ASP.NET CORE – AZURE MINI PROJECT

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:

- ProductID
- ProductName
- Price
- Brand
- ManufactureDate
- ExpirationDate

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.

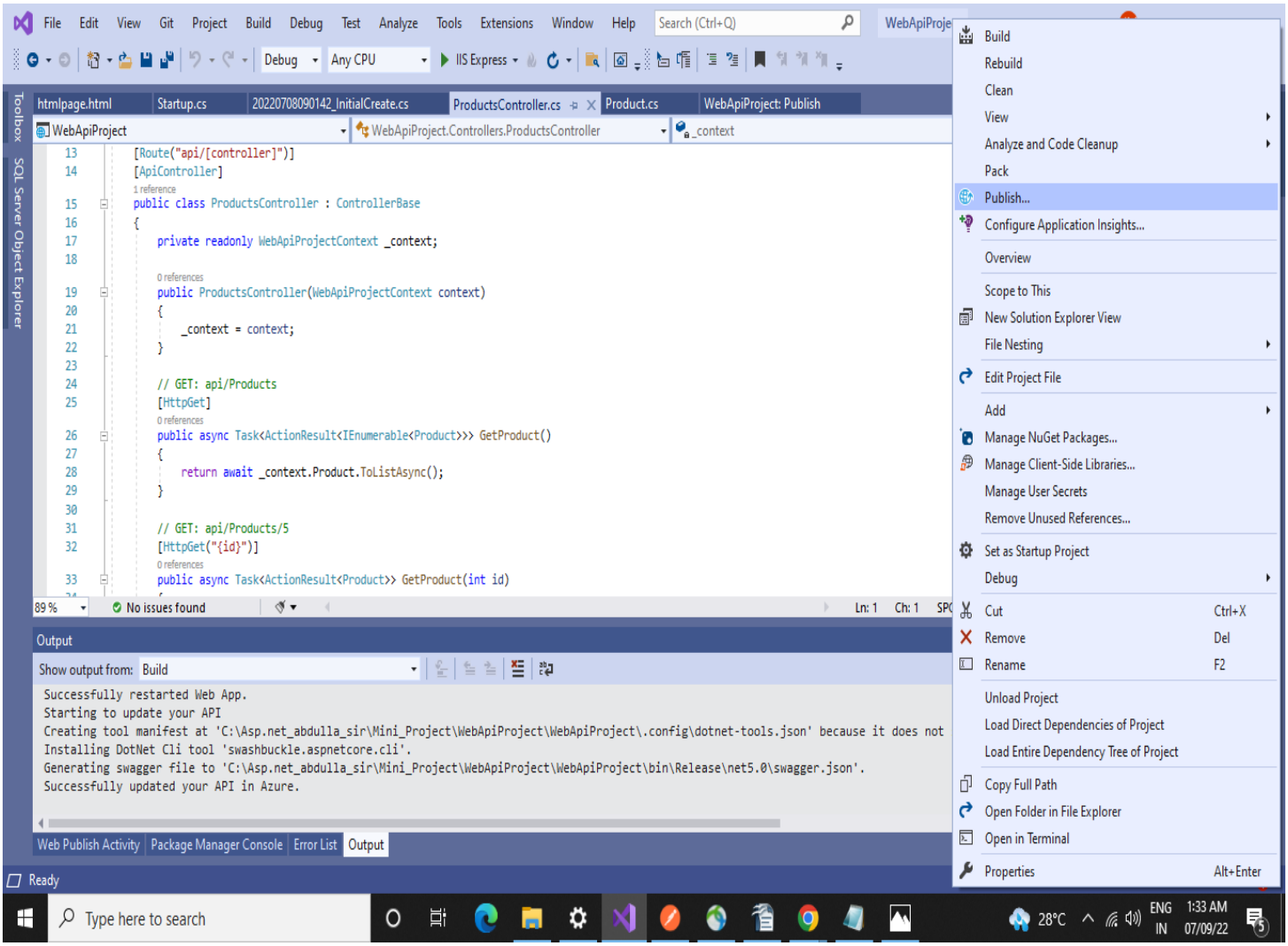
Azure Hosting

- 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

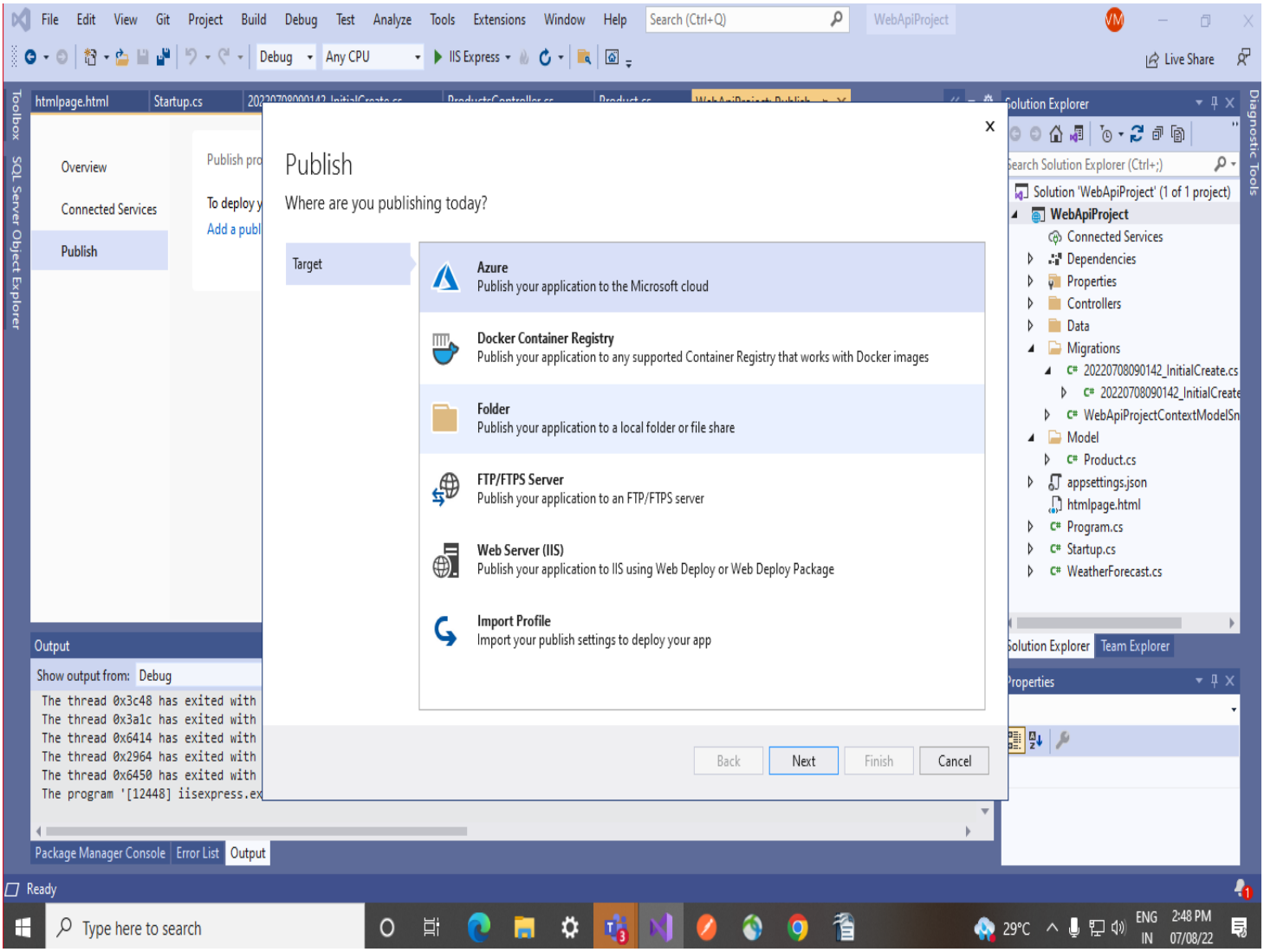
1. Host the web api in azure and consume the same using jQuery Client.

***** In Solution Explorer, right-click on the project and click on **Publish**

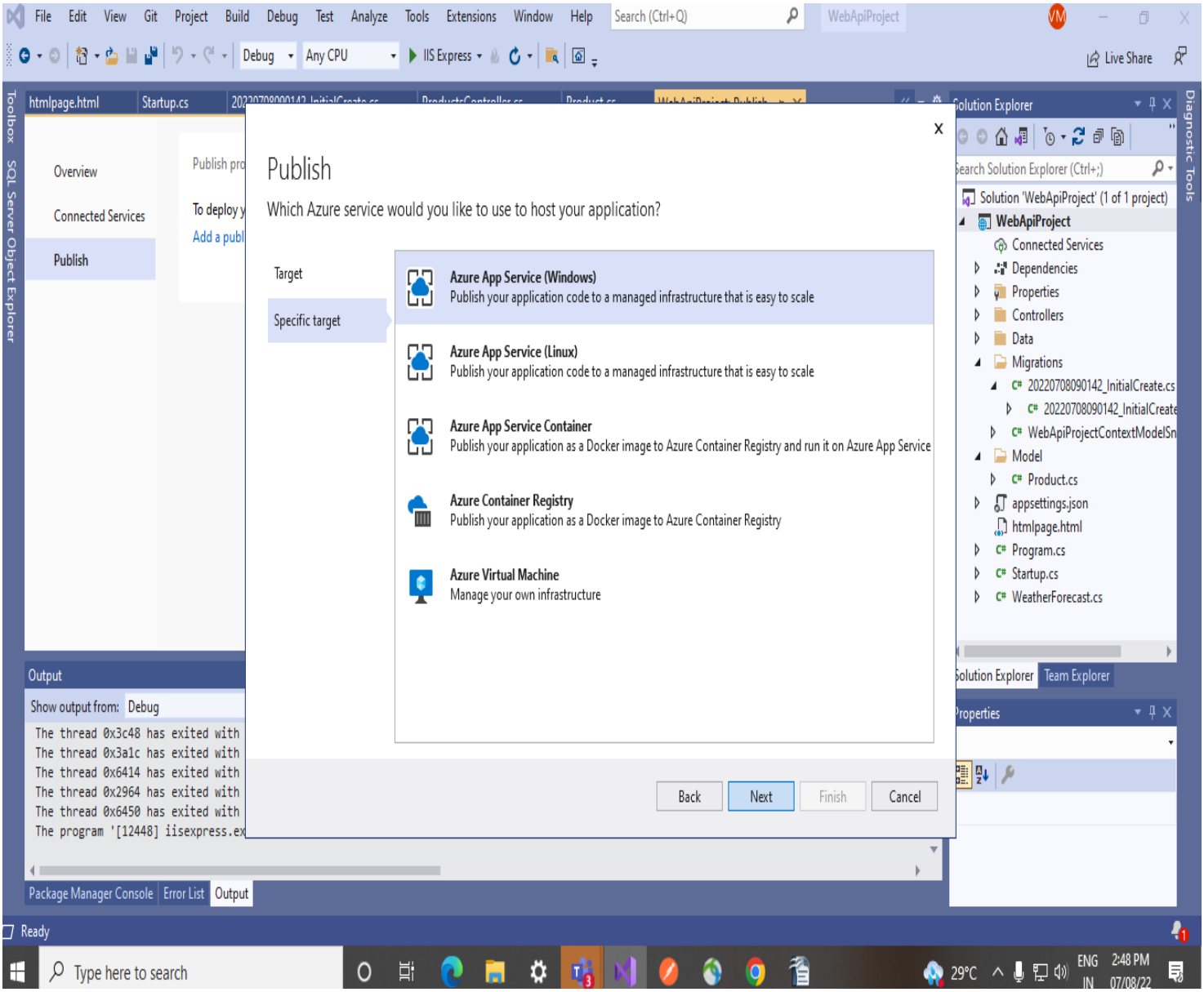
(**publish profile** is **a file that contains information and settings that Visual Studio uses to deploy applications and services to Azure.**)



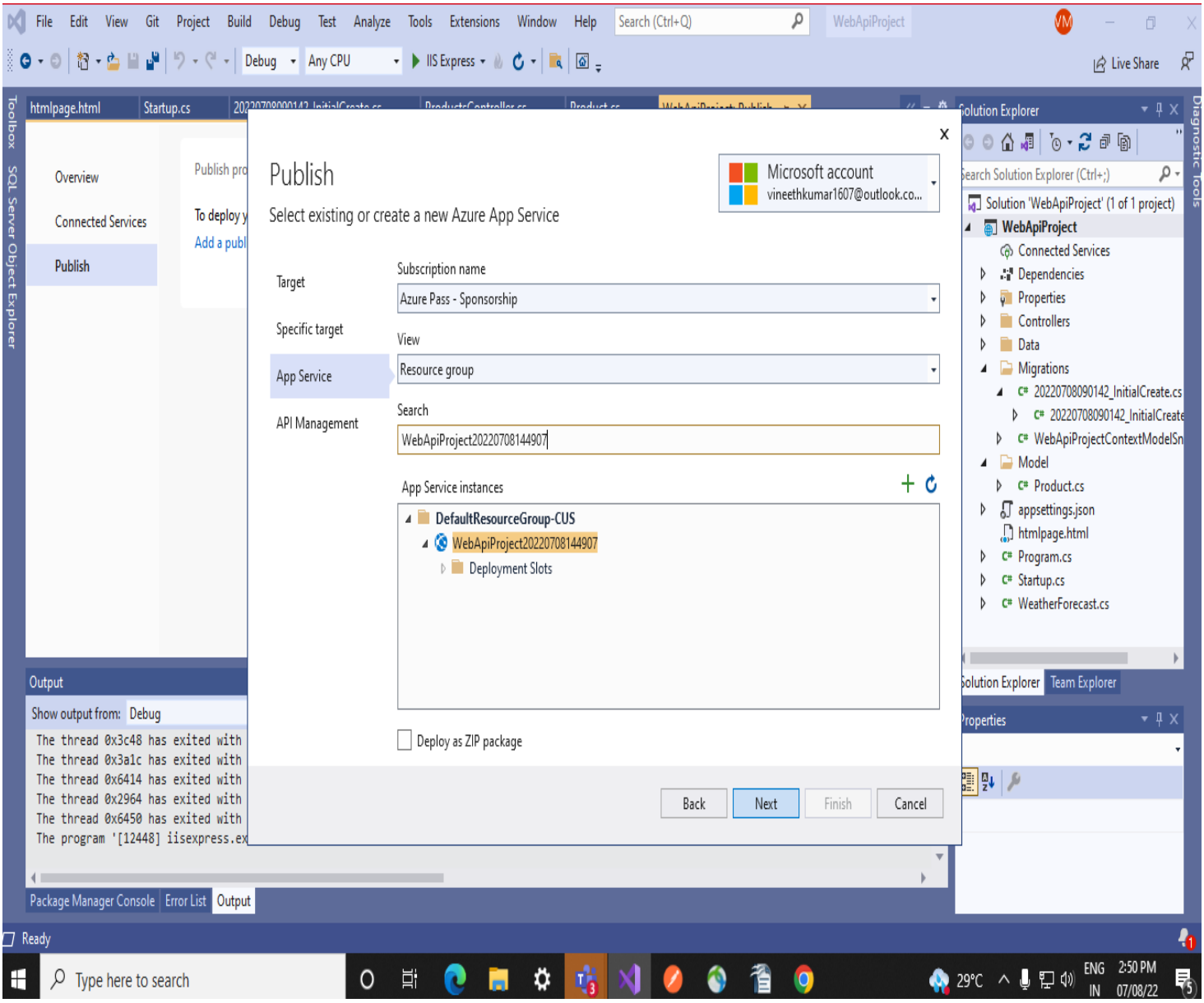
* In publish dialog box, select azure and click on next.



* In specific target dialog box select **Azure App Service (Windows)** and click on next.



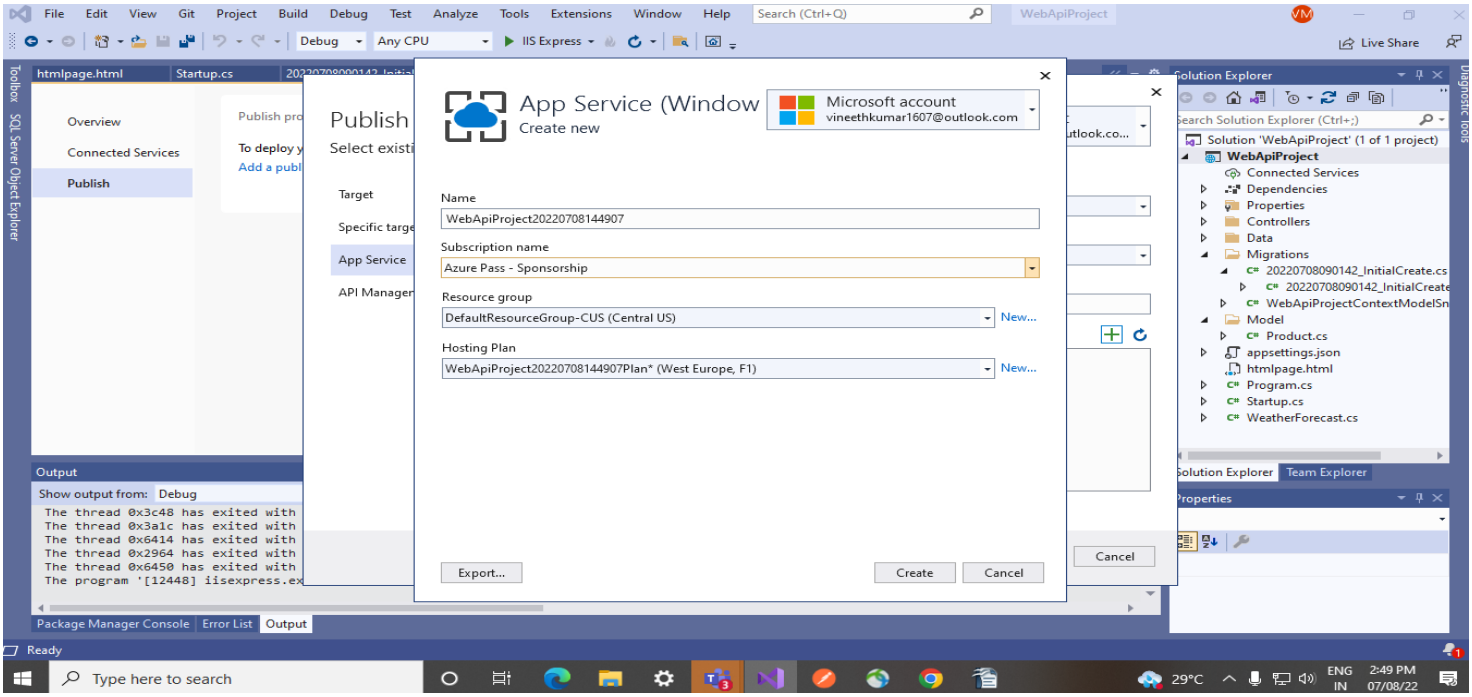
* Sign in into Azure Account and click on create a new Azure App Service (+).



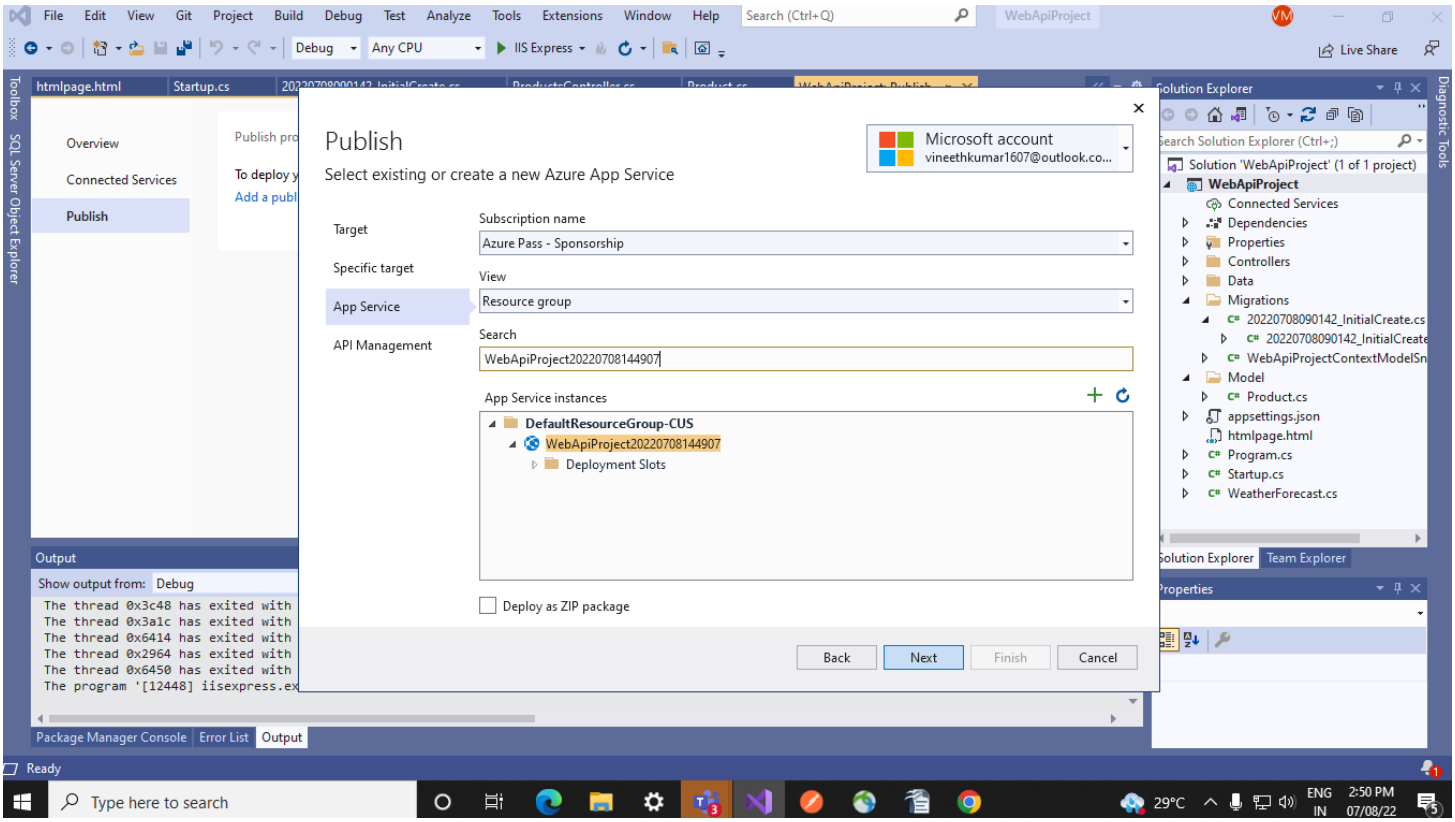
* Create App Service dialog box will appears.

Name, Subcription name, Resource group, and Hosting plan fields are selected by default. If we want, we can use those default names or else we can change them according to our convenience by clicking on new.

Click on the Create.



* Now the App Service which we have created gets selected in the publish dialog box.



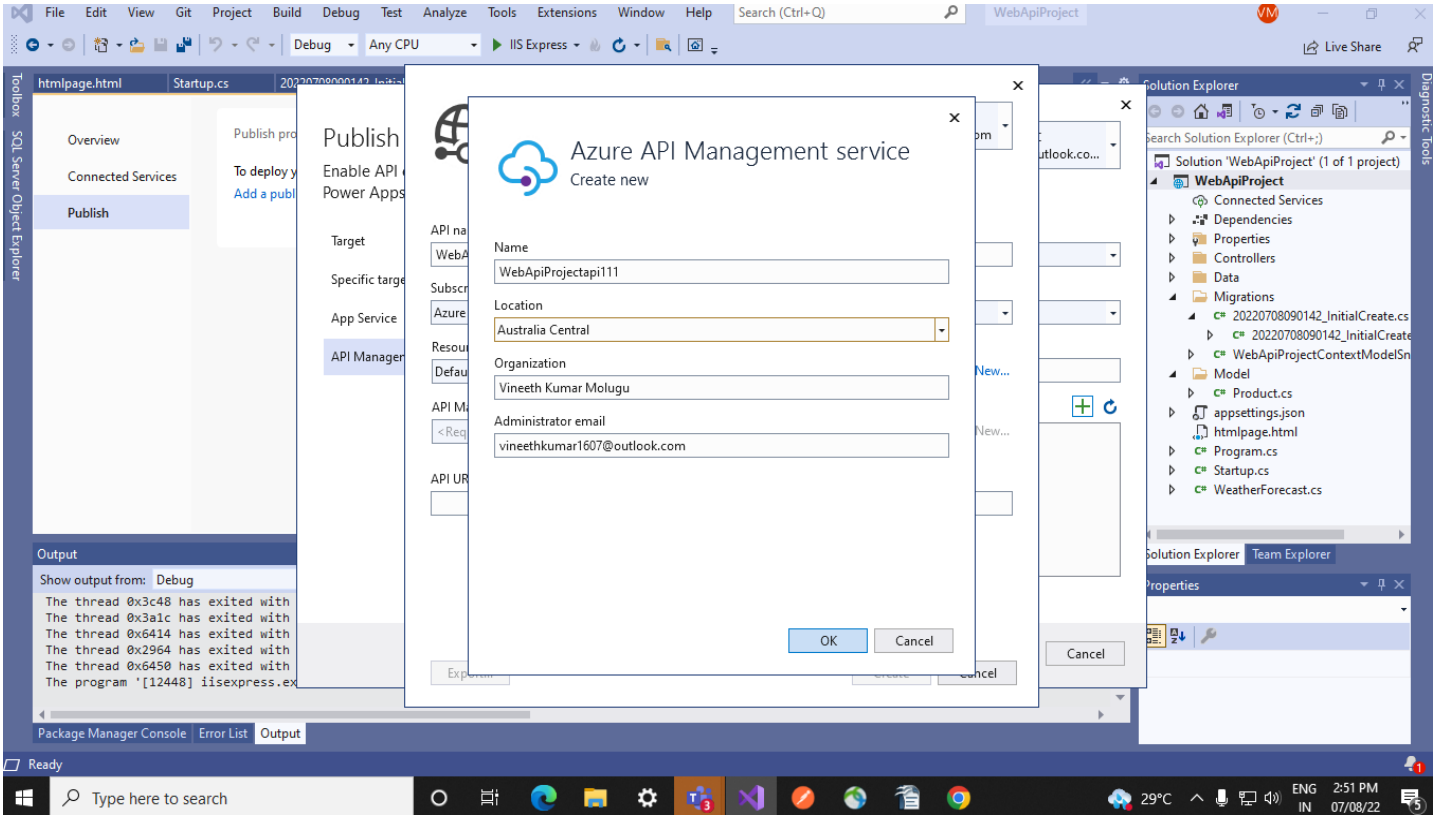
* Now new dialog box shows to create Azure API Management Service.

By selecting the create an API Management(+) we will see the following dialog box as appears below.

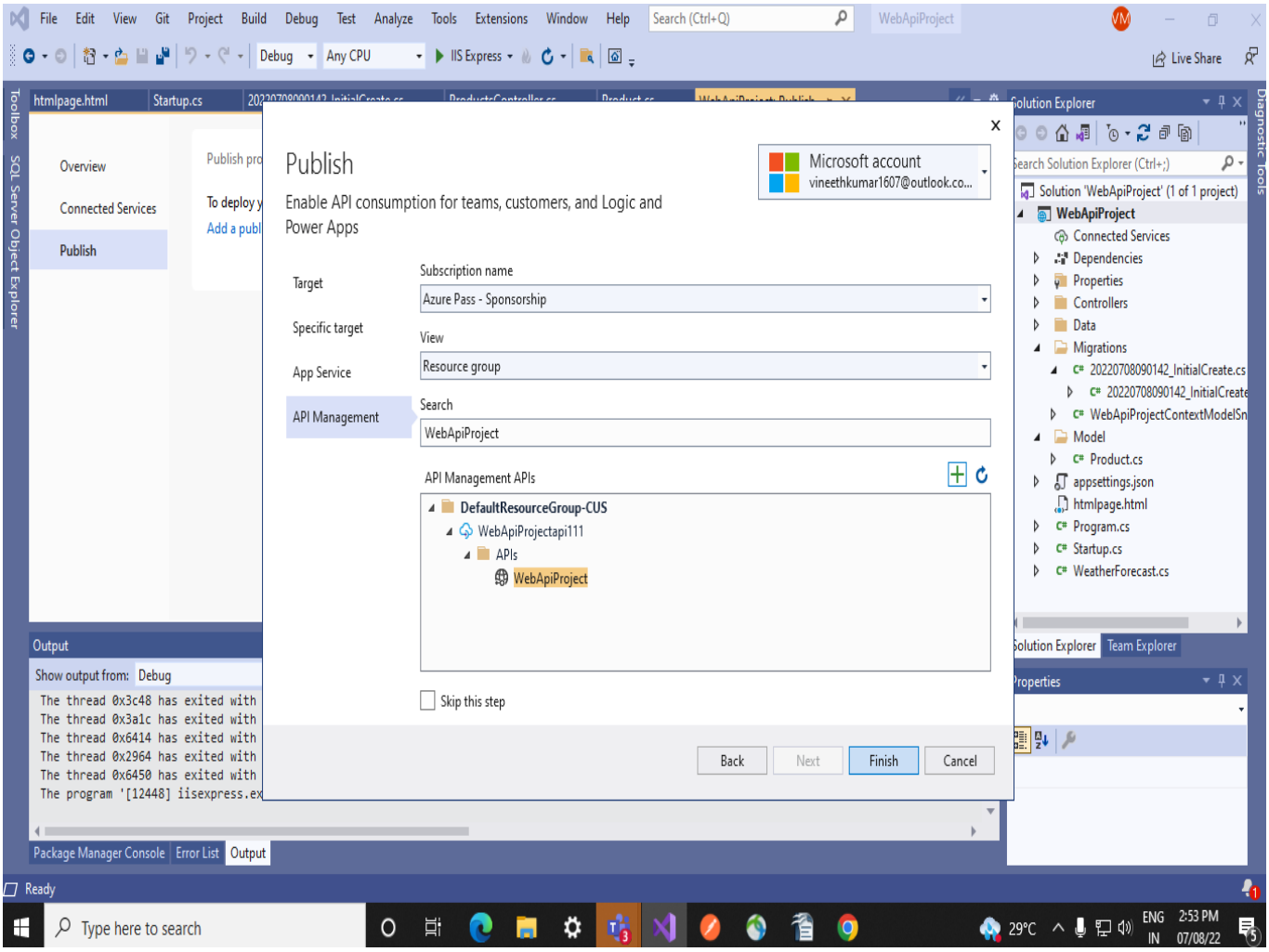
* Create new API Management dialog box will appears.

Name, Location, Organization, and Administrator email address fields are selected automatically. If we want, we can use those default values or else we can change them according to our convenience.

Now click on ok.



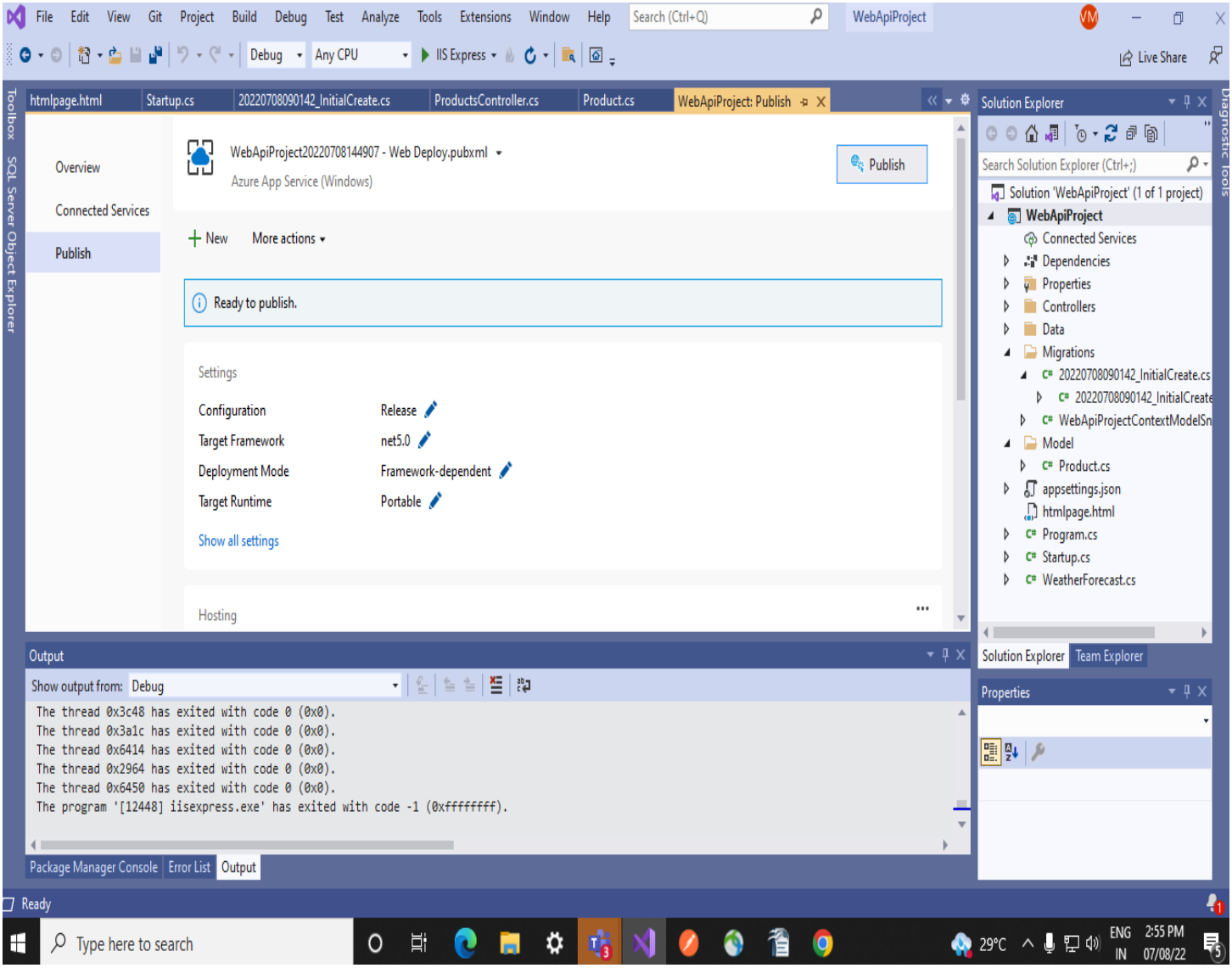
* Now the API Management has been created. In the publish dialog box, we can see that API management is added.



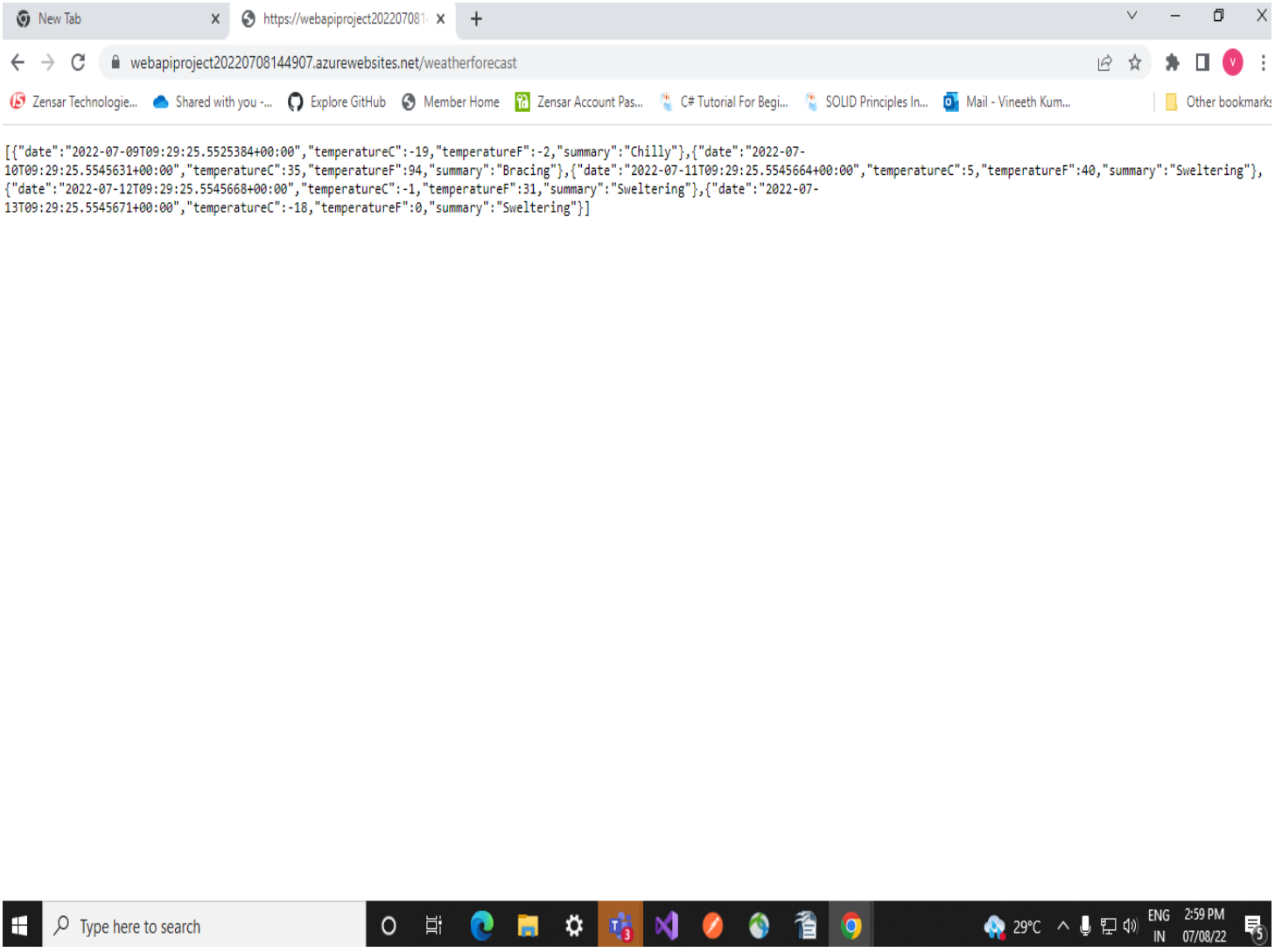
* Click on finish

* The publish dialog box gets closed and in visual studio we can see the detailed information about the project that we are going to publish.

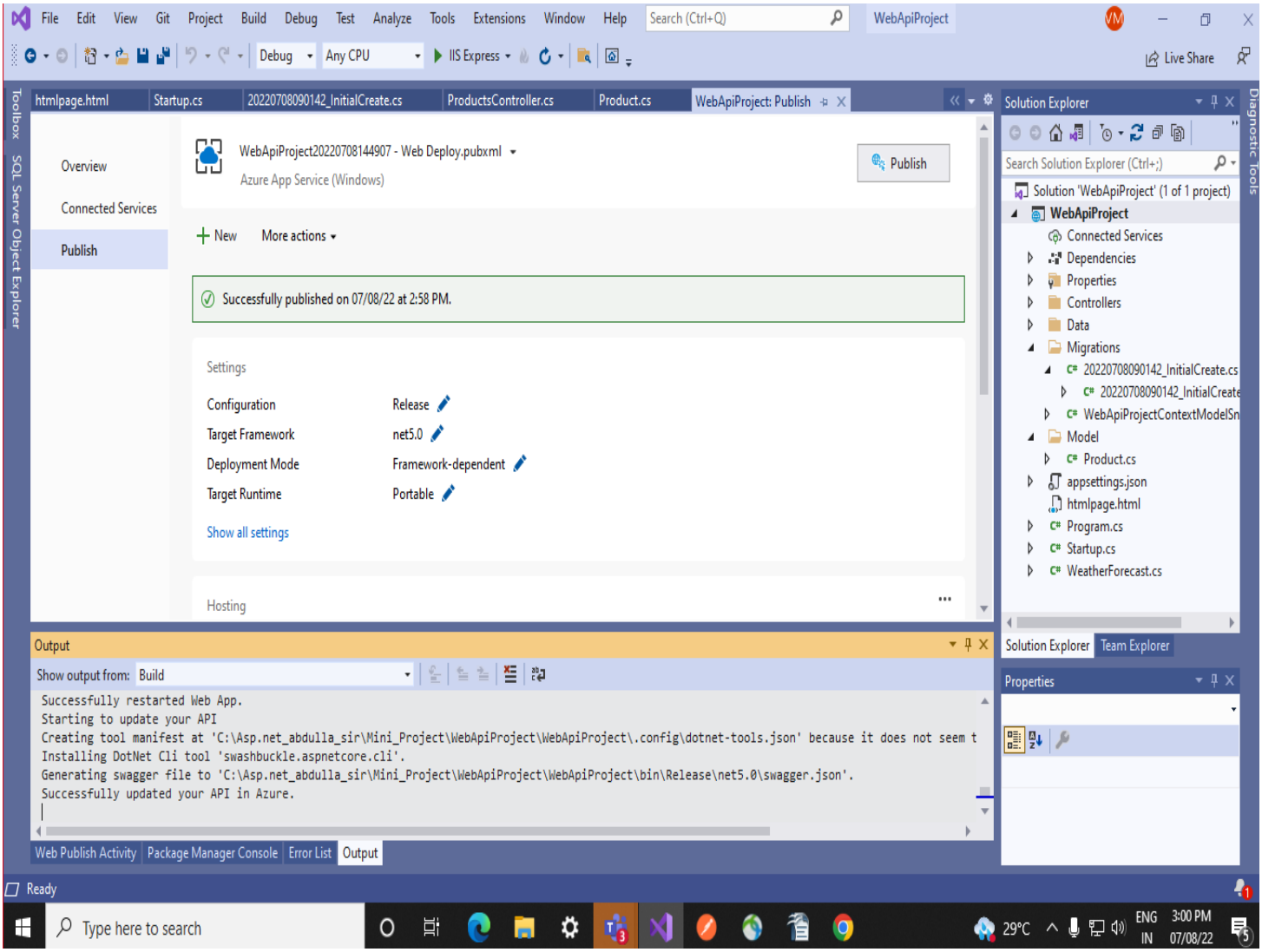
Click on publish



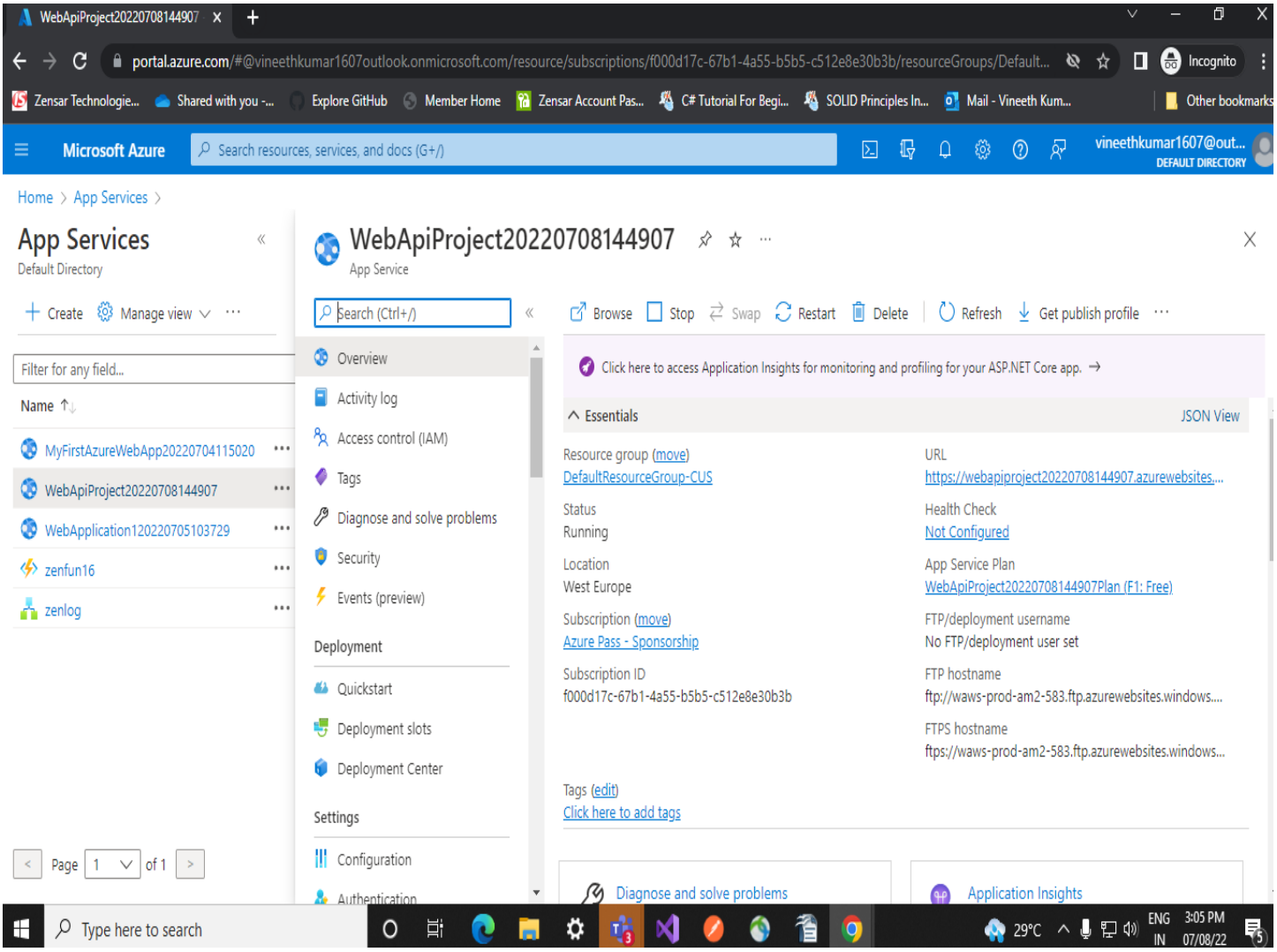
* The web API will publish to both Azure App Service and Azure API Management. A new browser window will appear and show the API running in Azure App Service.



*In visual studio, as we can see it Published successfully.



* Open azure portal. In Azure portal open app services. Select the API we have created in the preceding steps. It's now populated and we can explore around.



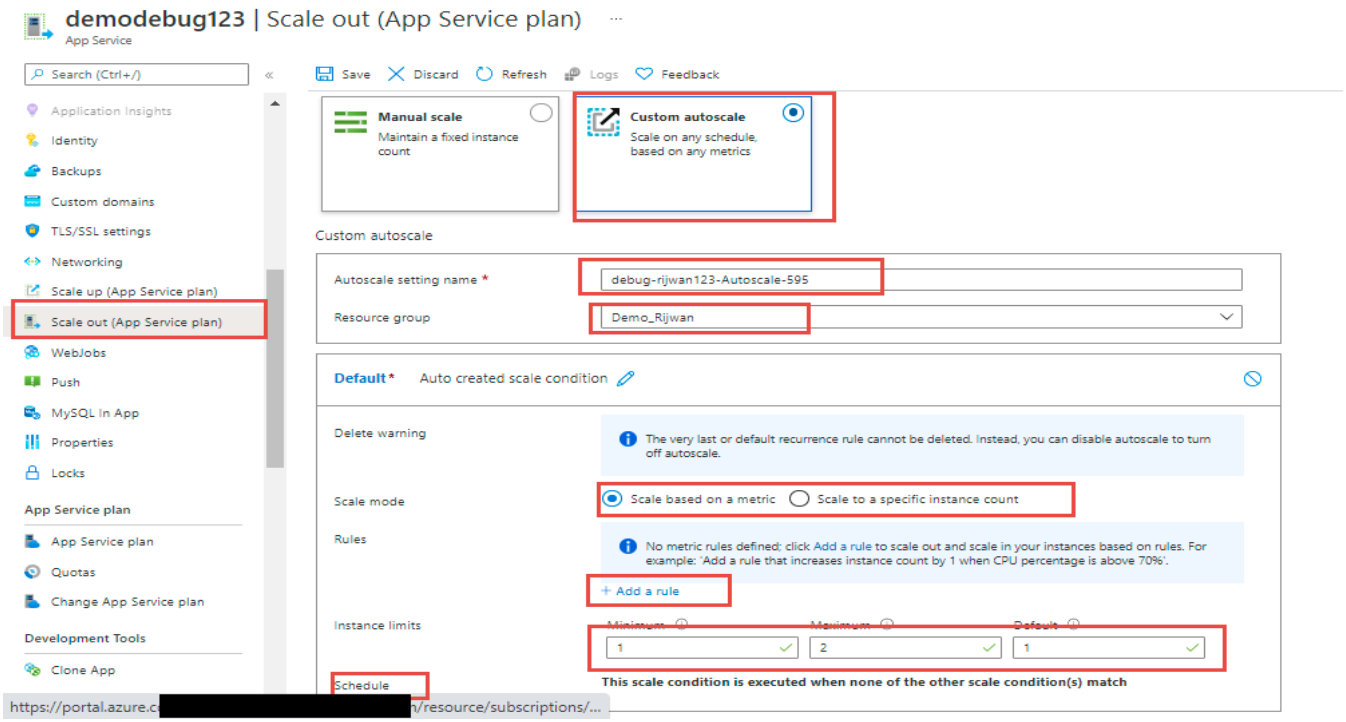
2. Configure Scale out by adding rules for custom scaling.

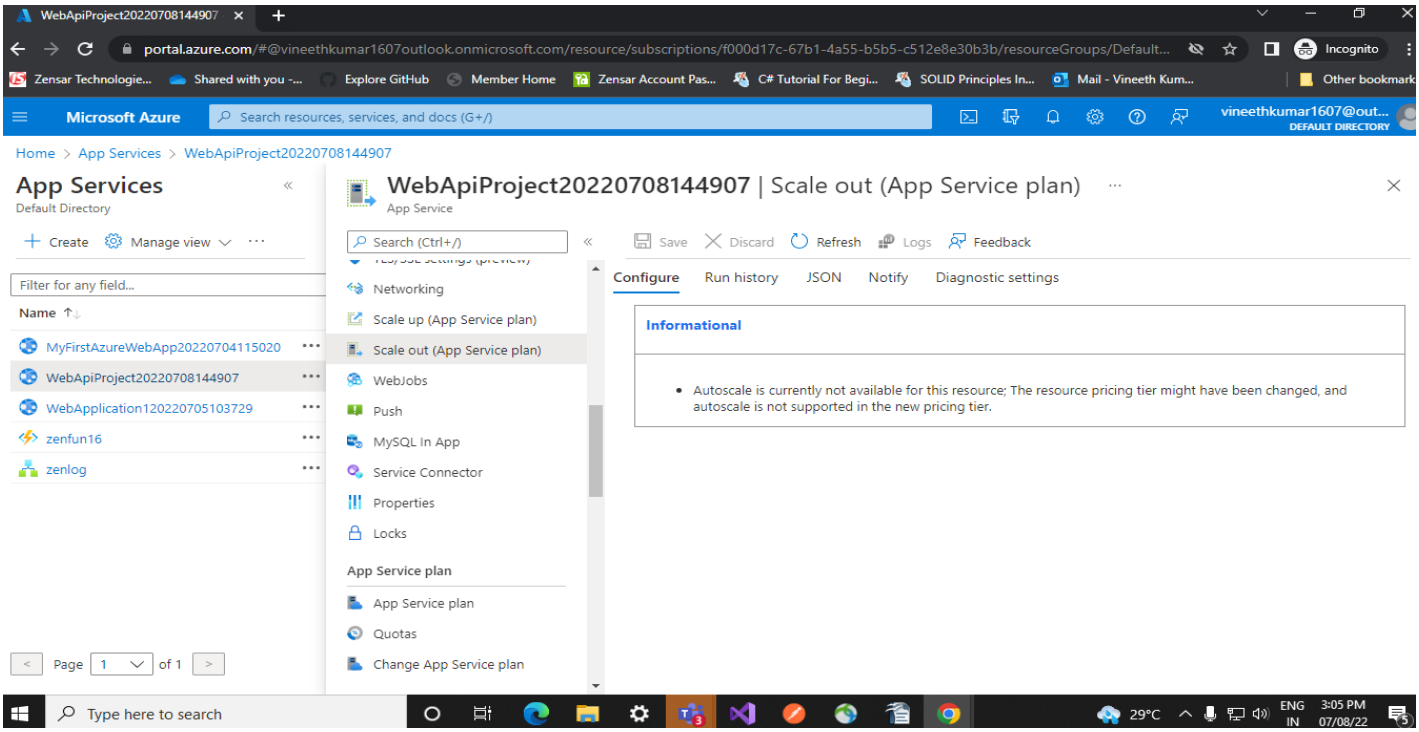
A scale out operation is **the equivalent of creating multiple copies of your web site and adding a load balancer to distribute the demand between them**.

Steps for scale out: -

1. Click on scale out
2. Select custom autoscale
3. In the rules section of the default scale condition, select Add a rule.
4. From the metric source dropdown, select current resource.
5. From resource type, select application Insights.
6. From the resource dropdown, select your App services plan standard metrics.
7. Select a metric name to CPU Percentage.
8. Select enable metric divide by instance count so that the number of sessions per instance is measured.
9. From the operator dropdown, select greater than.
10. Enter the metric threshold to trigger the scale action, for example, 70.
11. Under actions, set the operation to increase the count and set the Instance count to 1 and cool down by 5minutes and then click Add.
12. Set the maximum number of instances that can be spun up in the maximum field of the instance limits section for example, 1.
13. Click on save.

Note: - scale out is not available in my azure account. so collected it from internet



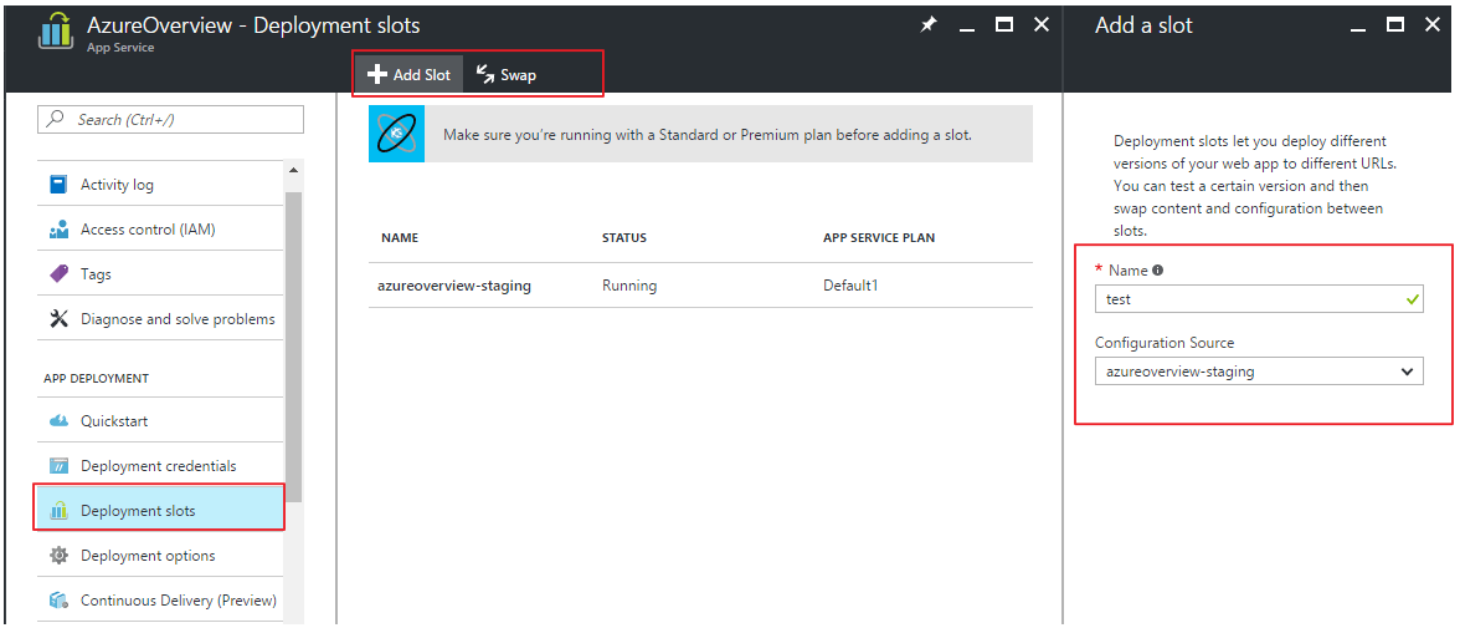


3. Configure Deployment slots for staging and production.

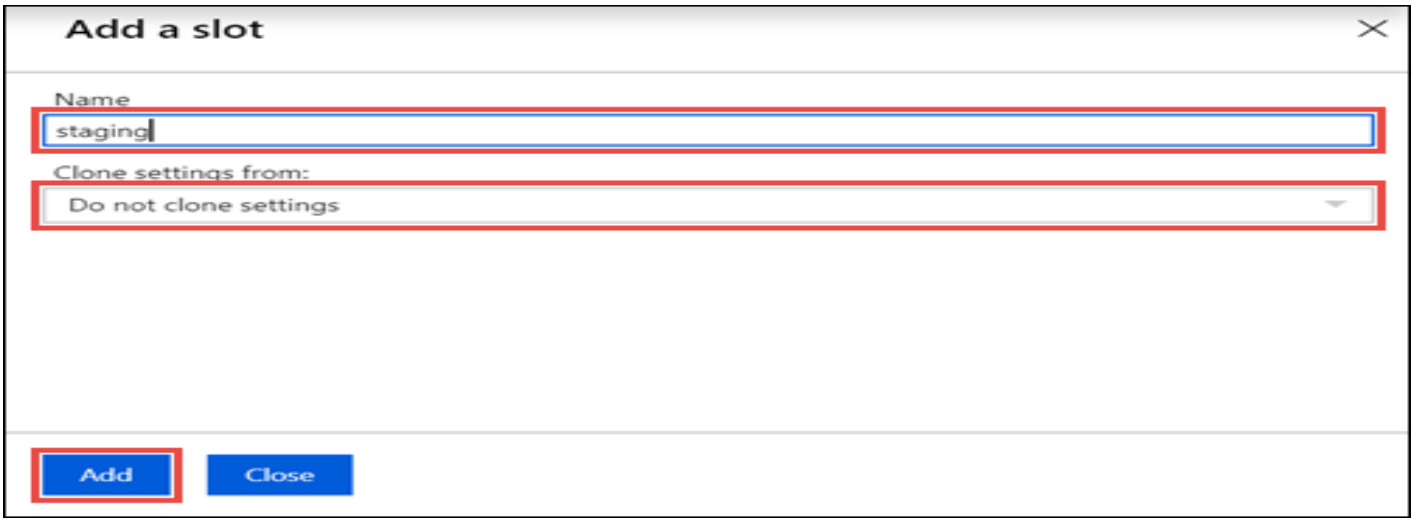
Deployment slots are **live apps with their own host names**. App content and configurations elements can be swapped between two deployment slots, including the production slot.

In Web App navigate and click on **deployment slots menu item**:

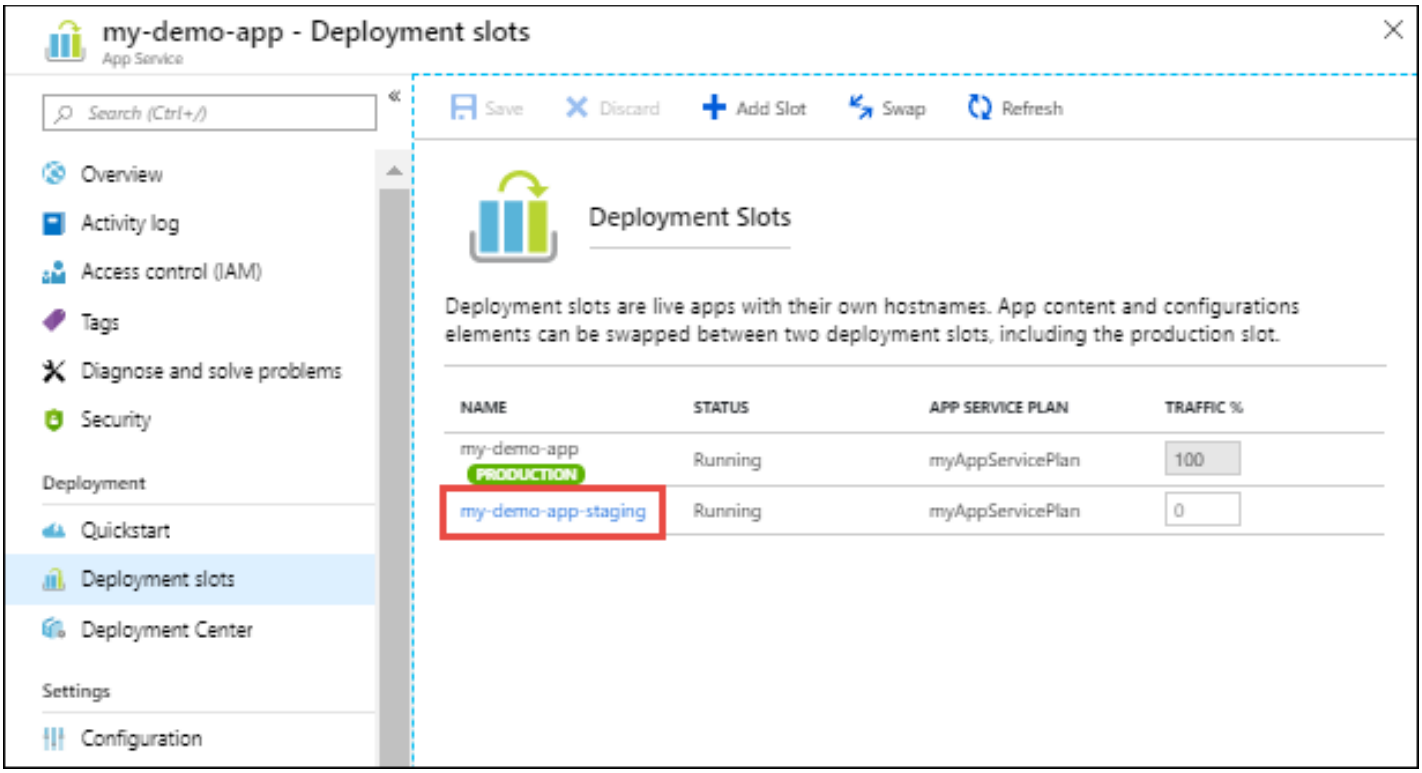
Note: - Deployment slots is not available in my azure account so I collected them from the internet



In the **Add a slot** dialog box, give the slot a name, and select whether to clone an app configuration from another deployment slot.

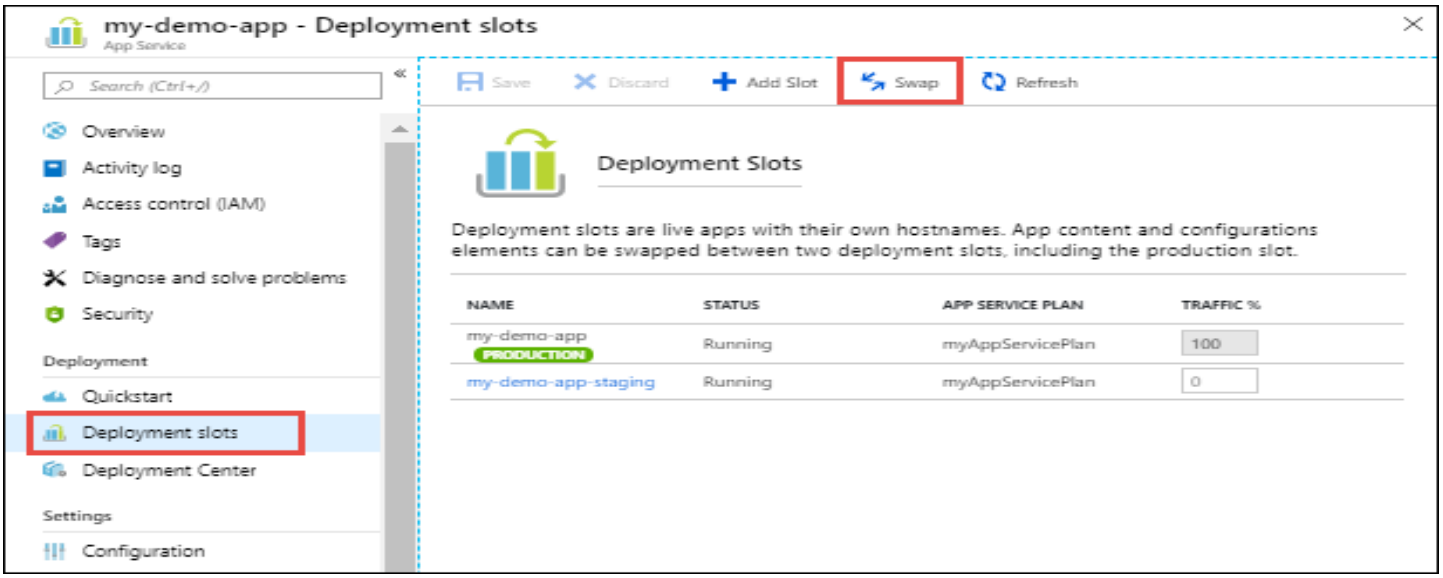


After the slot is added, select **Close** to close the dialog box. The new slot is now shown on the **Deployment slots** page.

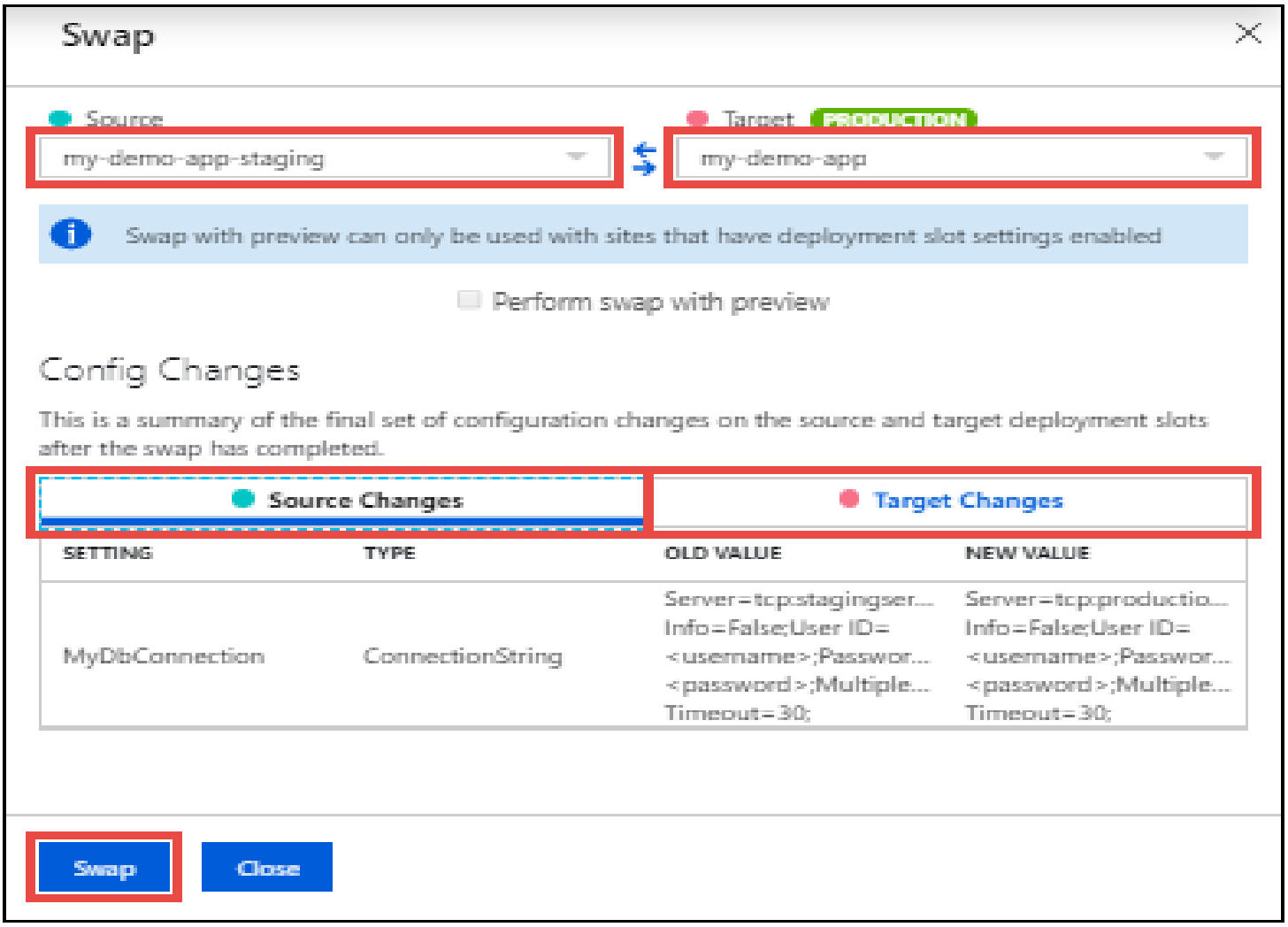


You can swap deployment slots on your app's **Deployment slots** page and the **Overview** page.

- Go to your app's **Deployment slots** page and select **Swap**.



- The **Swap** dialog box shows settings in the selected source and target slots that will be changed.
- Select the desired **Source** and **Target** slots. Usually, the target is the production slot. Also, select the **Source Changes** and **Target Changes** tabs and verify that the configuration changes are expected. When you're finished, you can swap the slots immediately by selecting **Swap**.

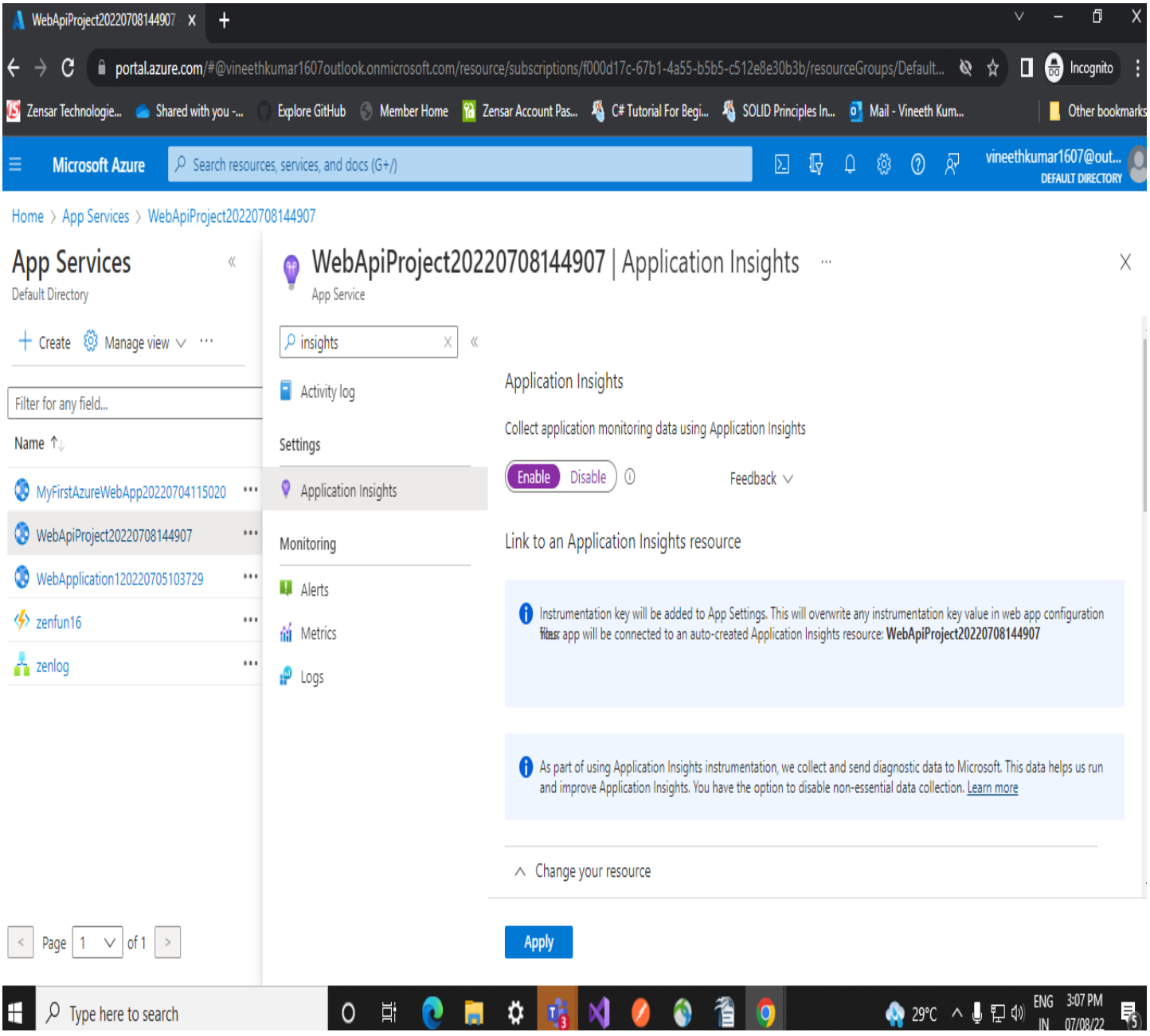


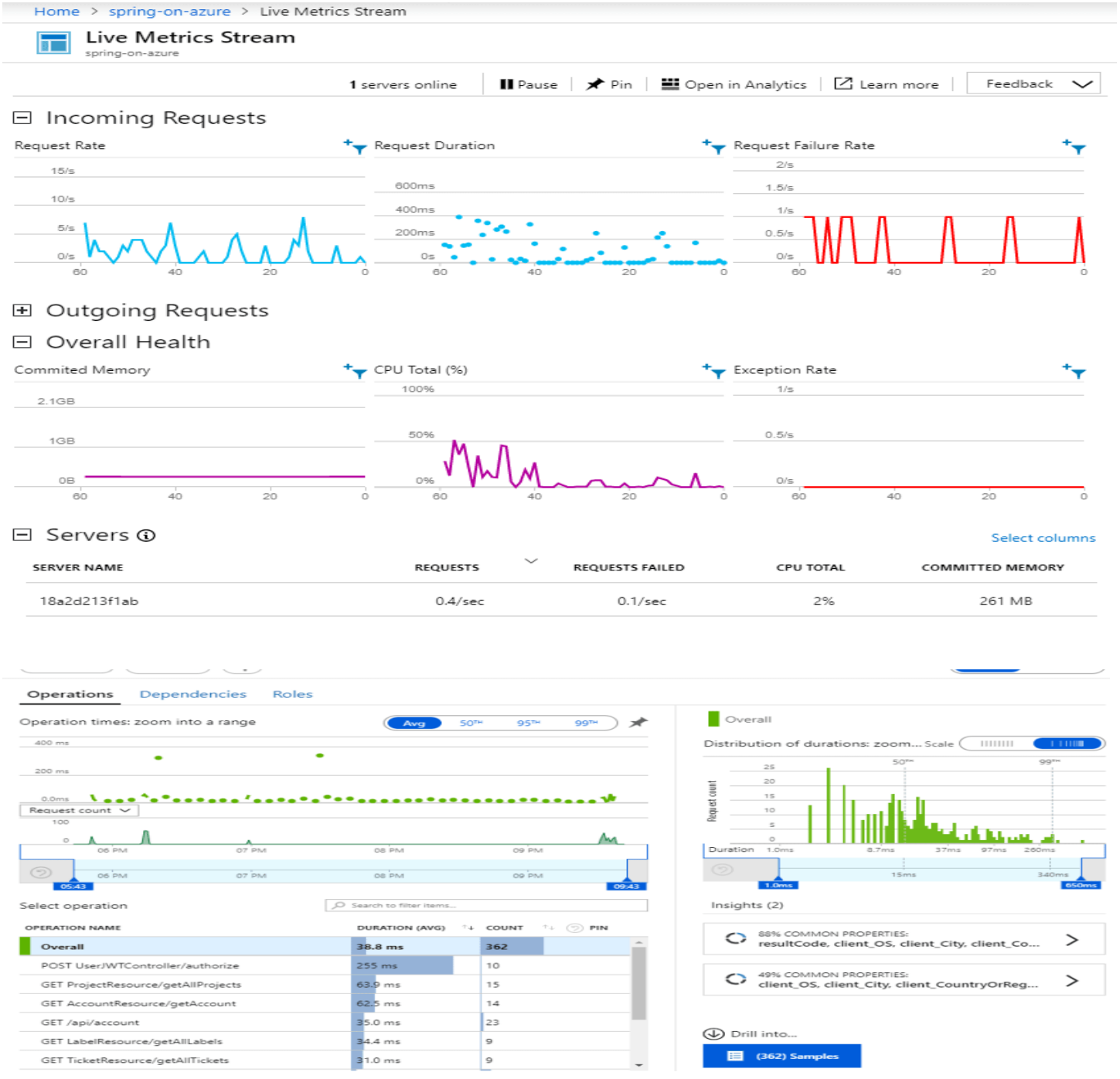
- To see how your target slot would run with the new settings before the swap actually happens, don't select **Swap**, but follow the instructions in Swap with preview.
- When you're finished, close the dialog box by selecting **Close**.

4. Configure Application Insights for the project

Application Insights is **a feature of Azure Monitor that provides extensible application performance management (APM) and monitoring for live web apps**

• Select the Application Insights




Home > spring-on-azure > Live Metrics Stream
Live Metrics Stream
spring-on-azure
1 servers online
Pause
Pin
Open in Analytics
Learn more
Feedback
Incoming Requests
Request Rate
15/s
10/s
5/s
0/s
Request Duration
600ms
400ms
200ms
0s
Request Failure Rate
2/s
1.5/s
1/s
0.5/s
0/s
60
40
20
0
Outgoing Requests
Overall Health
Commited Memory
2.1GB
1GB
0B
CPU Total (%)
100%
50%
0%
Exception Rate
1/s
0.5/s
0/s
Servers
Select columns
SERVER NAME
REQUESTS
REQUESTS FAILED
CPU TOTAL
COMMITTED MEMORY
18a2d213f1ab
0.4/sec
0.1/sec
2%
261 MB
Operations
Dependencies
Roles
Operation times: zoom into a range
Avg
50TH
95TH
99TH
400 ms
200 ms
0.0ms
Request count
100
0
06 PM
07 PM
08 PM
09 PM
05:43
09:43
Select operation
Search to filter items...
OPERATION NAME
DURATION (AVG)
COUNT
PIN
Overall
38.8 ms
362
POST User/JWTController/authorize
255 ms
10
GET ProjectResource/getAllProjects
63.9 ms
15
GET AccountResource/getAccount
62.5 ms
14
GET /api/account
35.0 ms
23
GET LabelResource/getAllLabels
34.4 ms
9
GET TicketResource/getAllTickets
31.0 ms
9
Overall
Distribution of durations: zoom...
Scale
50TH
99TH
25
20
15
10
5
0
Request count
Duration
1.0ms
8.7ms
37ms
97ms
260ms
15ms
340ms
650ms
Insights (2)
88% COMMON PROPERTIES:
resultCode, client_OS, client_City, client_Co...
49% COMMON PROPERTIES:
client_OS, client_City, client_CountryOrReg...
Drill into...
(362) Samples

5. Configure Swagger for the API

Swagger is **a language-agnostic specification for describing REST APIs**. It allows both computers and humans to understand the capabilities of a REST API without direct access to the source code. Its main goals are to: Minimize the amount of work needed to connect decoupled services.

Advantages:

1. Dependency Free - The UI works in any development environment, be it locally or in the web

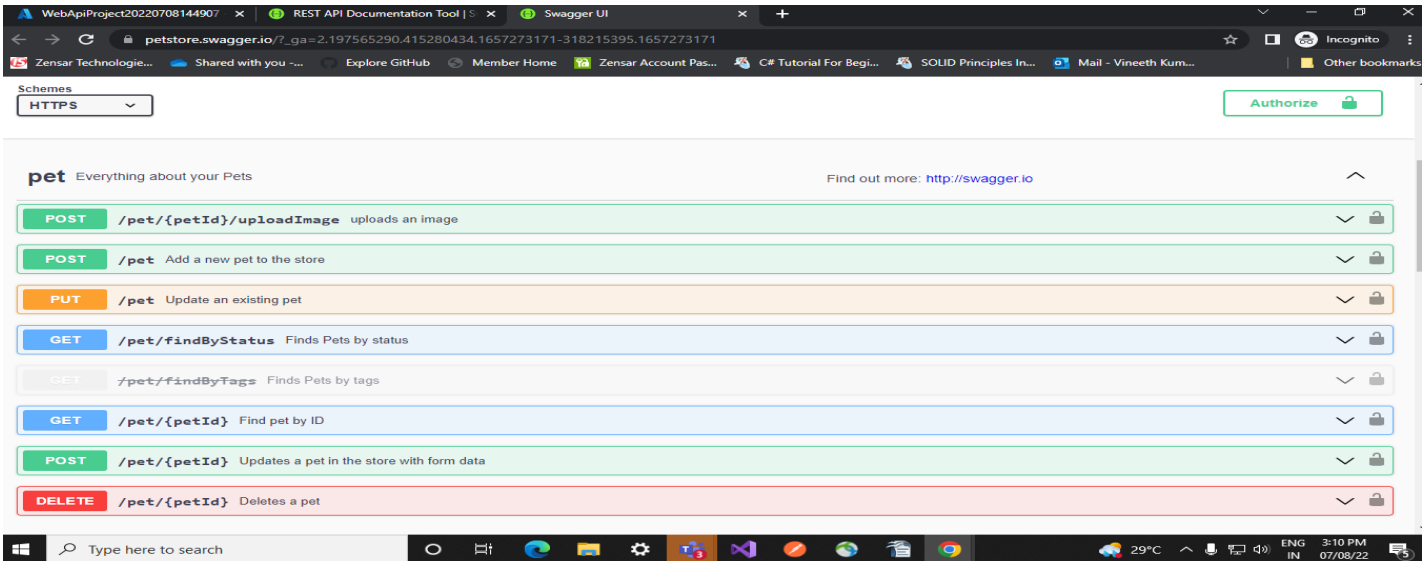
2. Human Friendly - Allow end developers to effortlessly interact and try out every single operation your API exposes for easy consumption

3. Easy to Navigate - Quickly find and work with resources and endpoints with neatly categorized documentation

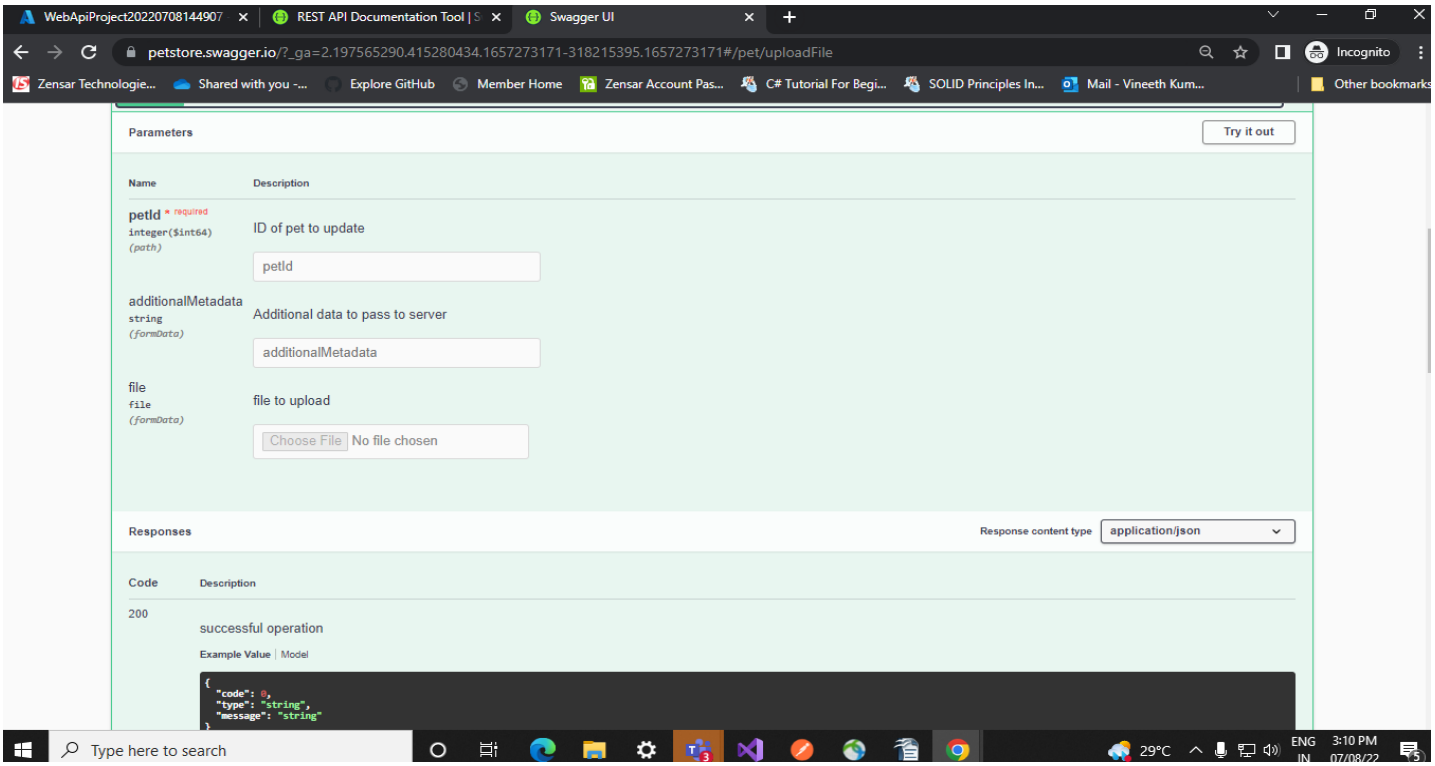
4. All Browser Support - Cater to every possible scenario with Swagger UI working in all major browsers.

5. Fully Customizable - Style and tweak your Swagger UI the way you want with full source code access.

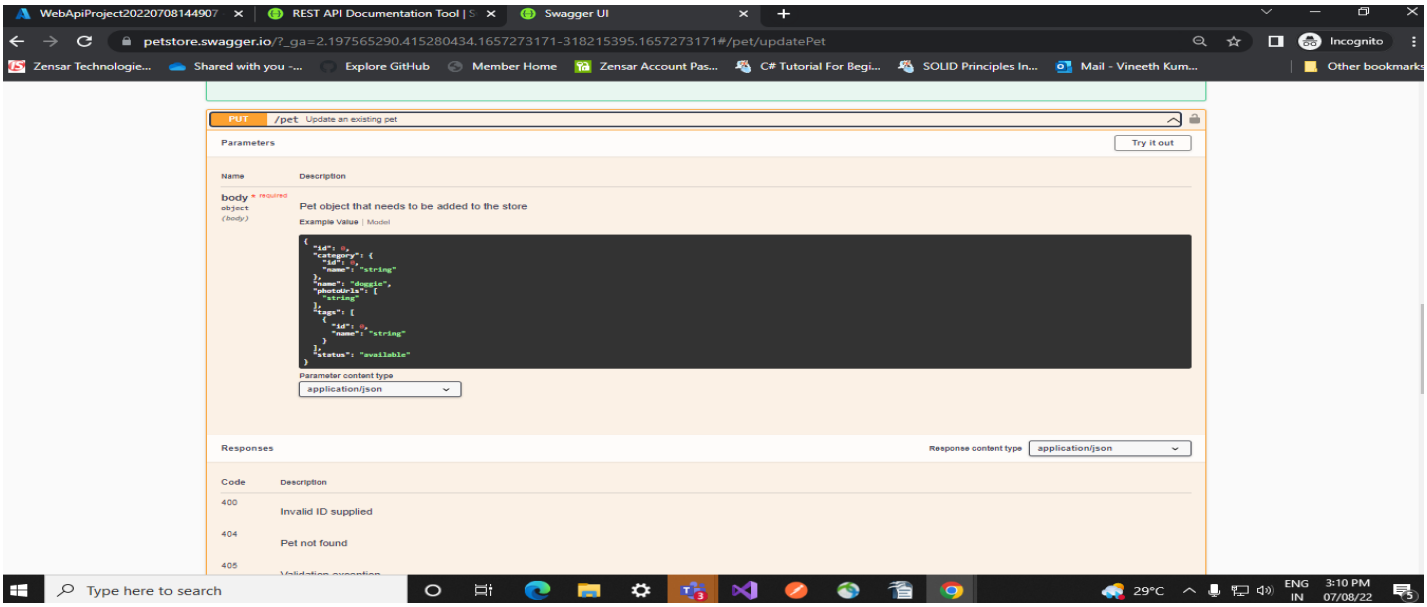
6. Complete OAS Support - Visualize APIs defined in Swagger 2.0 or OAS 3.0


WebApiProject20220708144907
REST API Documentation Tool | S
Swagger UI
petstore.swagger.io/?_ga=2.197565290.415280434.1657273171-318215395.1657273171
Incognito
Zensar Technologie...
Shared with you -...
Explore GitHub
Member Home
Zensar Account Pas...
C# Tutorial For Begi...
SOLID Principles In...
Mail - Vineeth Kum...
Other bookmarks
Schemes
HTTPS
Authorize
pet Everything about your Pets
Find out more: http://swagger.io
POST /pet/{petId}/uploadImage uploads an image
POST /pet Add a new pet to the store
PUT /pet Update an existing pet
GET /pet/findByStatus Finds Pets by status
GET /pet/findByTags Finds Pets by tags
GET /pet/{petId} Find pet by ID
POST /pet/{petId} Updates a pet in the store with form data
DELETE /pet/{petId} Deletes a pet
Type here to search
29°C
ENG
IN
3:10 PM
07/08/22

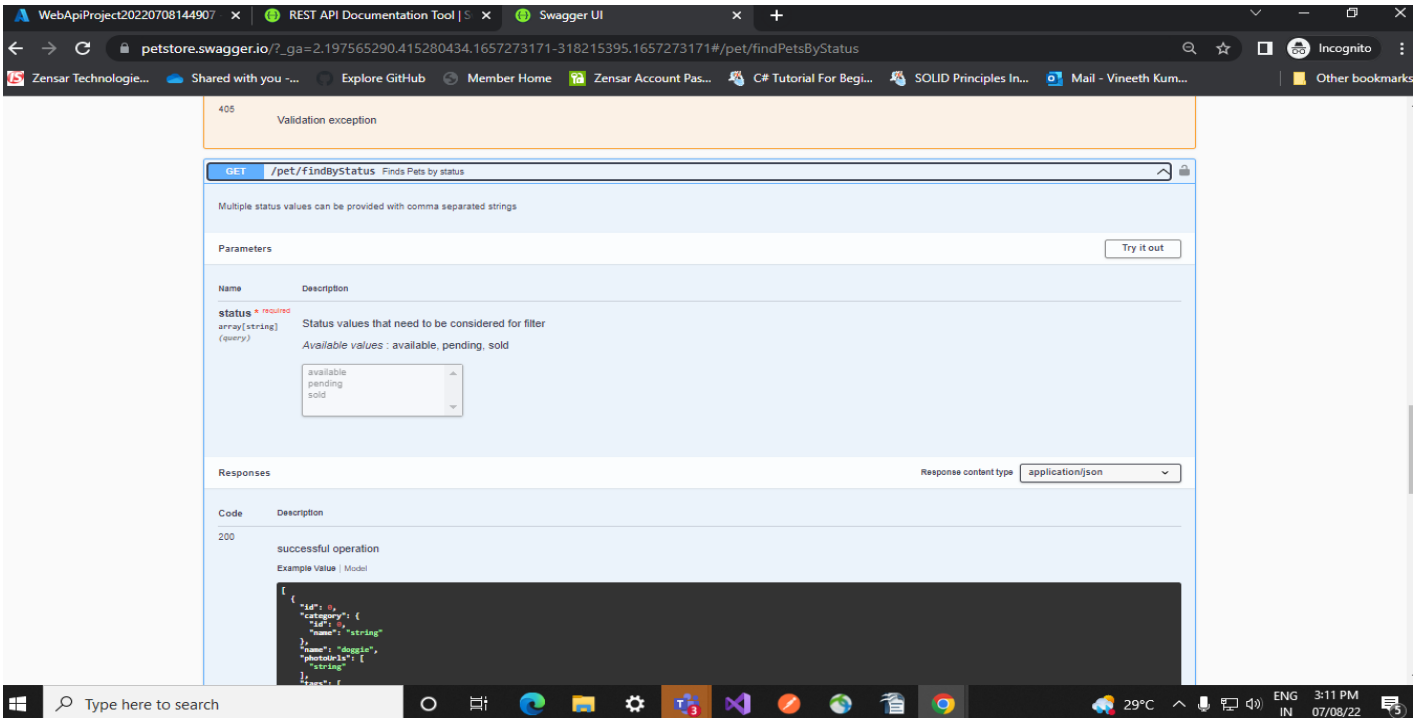
Post


WebApiProject20220708144907
REST API Documentation Tool | S
Swagger UI
petstore.swagger.io/?_ga=2.197565290.415280434.1657273171-318215395.1657273171#/pet/uploadFile
Incognito
Zensar Technologie...
Shared with you -...
Explore GitHub
Member Home
Zensar Account Pas...
C# Tutorial For Begi...
SOLID Principles In...
Mail - Vineeth Kum...
Other bookmarks
Parameters
Try it out
Name
Description
petId * required
integer($int64)
(path)
ID of pet to update
petId
additionalMetadata
string
(formData)
Additional data to pass to server
additionalMetadata
file
file
(formData)
file to upload
Choose File No file chosen
Responses
Response content type
application/json
Code
Description
200
successful operation
Example Value | Model
{
"code": 0,
"type": "string",
"message": "string"
}
Type here to search
29°C
ENG
IN
3:10 PM
07/08/22

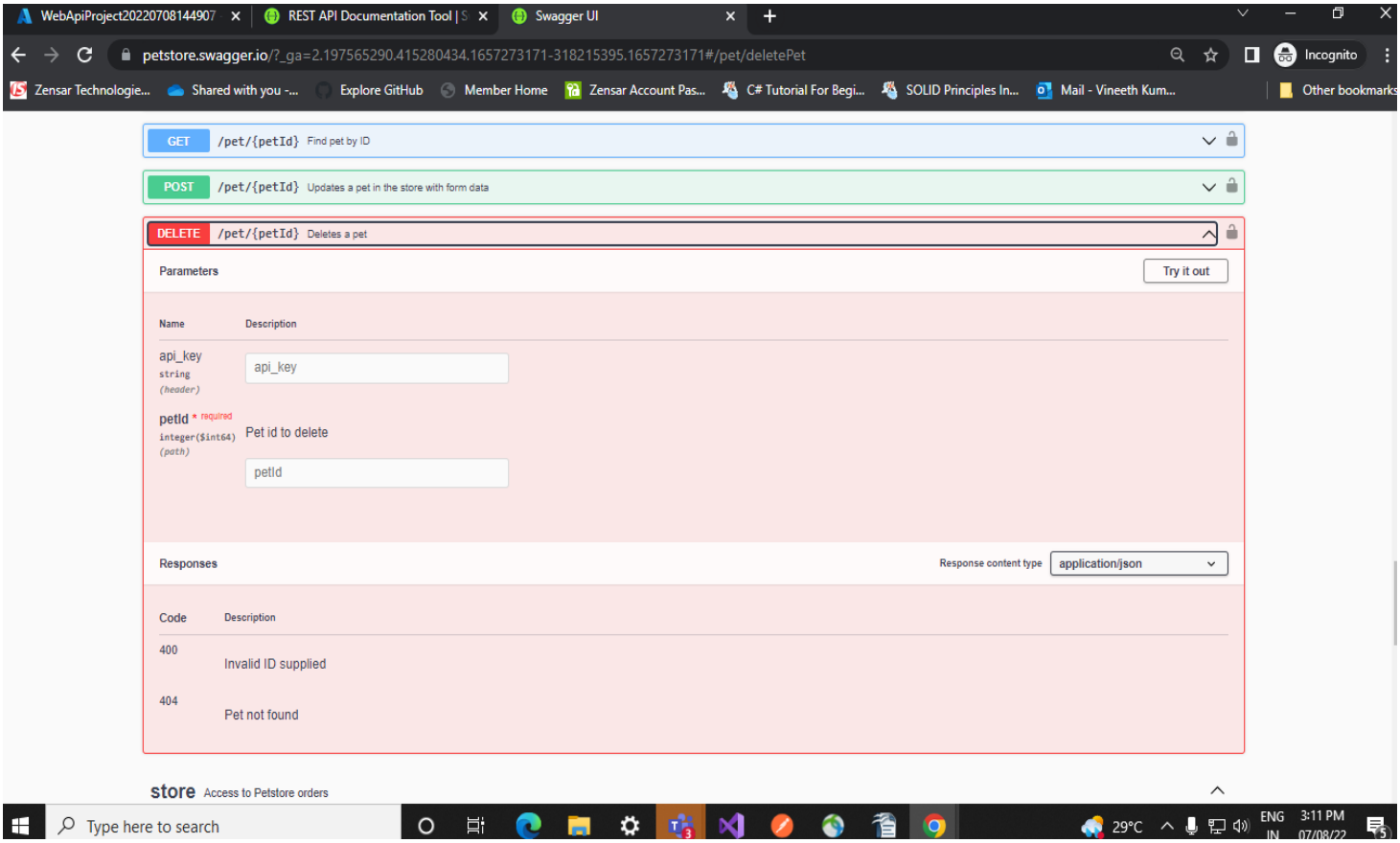
Put


WebApiProject20220708144907
REST API Documentation Tool | S
Swagger UI
petstore.swagger.io/?_ga=2.197565290.415280434.1657273171-318215395.1657273171#/pet/updatePet
Incognito
Zensar Technologie...
Shared with you -...
Explore GitHub
Member Home
Zensar Account Pas...
C# Tutorial For Begi...
SOLID Principles In...
Mail - Vineeth Kum...
Other bookmarks
PUT /pet Update an existing pet
Parameters
Try it out
Name
Description
body * required
object
(body)
Pet object that needs to be added to the store
Example Value | Model
{
"id": 0,
"category": {
"id": 0,
"name": "string"
},
"name": "doggie",
"photoUrls": [
"string"
],
"tags": [
{
"id": 0,
"name": "string"
}
],
"status": "available"
}
Parameter content type
application/json
Responses
Response content type
application/json
Code
Description
400
Invalid ID supplied
404
Pet not found
405
Type here to search
29°C
ENG
IN
3:10 PM
07/08/22

Get


WebApiProject20220708144907
REST API Documentation Tool | S
Swagger UI
petstore.swagger.io/?_ga=2.197565290.415280434.1657273171-318215395.1657273171#/pet/findPetsByStatus
Incognito
Zensar Technologie...
Shared with you -...
Explore GitHub
Member Home
Zensar Account Pas...
C# Tutorial For Begi...
SOLID Principles In...
Mail - Vineeth Kum...
Other bookmarks
405
Validation exception
GET /pet/findByStatus Finds Pets by status
Multiple status values can be provided with comma separated strings
Parameters
Try it out
Name
Description
status * required
array[string]
(query)
Status values that need to be considered for filter
Available values : available, pending, sold
available
pending
sold
Responses
Response content type
application/json
Code
Description
200
successful operation
Example Value | Model
[
{
"id": 0,
"category": {
"id": 0,
"name": "string"
},
"name": "doggie",
"photoUrls": [
"string"
],
Type here to search
29°C
ENG
IN
3:11 PM
07/08/22

Delete


WebApiProject20220708144907
REST API Documentation Tool | S
Swagger UI
petstore.swagger.io/?_ga=2.197565290.415280434.1657273171-318215395.1657273171#/pet/deletePet
Incognito
Zensar Technologie...
Shared with you -...
Explore GitHub
Member Home
Zensar Account Pas...
C# Tutorial For Begi...
SOLID Principles In...
Mail - Vineeth Kum...
Other bookmarks
GET /pet/{petId} Find pet by ID
POST /pet/{petId} Updates a pet in the store with form data
DELETE /pet/{petId} Deletes a pet
Parameters
Try it out
Name
Description
api_key
string
(header)
api_key
petId * required
integer($int64)
(path)
Pet id to delete
petId
Responses
Response content type
application/json
Code
Description
400
Invalid ID supplied
404
Pet not found
store Access to Petstore orders
Type here to search
29°C
ENG
IN
3:11 PM
07/08/22

6.Work with Log Analytics with the sample logs available

Log Analytics is a tool in the Azure portal to edit and run log queries from data collected by Azure Monitor logs and interactively analyze their results. You can use Log Analytics queries to retrieve records that match particular criteria, identify trends, analyze patterns, and provide various insights into your data.

• 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.

• All queries return records generated within a set time range. By default, the query returns records generated in the last 24 hours. You can set a different time range by using the where operator in the query. You can also use the Time range dropdown list at the top of the screen. Change the time range of the query by selecting Last 7 days from the Time range dropdown. Select Run to return the results.

• 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.