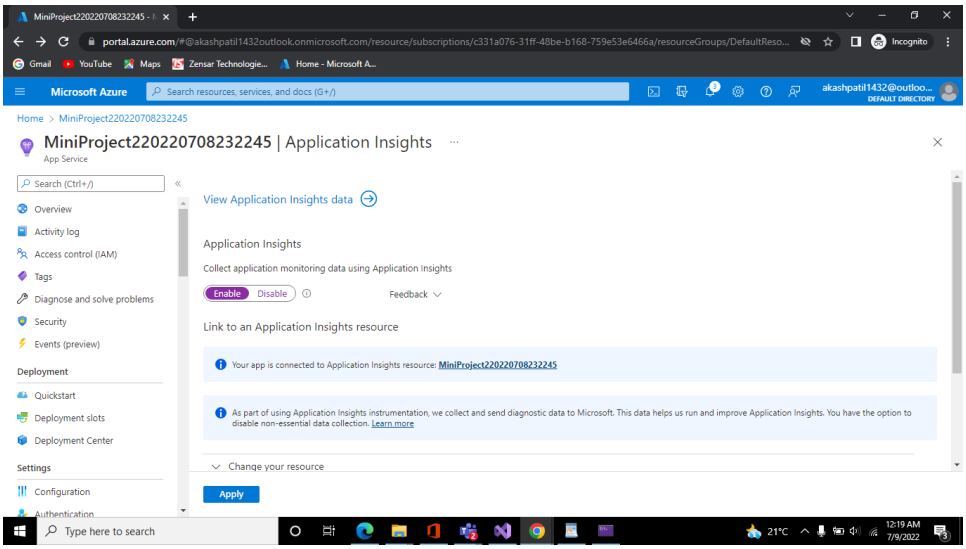


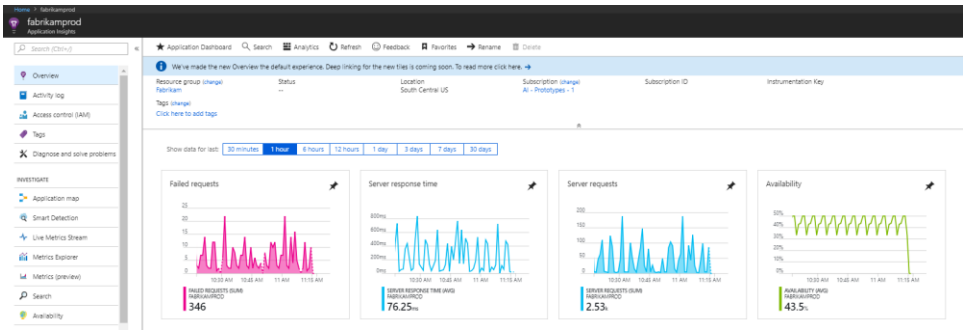
- After Add Slot Successful both the Production and Staging is Displayed

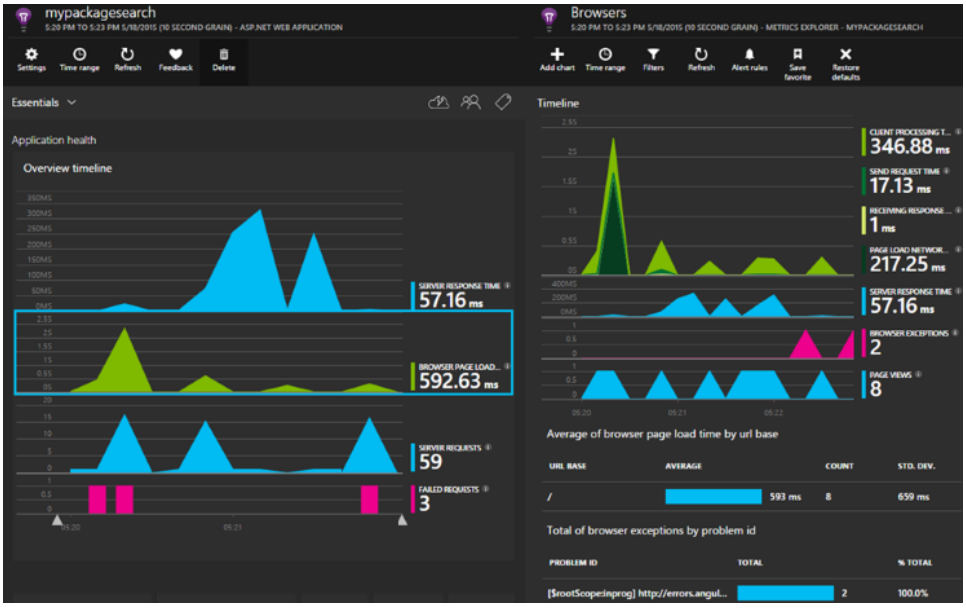


4. Configure Application Insights for the project

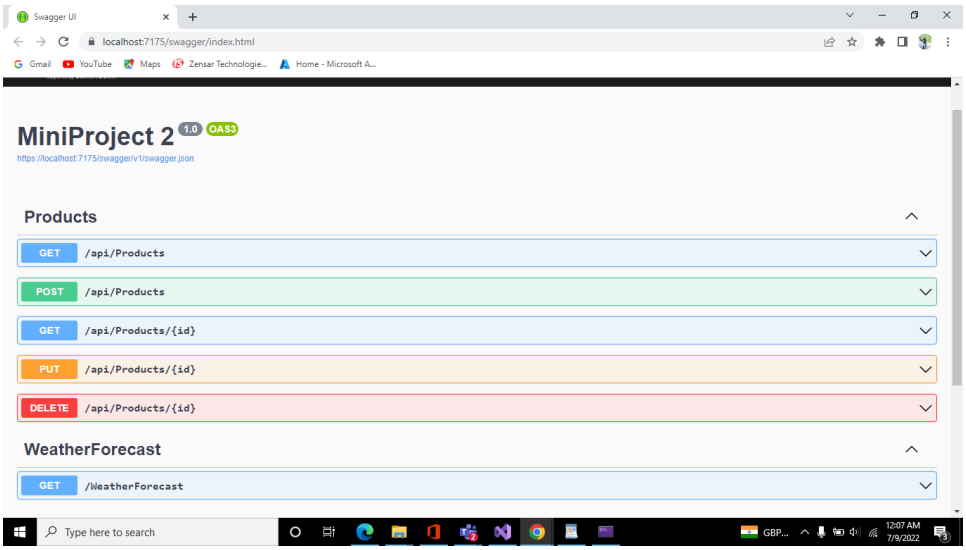
- Select the Application Insights

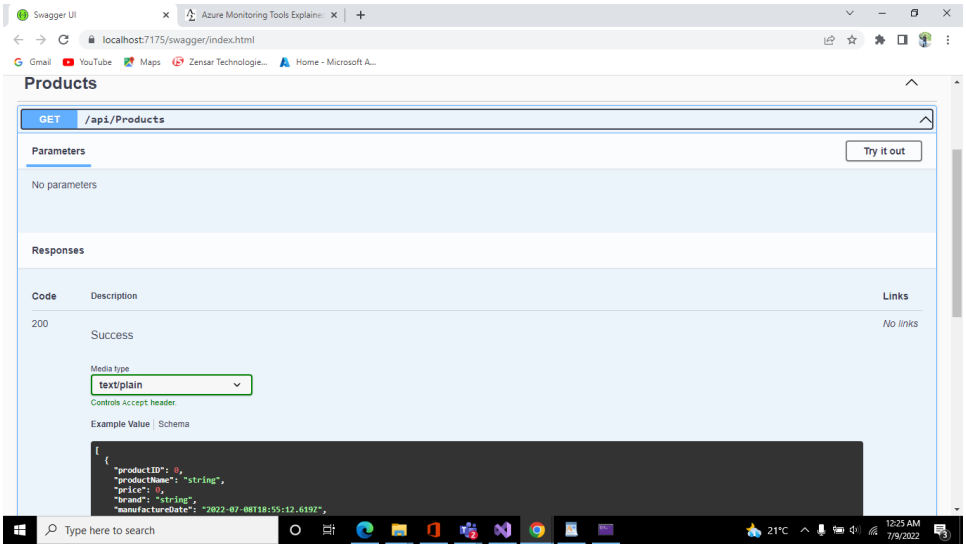

MiniProject220220708232245
Microsoft Azure
Home > MiniProject220220708232245
MiniProject220220708232245 | Application Insights
App Service
View Application Insights data
Application Insights
Collect application monitoring data using Application Insights
Enable
Disable
Feedback
Link to an Application Insights resource
Your app is connected to Application Insights resource: MiniProject220220708232245
Change your resource
Apply
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Security
Events (preview)
Deployment
Quickstart
Deployment slots
Deployment Center
Settings
Configuration
Type here to search

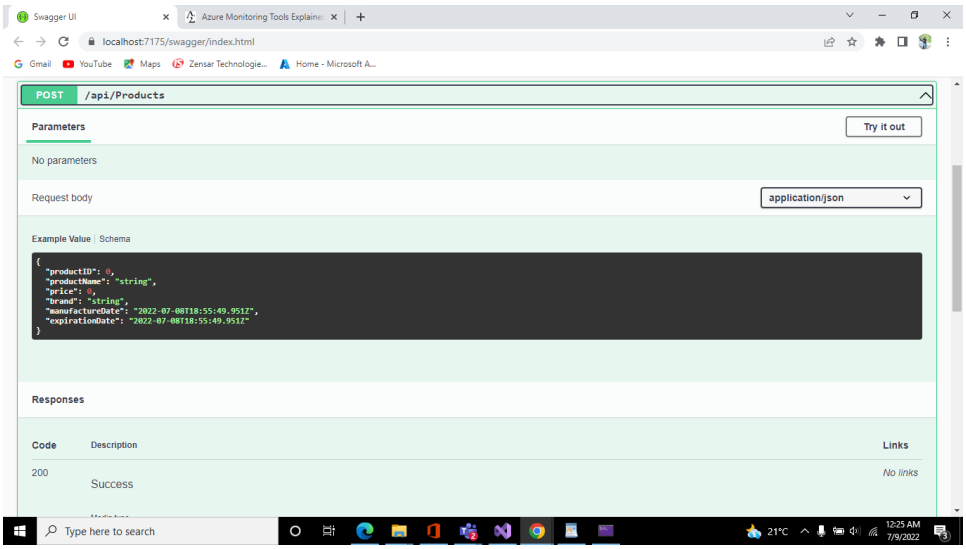

fabrikamprod
Application Insights
Failed requests
Server response time
Server requests
Availability
346
76.25ms
2.53k
43.5%

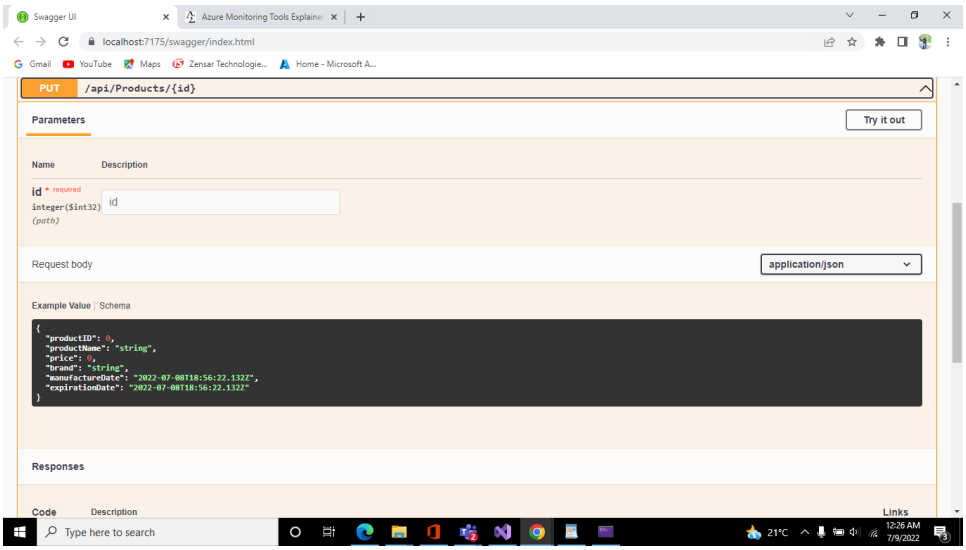

mypackagesearch
Settings
Time range
Refresh
Feedback
Delete
Essentials
Application health
Overview timeline
SERVER RESPONSE TIME
57.16 ms
BROWSER PAGE LOAD...
592.63 ms
SERVER REQUESTS
59
FAILED REQUESTS
3
Browsers
Add chart
Time range
Filters
Refresh
Alert rules
Save favorite
Restore defaults
Timeline
CLIENT PROCESSING T...
346.88 ms
SEND REQUEST TIME
17.13 ms
RECEIVING RESPONSE...
1 ms
PAGE LOAD NETWOR...
217.25 ms
SERVER RESPONSE TIME
57.16 ms
BROWSER EXCEPTIONS
2
PAGE VIEWS
8
Average of browser page load time by url base
URL BASE
AVERAGE
COUNT
STD. DEV.
/
593 ms
8
659 ms
Total of browser exceptions by problem id
PROBLEM ID
TOTAL
% TOTAL
[$rootScope.inprog] http://errors.angul...
2
100.0%

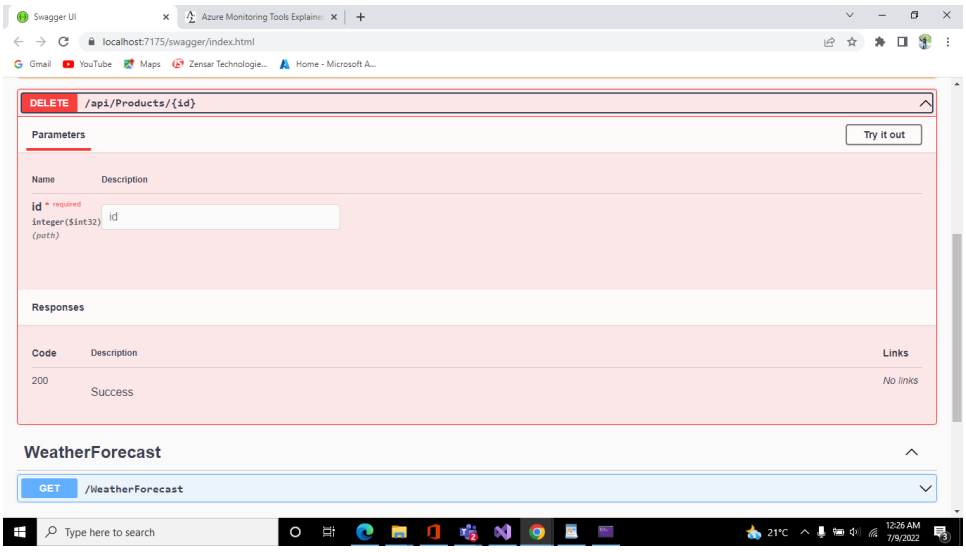
5. Configure Swagger for the API





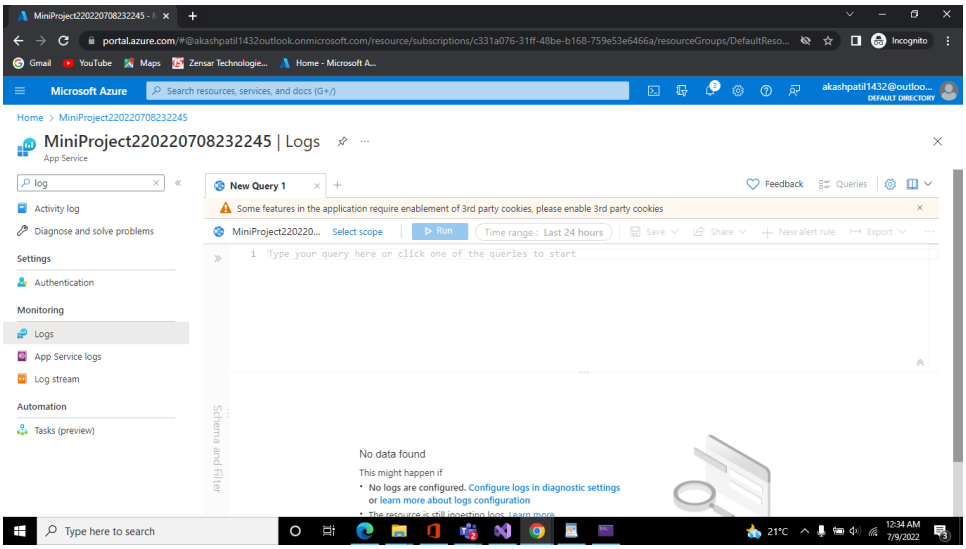

Swagger UI
Azure Monitoring Tools Explained
localhost:7175/swagger/index.html
POST /api/Products
Parameters
Try it out
No parameters
Request body
application/json
Example Value | Schema
{
"productID": 0,
"productName": "string",
"price": 0,
"brand": "string",
"manufactureDate": "2022-07-08T18:55:49.951Z",
"expirationDate": "2022-07-08T18:55:49.951Z"
}
Responses
Code
Description
Links
200
Success
No links
Type here to search
21°C
12:25 AM
7/9/2022


Swagger UI
Azure Monitoring Tools Explained
localhost:7175/swagger/index.html
PUT /api/Products/{id}
Parameters
Try it out
Name
Description
id * required
integer($int32)
(path)
id
Request body
application/json
Example Value | Schema
{
"productID": 0,
"productName": "string",
"price": 0,
"brand": "string",
"manufactureDate": "2022-07-08T18:56:22.132Z",
"expirationDate": "2022-07-08T18:56:22.132Z"
}
Responses
Code
Description
Links
Type here to search
21°C
12:26 AM
7/9/2022

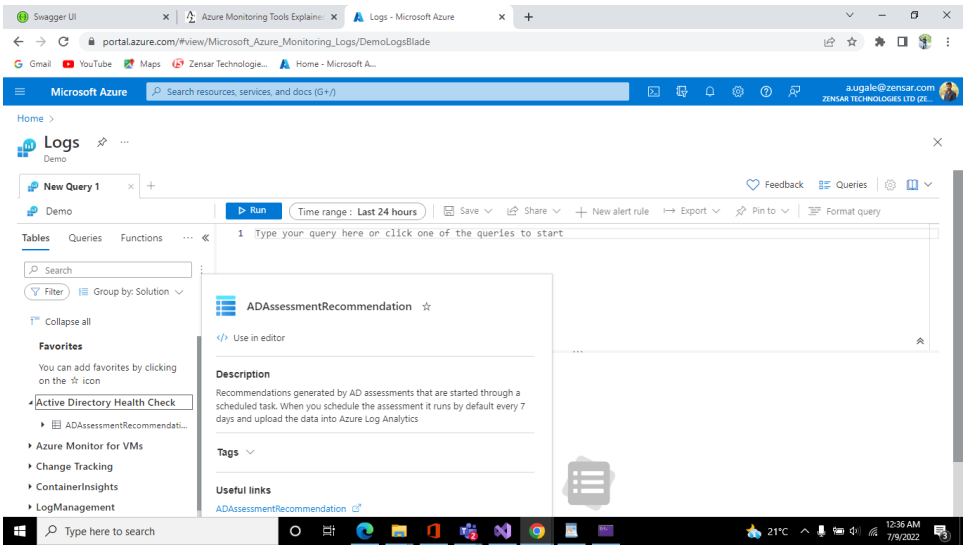

Swagger UI
Azure Monitoring Tools Explained
localhost:7175/swagger/index.html
DELETE /api/Products/{id}
Parameters
Try it out
Name
Description
id * required
integer($int32)
(path)
id
Responses
Code
Description
Links
200
Success
No links
WeatherForecast
GET /WeatherForecast
Type here to search
21°C
12:26 AM
7/9/2022

6.Work with Log Analytics with the sample logs available

- Select the Logs in Azure Portal.



- Select Logs from the Azure Monitor menu . This step sets the initial scope to a Log Analytics workspace so that your query selects from all data in that workspace



?

- This is the simplest query that we can write. It just returns all the records to a table. Run it by selecting the Run button or by selecting Shift Enter with the cursor positioned anywhere in the query text and Select Run to return the results