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 with a new project on visual studo, with project type as **ASP.NET Core Web API**.
2. Create a new **Folder** called **Models** with class **Products** with the given specifications, added all the **validation attributes** on to the fields of the Product class.
3. Added required **entity framework** libraries using **nuget package** manager.
4. Performed data base **migration** operations on PS console.
5. Created a **controller** by right clicking on Controller and selecting the option Conroller, **create a controller** with the created model class by using **scaffolidng** 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

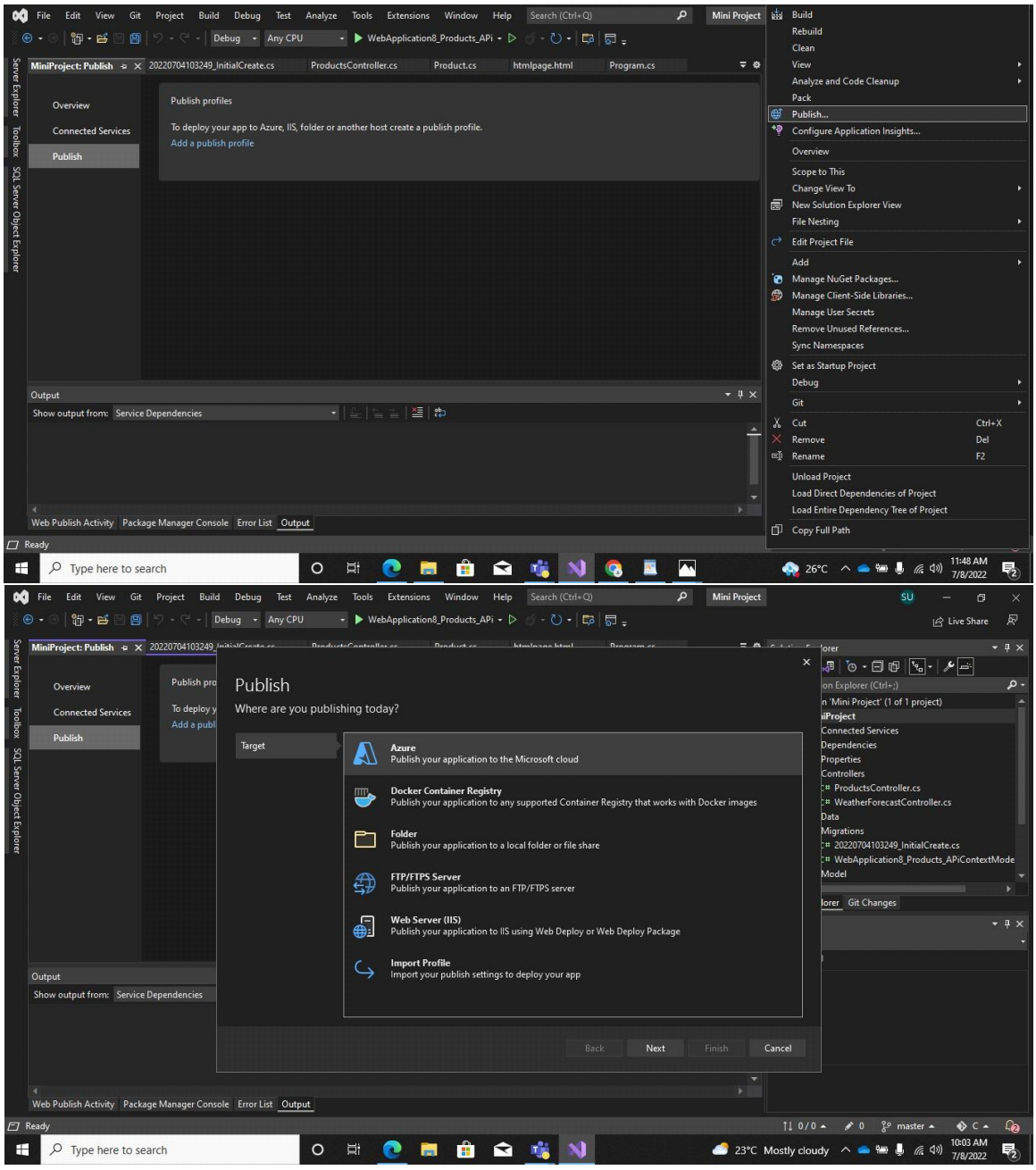
·Configure Deployment slots for staging and production

·Configure Application Insights for the project

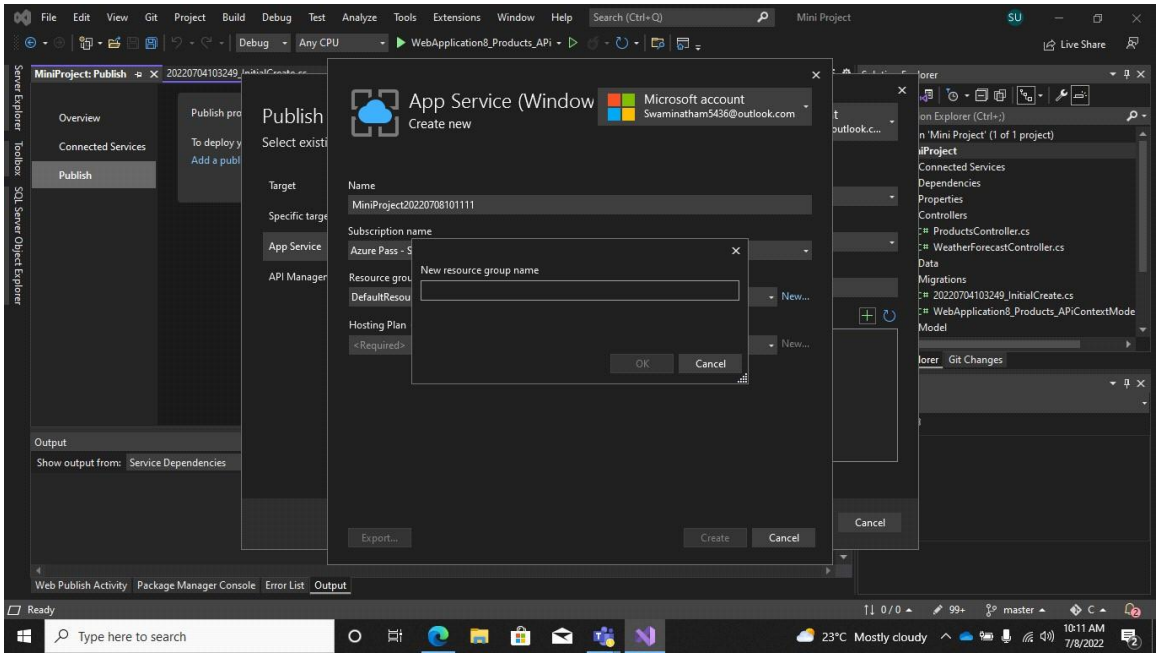
·Configure Swagger for the api

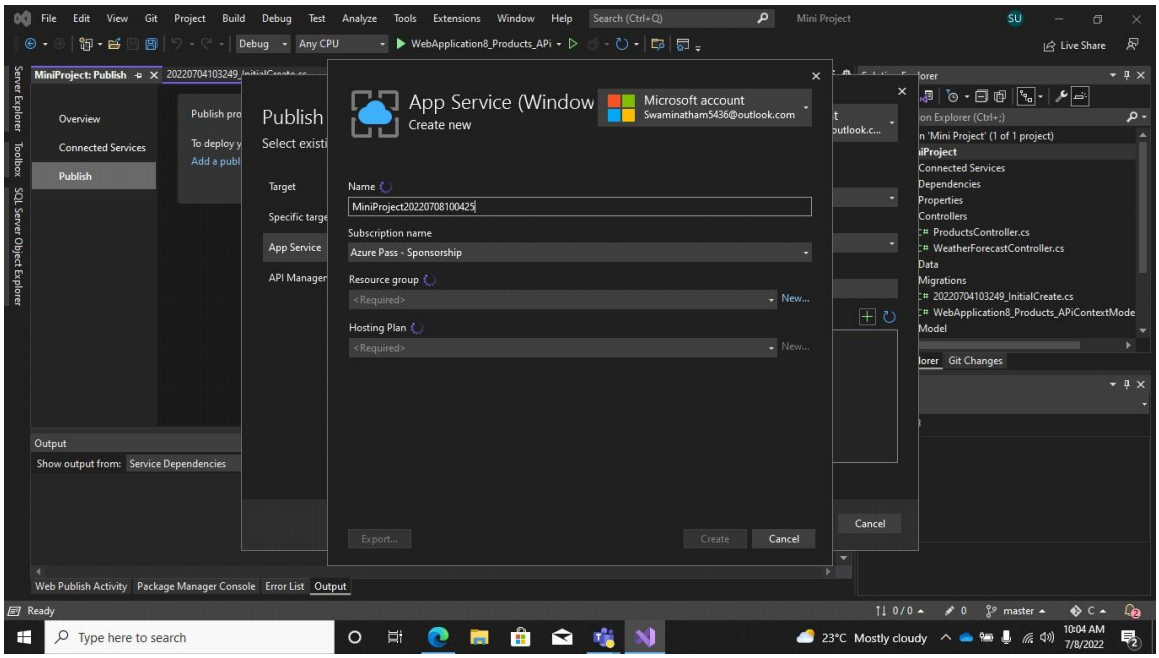
·Work with Log Analytics with the sample logs available.

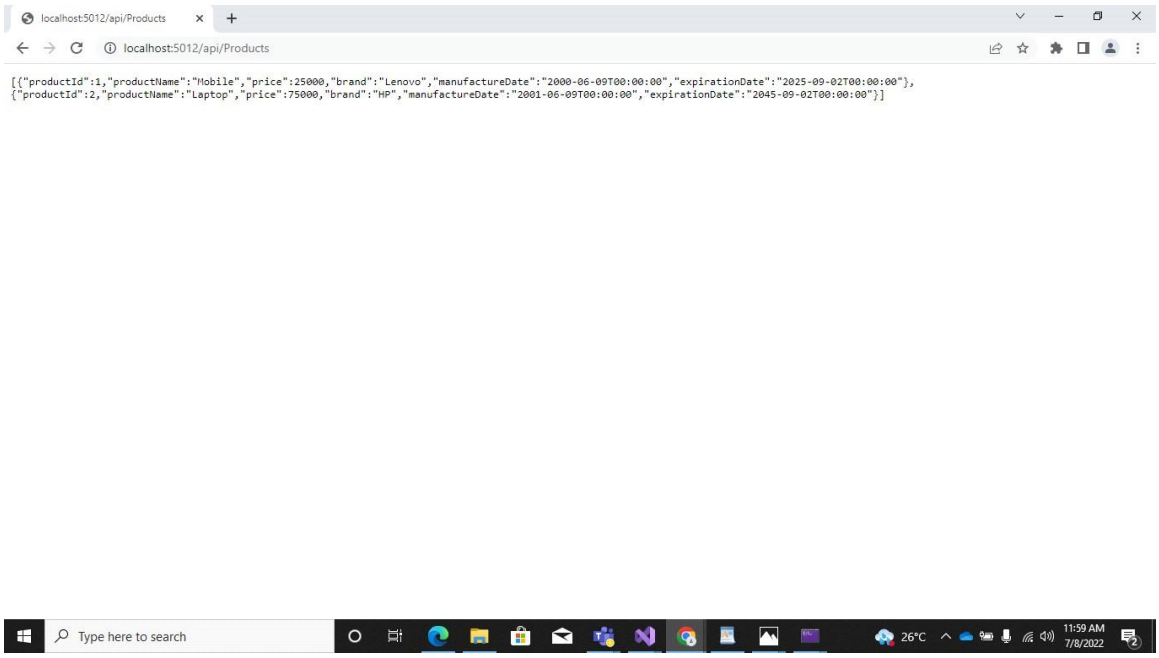
1.Host the Web API in azure and consume the same using jQuery Client.


Mini Project
Publish...
Publish profiles
To deploy your app to Azure, IIS, folder or another host create a publish profile.
Add a publish profile
Overview
Connected Services
Publish
Output
Show output from: Service Dependencies
Web Publish Activity
Package Manager Console
Error List
Output
Ready
Type here to search
26°C
11:48 AM
7/8/2022

Publish
Where are you publishing today?
Target
Azure
Publish your application to the Microsoft cloud
Docker Container Registry
Publish your application to any supported Container Registry that works with Docker images
Folder
Publish your application to a local folder or file share
FTP/FTPS Server
Publish your application to an FTP/FTPS server
Web Server (IIS)
Publish your application to IIS using Web Deploy or Web Deploy Package
Import Profile
Import your publish settings to deploy your app
Back
Next
Finish
Cancel
Mini Project
Live Share
Output
Show output from: Service Dependencies
Web Publish Activity
Package Manager Console
Error List
Output
Ready
Type here to search
23°C Mostly cloudy
10:03 AM
7/8/2022

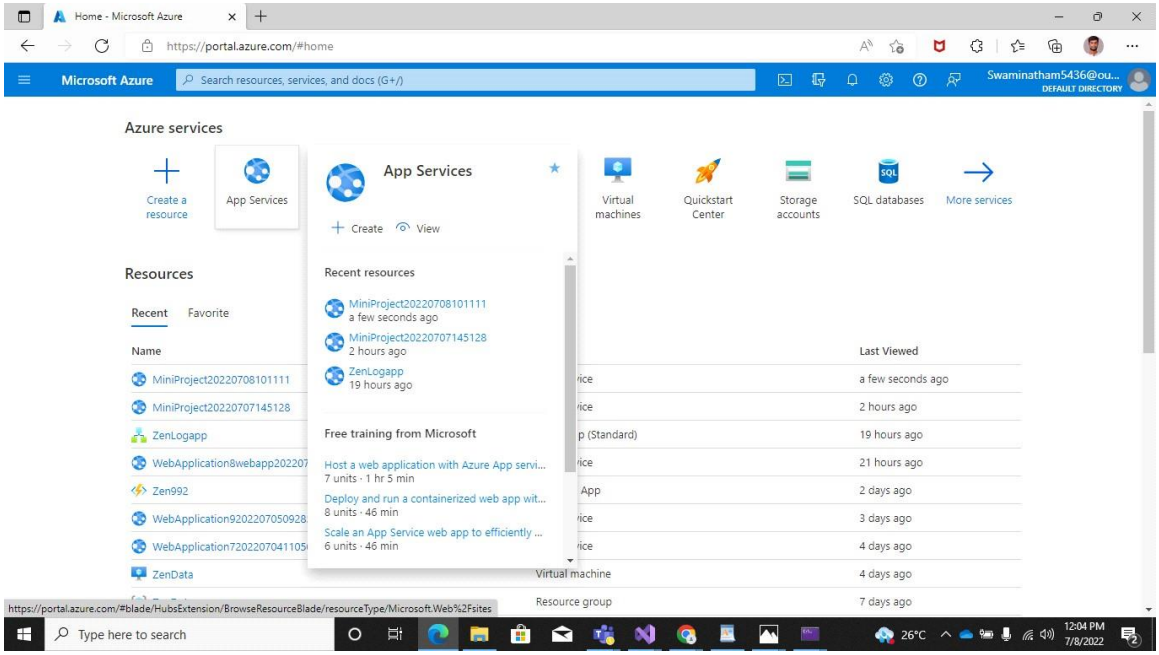

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Mini Project
Debug
Any CPU
WebApplication8_Products_APi
Live Share
MiniProject: Publish
Overview
Connected Services
Publish
Publish
Select existi
Target
Specific targe
App Service
API Manager
App Service (Window
Create new
Microsoft account
Swaminatham5436@outlook.com
Name
MiniProject20220708101111
Subscription name
Azure Pass - S
Resource grou
DefaultResou
New...
Hosting Plan
<Required>
New resource group name
OK
Cancel
Export...
Create
Cancel
Output
Show output from: Service Dependencies
Web Publish Activity
Package Manager Console
Error List
Output
Ready
master
Type here to search
23°C Mostly cloudy
10:11 AM
7/8/2022

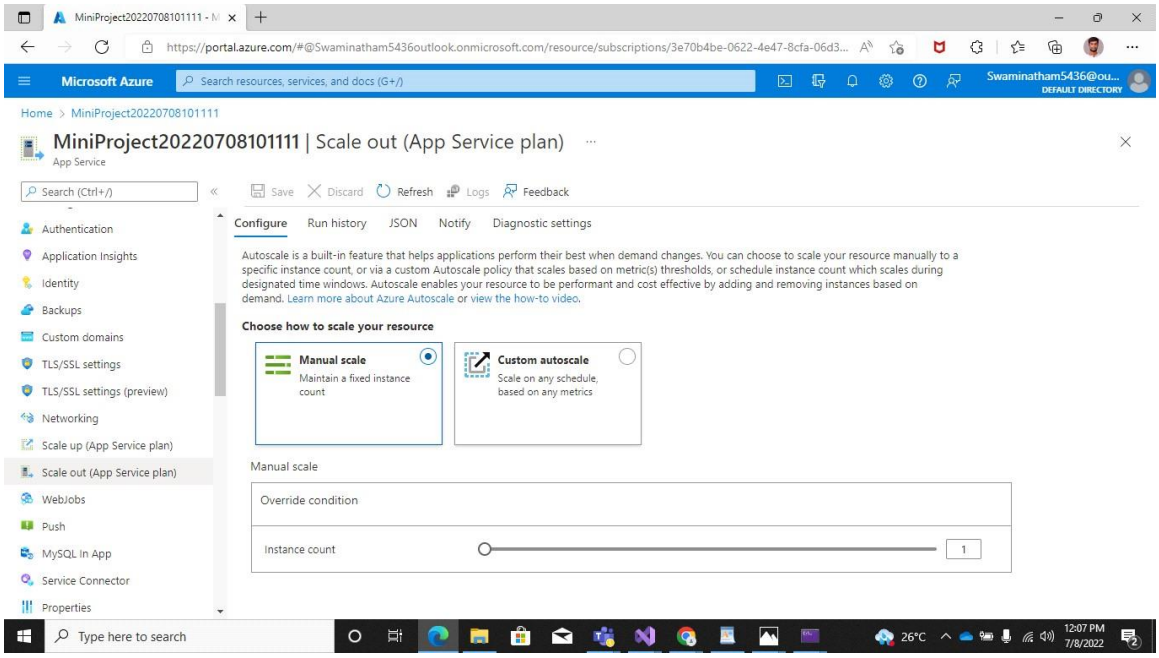

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Mini Project
Debug
Any CPU
WebApplication8_Products_APi
Live Share
MiniProject: Publish
Overview
Connected Services
Publish
Publish
Select existi
Target
Specific targe
App Service
API Manager
App Service (Window
Create new
Microsoft account
Swaminatham5436@outlook.com
Name
MiniProject20220708100425
Subscription name
Azure Pass - Sponsorship
Resource group
<Required>
New...
Hosting Plan
<Required>
New...
Export...
Create
Cancel
Output
Show output from: Service Dependencies
Web Publish Activity
Package Manager Console
Error List
Output
Ready
master
Type here to search
23°C Mostly cloudy
10:04 AM
7/8/2022

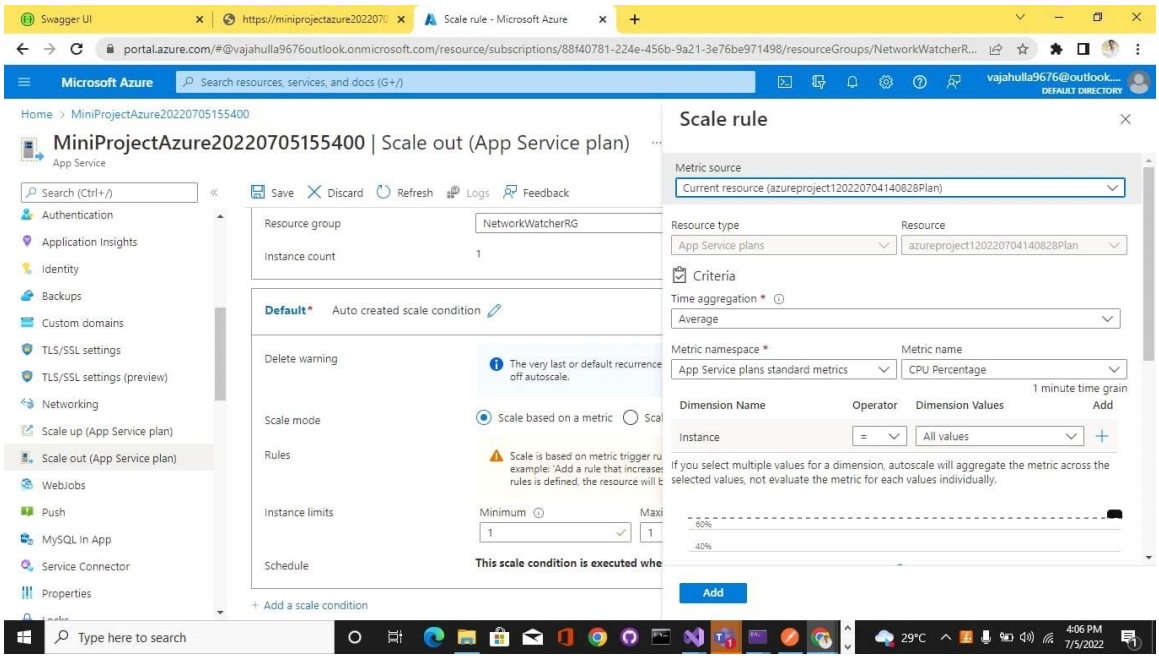


This image shows the localhost5012/API/Products.

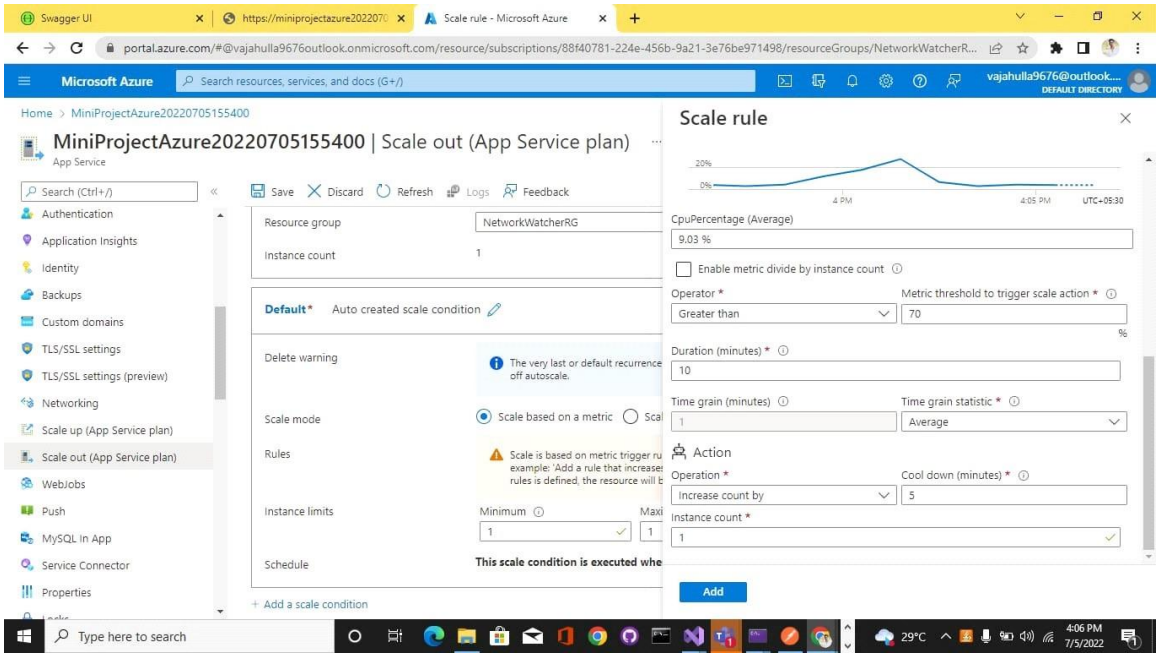
2. Configure Scale out by adding rules for custom scaling

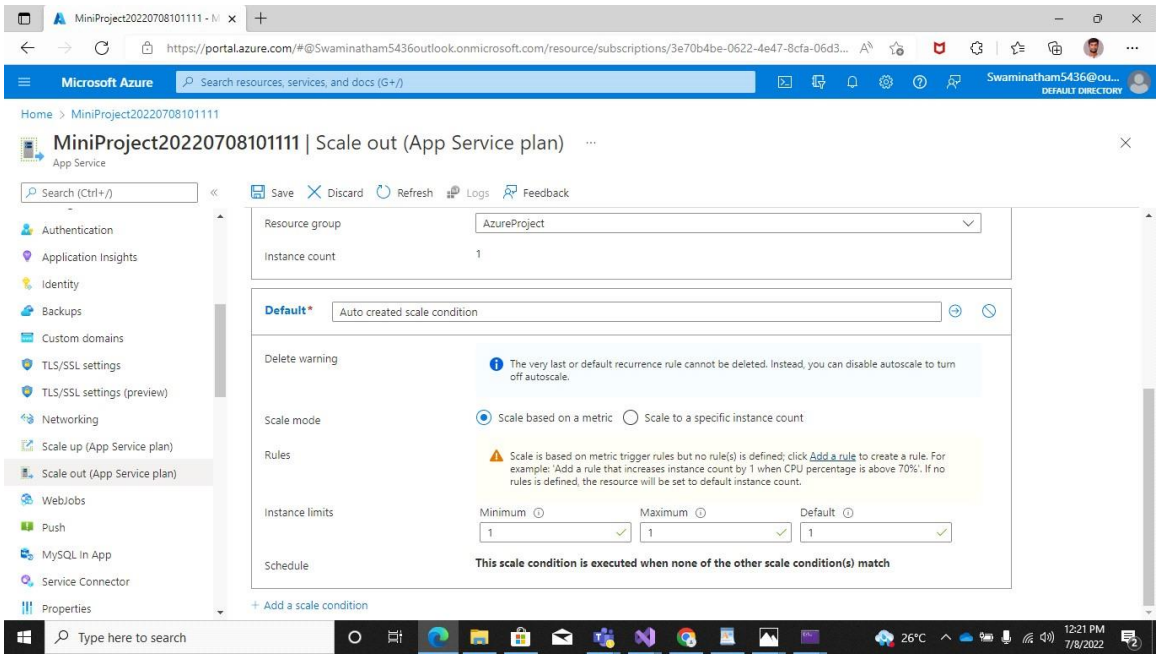






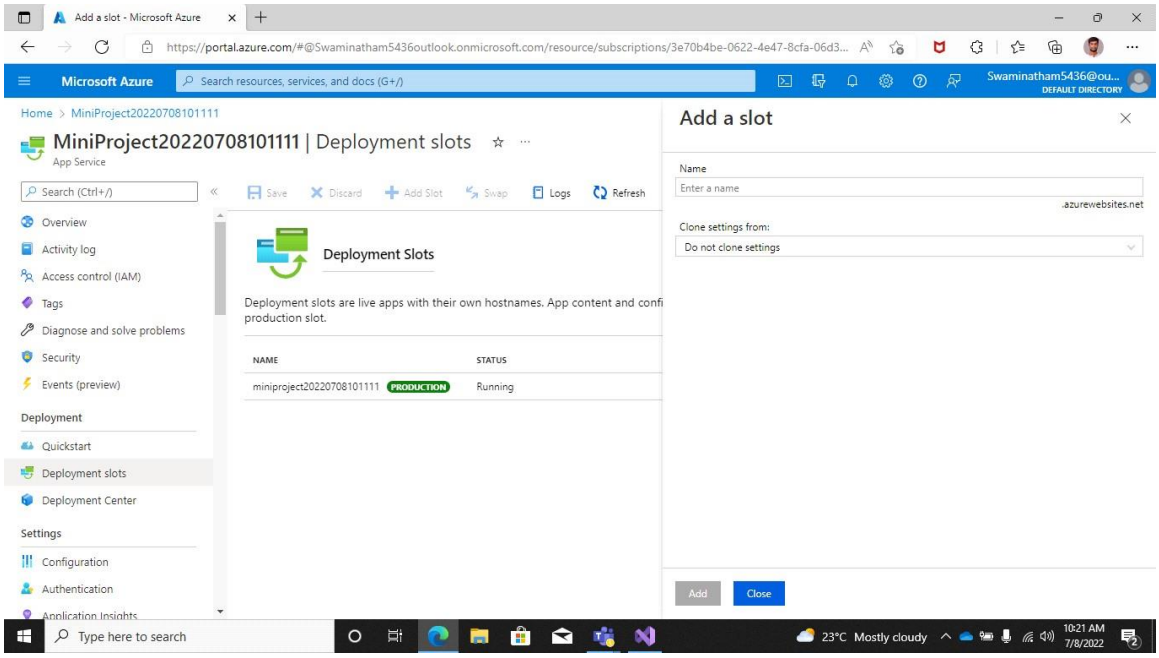
These two images are showing App Service plan of Scale rule

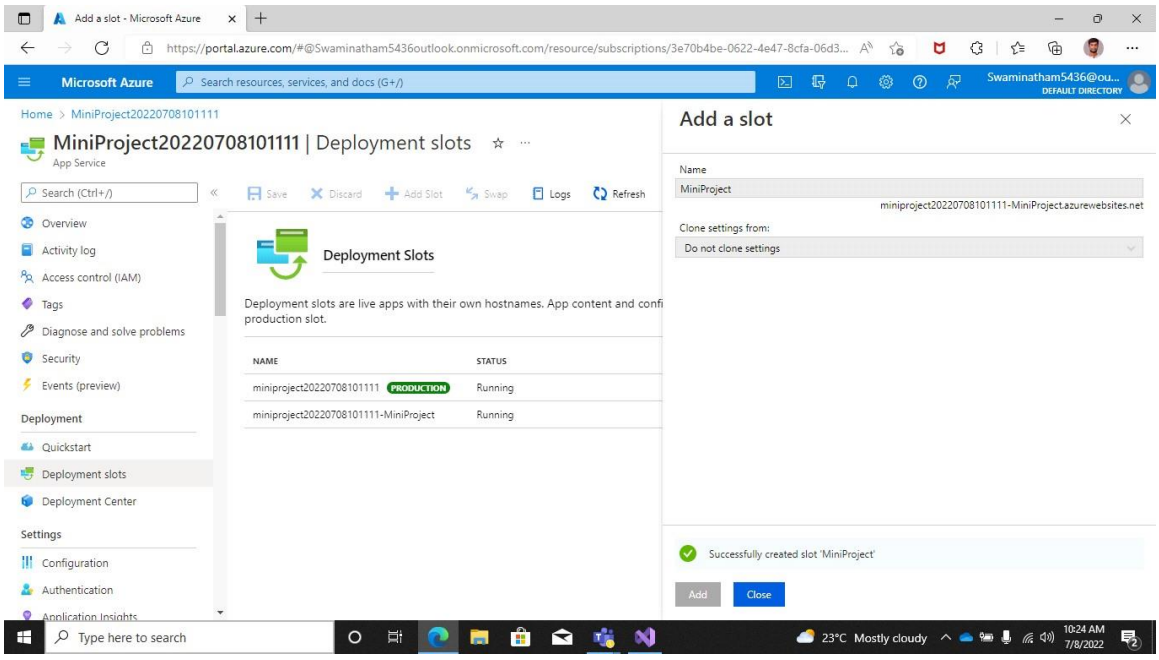




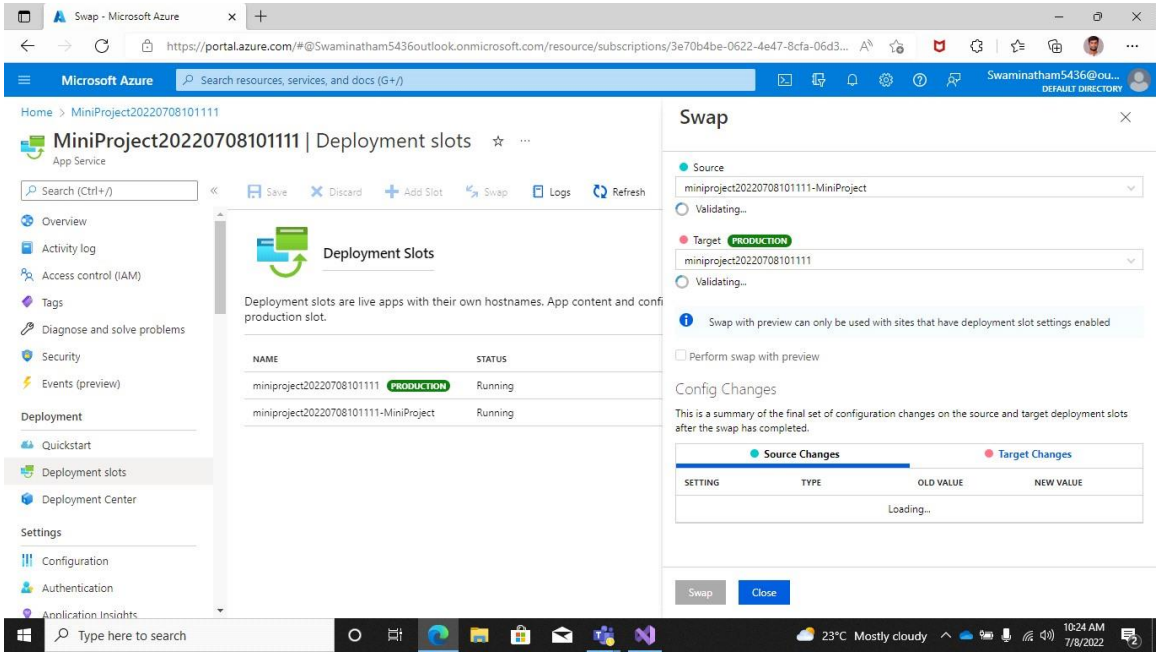
3.Configure Deployment slots for staging and production

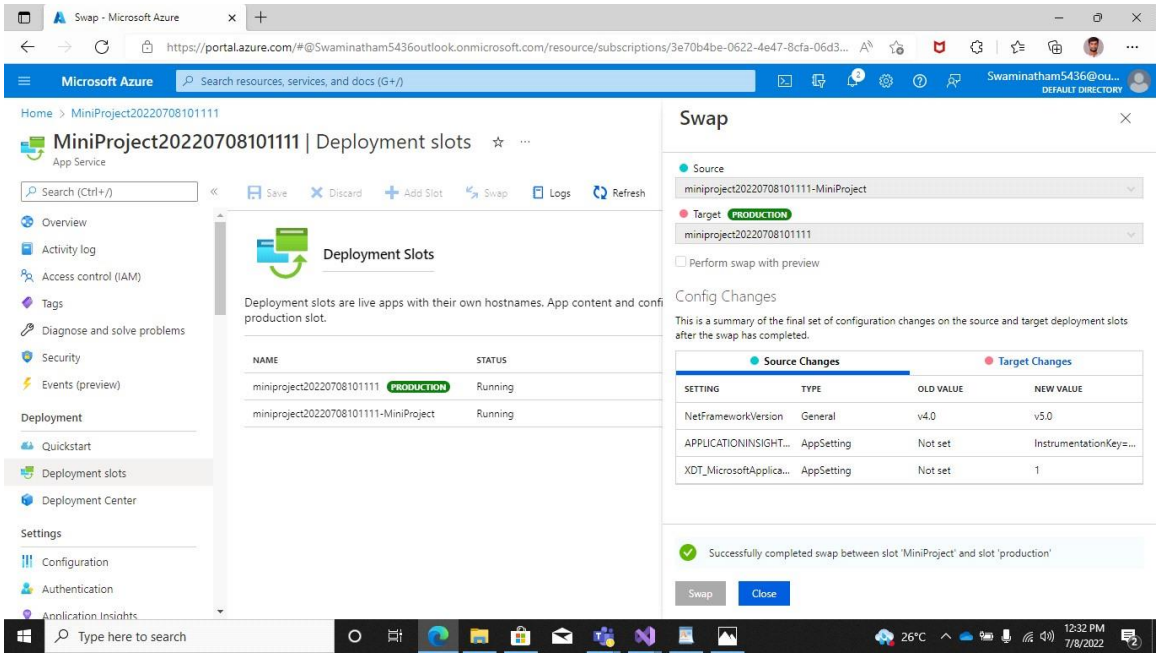
Azure Functions deployment slots allow your function app to run different instances called "slots". Slots are different environments exposed via a publicly available endpoint. One app instance is always mapped to the production slot, and you can swap instances assigned to a slot on demand. Function apps running under the Apps Service plan may have multiple slots, while under the Consumption plan only one slot is allowed.





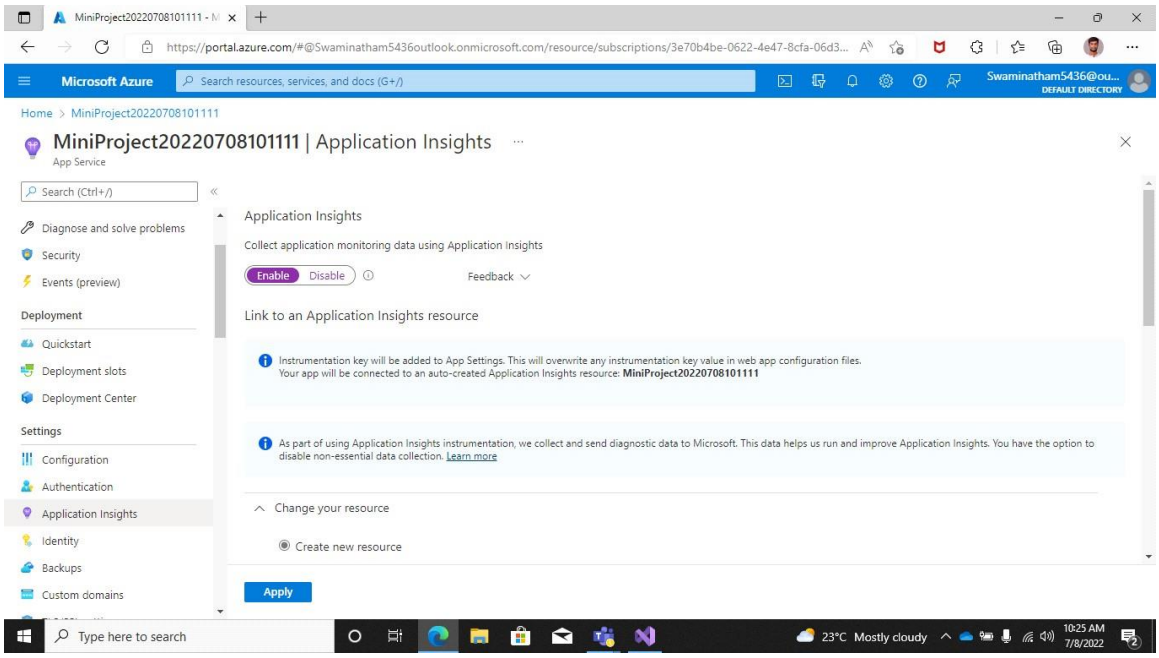
These two images show the Deployment slots of Add a Slot creating

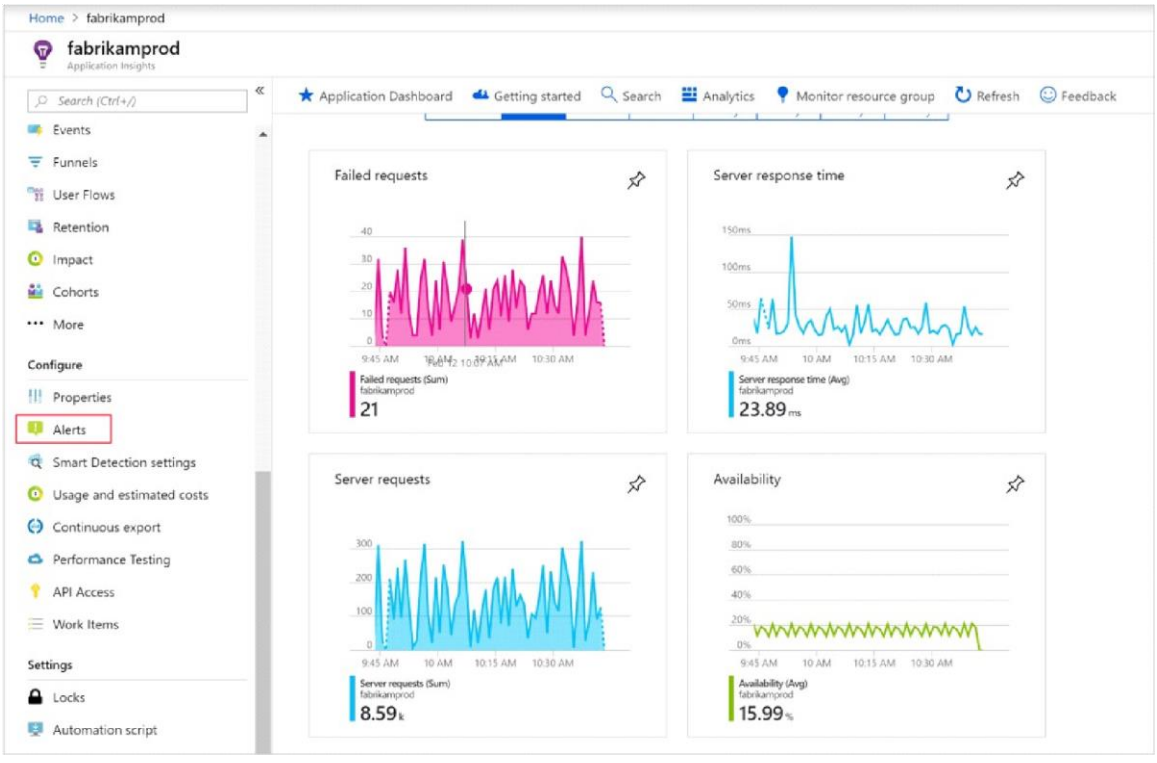


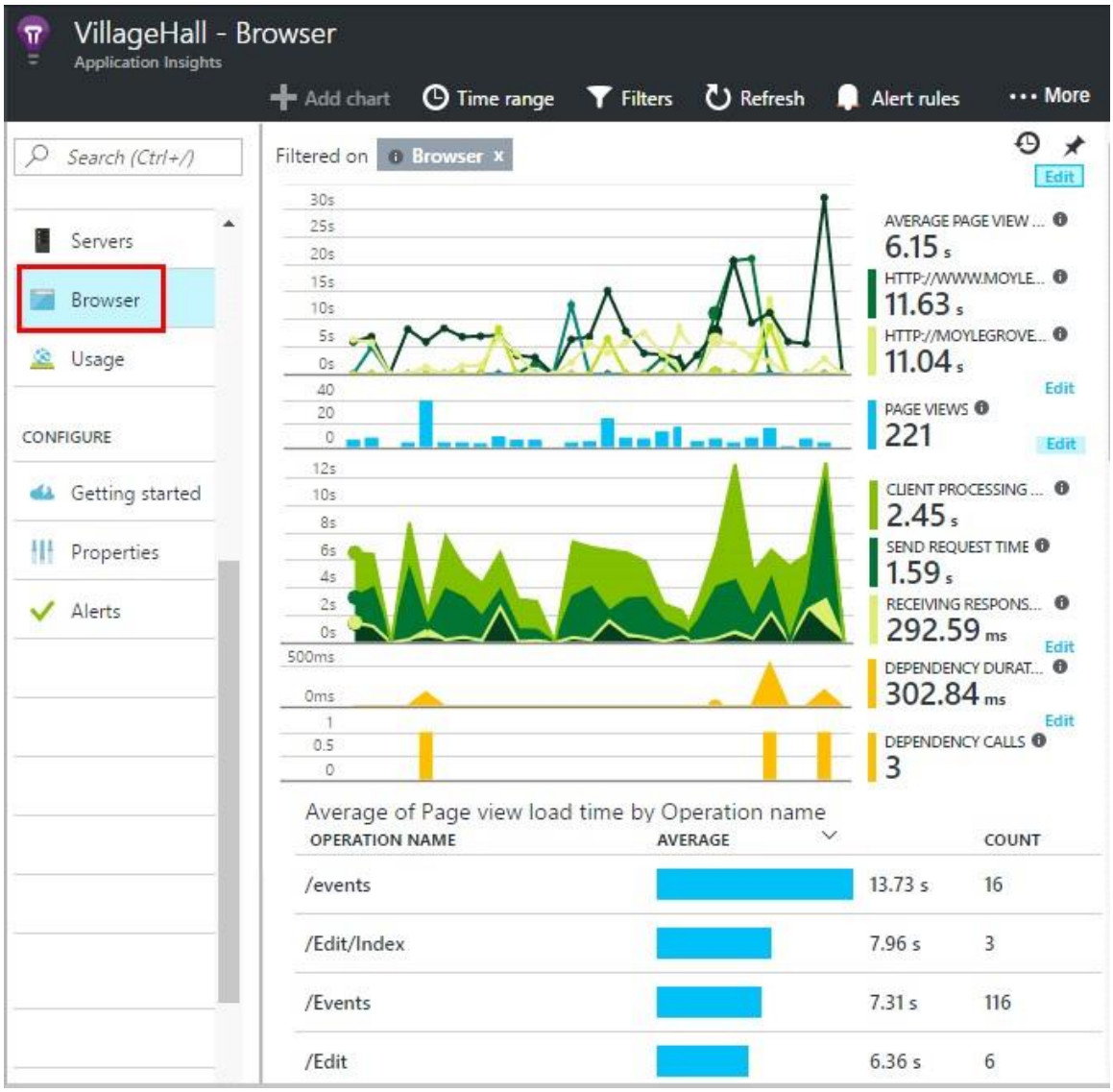


These two images are showing Deployment slots of Swap

4. Configure Application Insights for the project


MiniProject20220708101111 - M
https://portal.azure.com/#@Swaminatham5436outlook.onmicrosoft.com/resource/subscriptions/3e70b4be-0622-4e47-8cfa-06d3...
Microsoft Azure
Search resources, services, and docs (G+/)
Swaminatham5436@ou...
DEFAULT DIRECTORY
Home > MiniProject20220708101111
MiniProject20220708101111 | Application Insights
App Service
Search (Ctrl+/)
Diagnose and solve problems
Security
Events (preview)
Deployment
Quickstart
Deployment slots
Deployment Center
Settings
Configuration
Authentication
Application Insights
Identity
Backups
Custom domains
Application Insights
Collect application monitoring data using Application Insights
Enable
Disable
Feedback
Link to an Application Insights resource
Instrumentation key will be added to App Settings. This will overwrite any instrumentation key value in web app configuration files.
Your app will be connected to an auto-created Application Insights resource: MiniProject20220708101111
As part of using Application Insights instrumentation, we collect and send diagnostic data to Microsoft. This data helps us run and improve Application Insights. You have the option to disable non-essential data collection. Learn more
Change your resource
Create new resource
Apply
Type here to search
23°C Mostly cloudy
10:25 AM
7/8/2022


Home > fabrikamprod
fabrikamprod
Application Insights
Search (Ctrl+/)
Events
Funnels
User Flows
Retention
Impact
Cohorts
More
Configure
Properties
Alerts
Smart Detection settings
Usage and estimated costs
Continuous export
Performance Testing
API Access
Work Items
Settings
Locks
Automation script
Application Dashboard
Getting started
Search
Analytics
Monitor resource group
Refresh
Feedback
Failed requests
Failed requests (Sum)
fabrikamprod
21
Server response time
Server response time (Avg)
fabrikamprod
23.89 ms
Server requests
Server requests (Sum)
fabrikamprod
8.59 k
Availability
Availability (Avg)
fabrikamprod
15.99 %



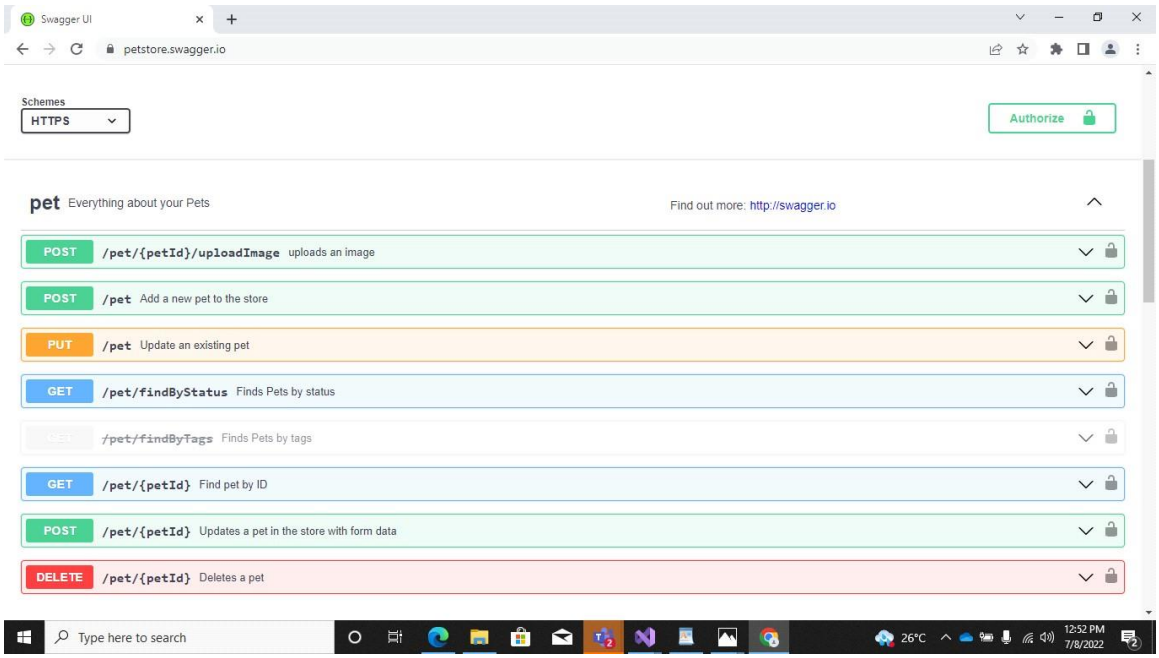
5. Configure Swagger for the API

Swagger UI allows anyone be it your development team or your end consumers to visualize and interact with the API's resources without having any of the implementation logic in place. It's automatically generated from your Open API (formerly known as Swagger) Specification, with the visual documentation making it easy for backend implementation and client-side consumption.

Advantages:

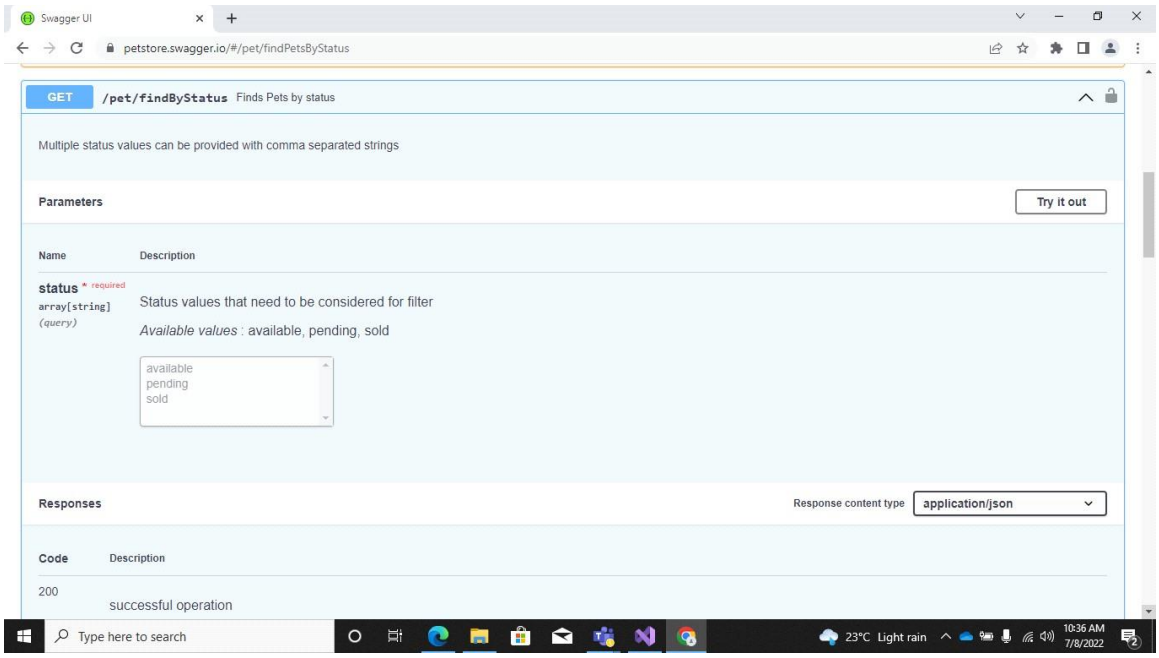
- Dependency Free -The UI works in any development environment, be it locally or in the web.

- Human Friendly -Allow end developers to effortlessly interact and try out every single operation your API exposes for easy consumption.
- Easy to Navigate -Quickly find and work with resources and endpoints with neatly categorized documentation.
- All Browser Support -Cater to every possible scenario with Swagger UI working in all major browsers. ·Fully Customizable -Style and tweak your Swagger UI the way you want with full source code access.
- Complete OAS Support-Visualize APIs defined in Swagger 2.0 or OAS 3.0



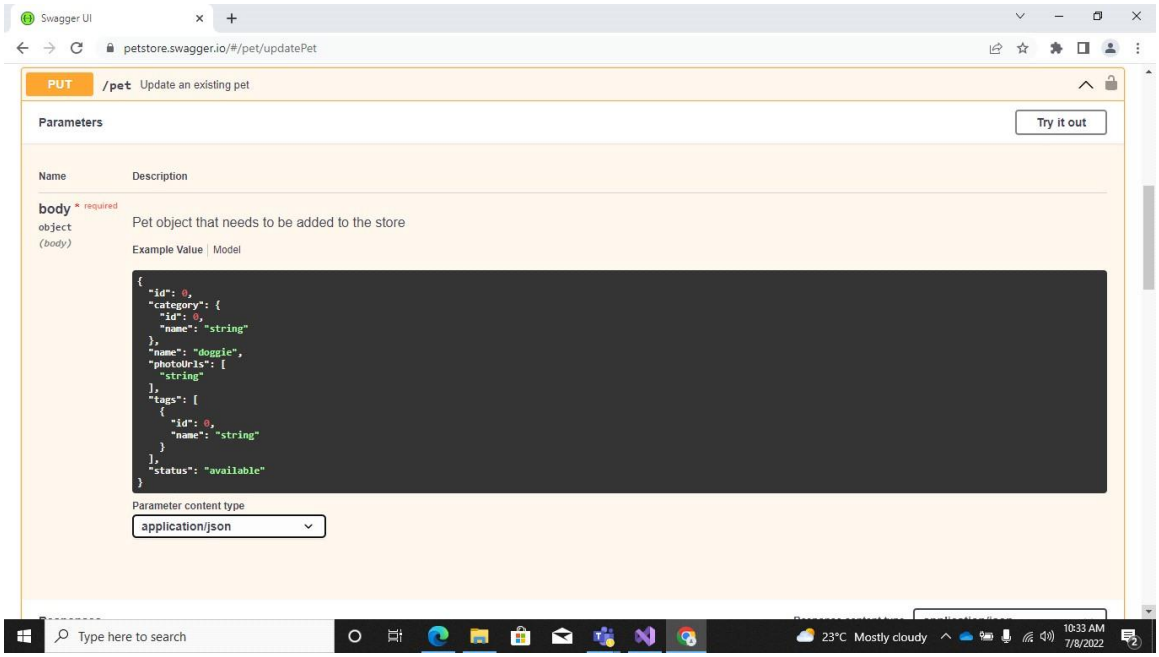
This image is showing swagger documentation for the created Products Web API project.

GET:



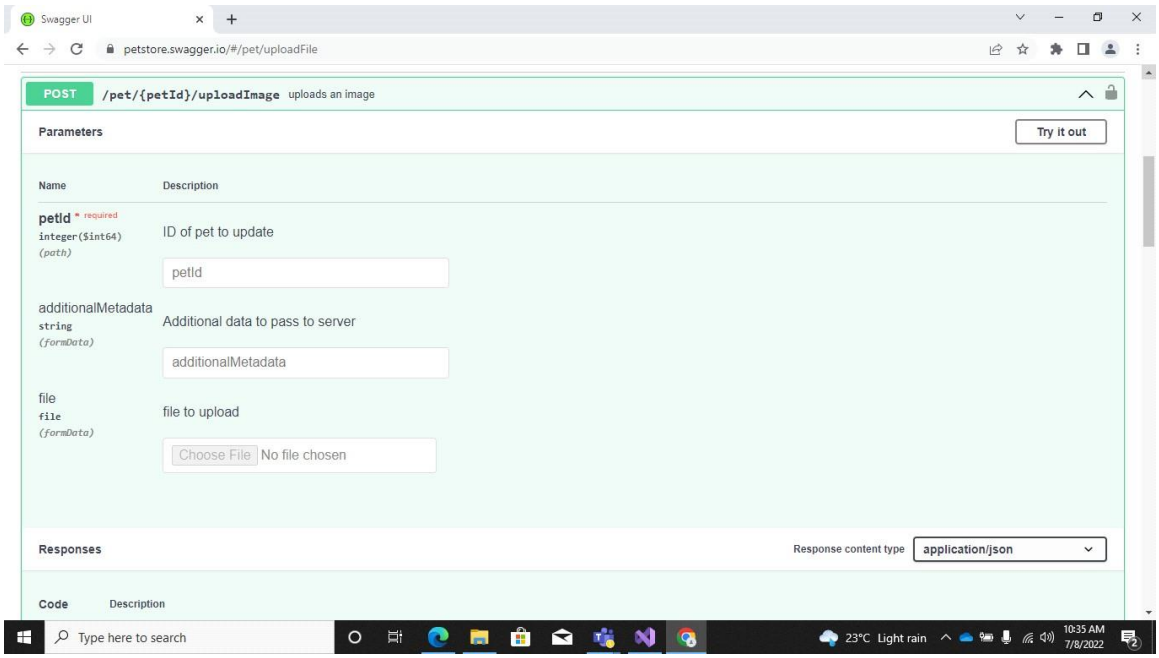
This Image showing the get call for a product with specified product id passed as a path variable

PUT:



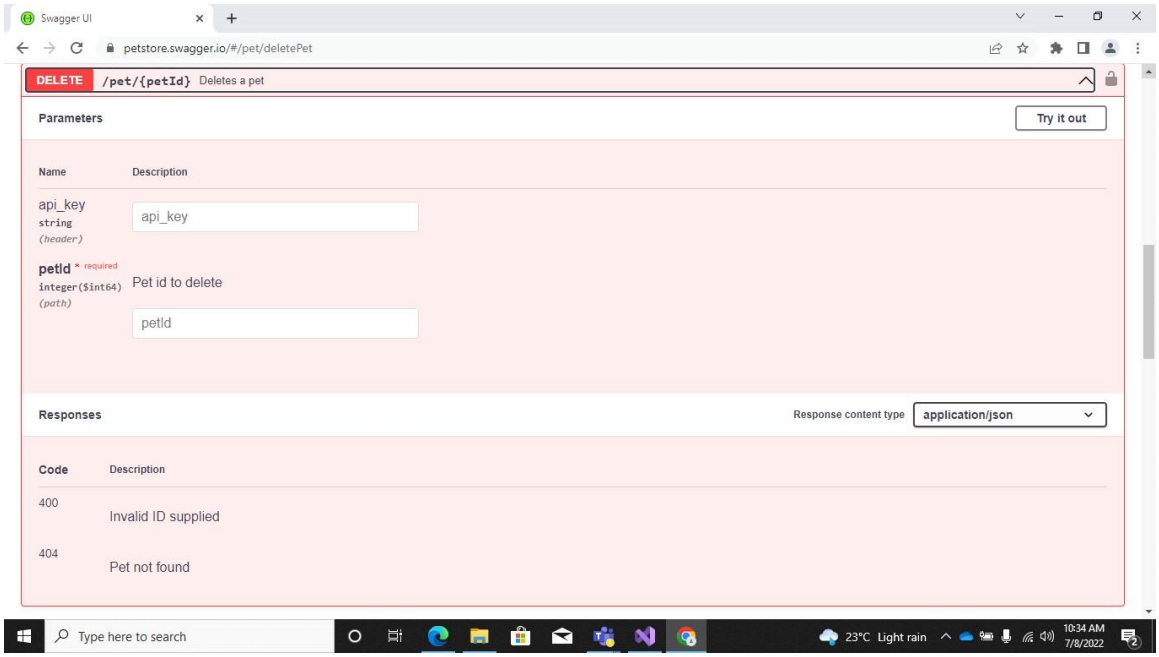
This Image showing the put call for updating an existing product with request body (fields to be updated) and path variable as product id.

POST:



This Image showing the post call for creating a new product.

DELETE:



This Image showing the delete call for deleting an existing product with the specified id passed as a path variable.

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 analyse their results. You can use Log Analytics queries to retrieve records that match particular criteria, identify trends, analyse patterns, and provide various insights into your data.