**WEB API PROJECT**

**1.** Create WEB API Project.

**2.** Add models folder and in that folder add Product.cs Class.

**3.** Add Controller for Product class Using Entity Frame Work.

**4.** Open package manager Consoler and follow the below steps.

**4.1.** Add-Migration ApiCreate.

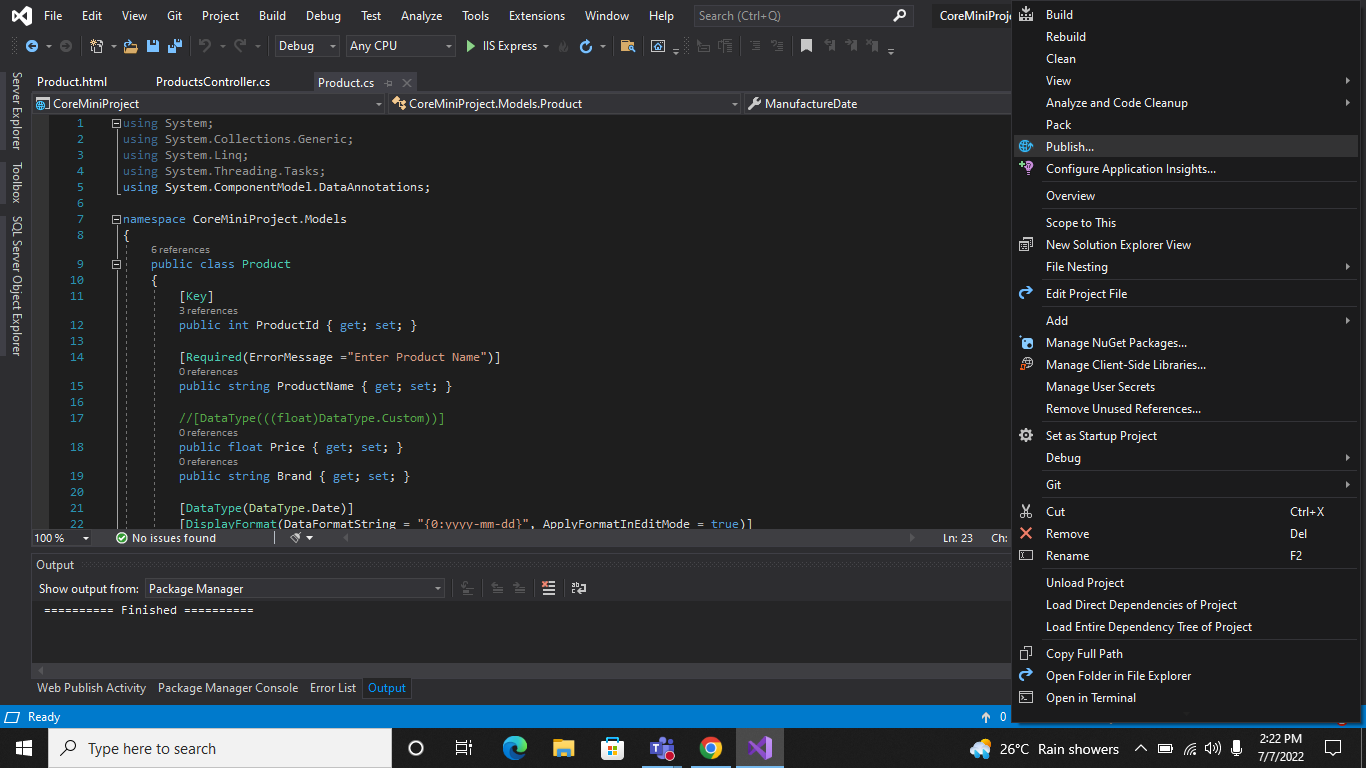
**4.2.** Update-Database.

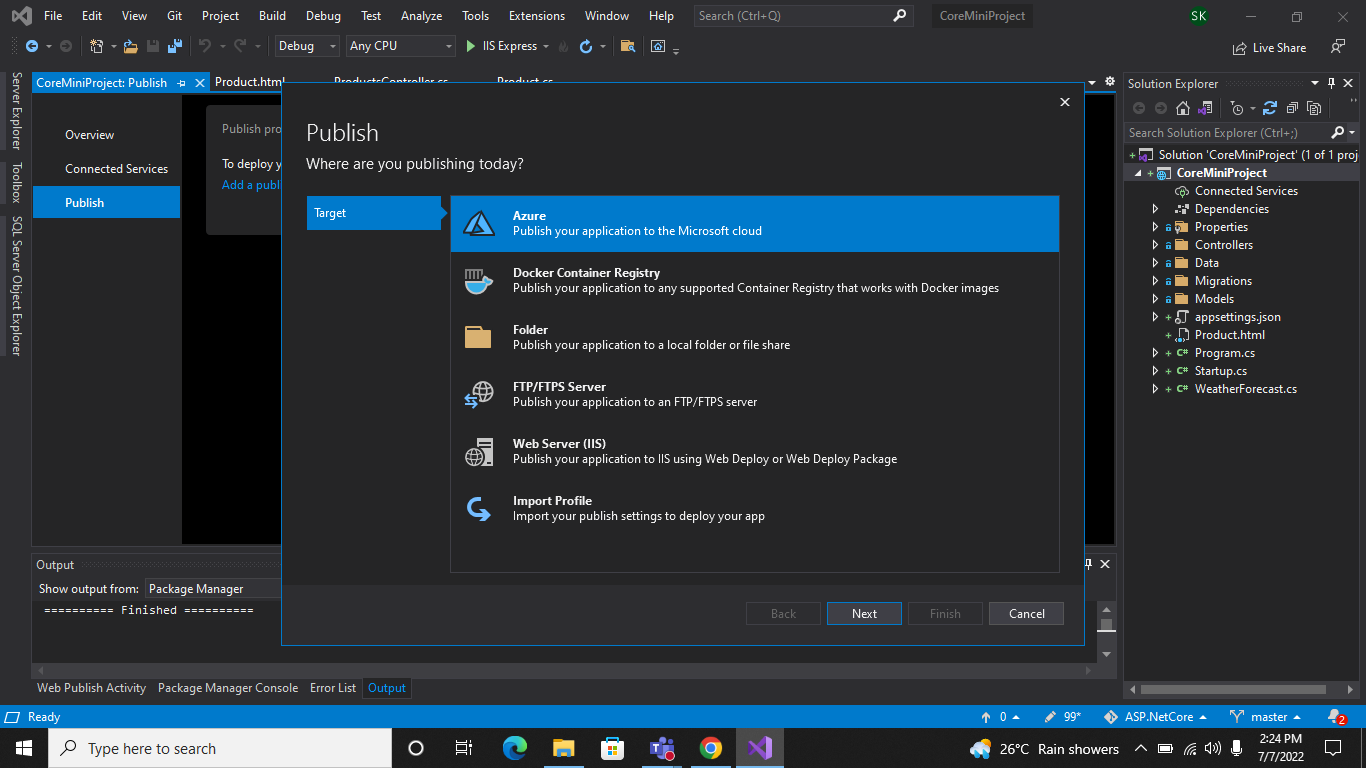
**5.** Add Web Api Client html page in the Project Using Ajax and JQuery.

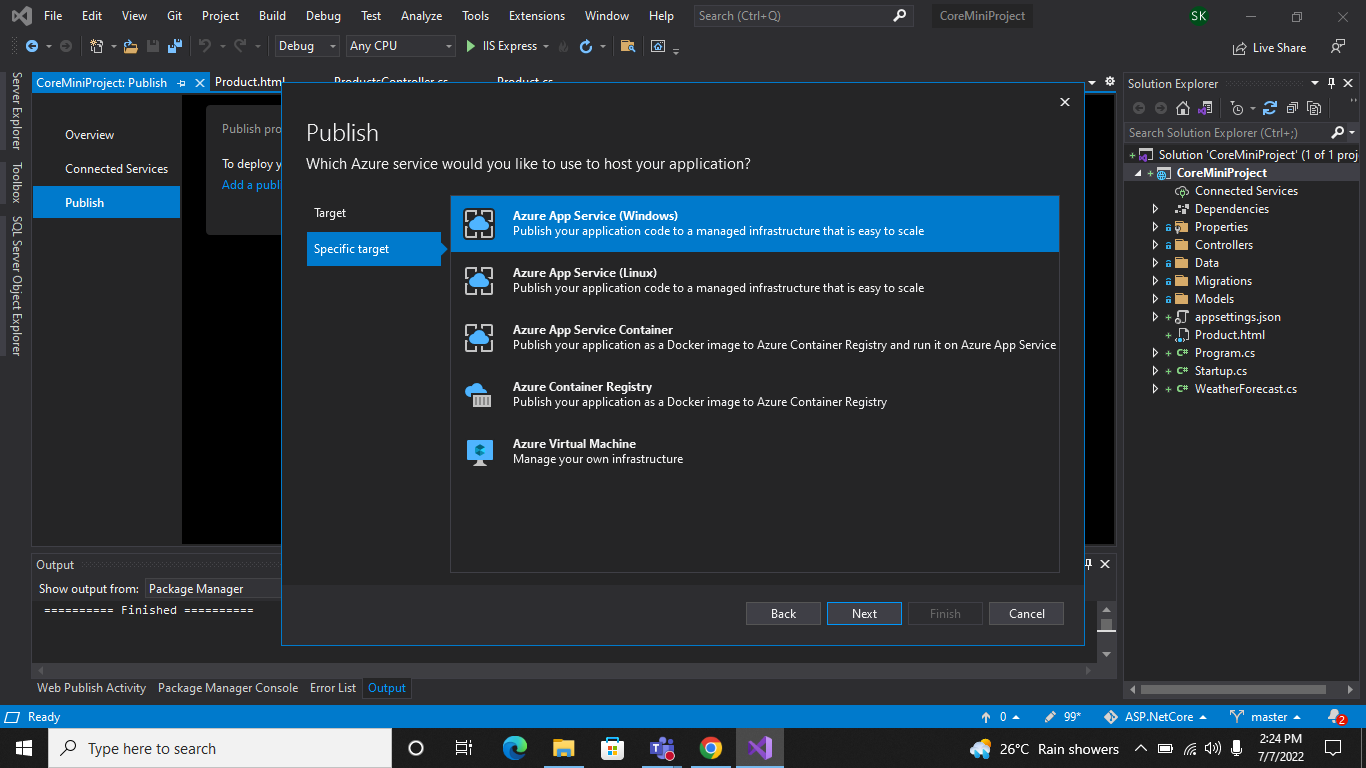
**6.** Finally Hosting to the Azure. Follow the bellow Pics.

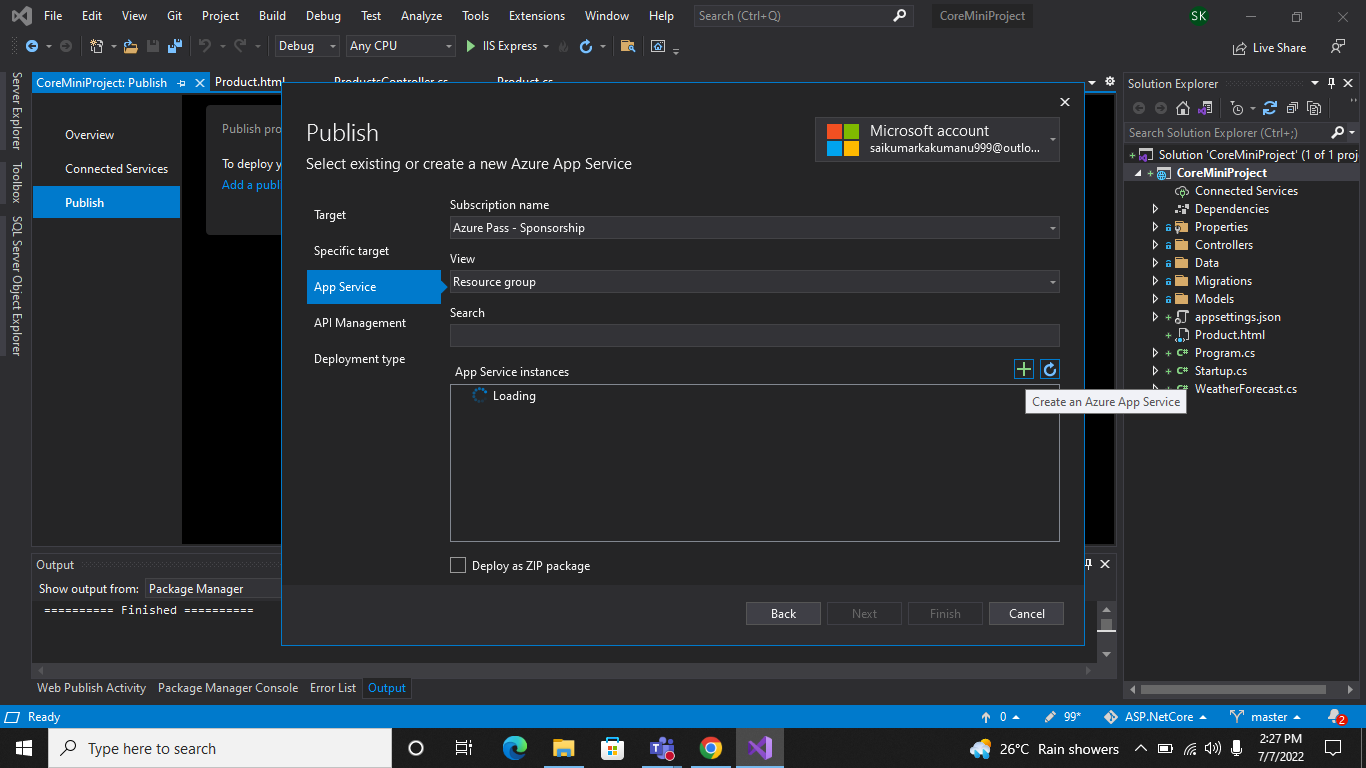
**Azure Hosting :-**

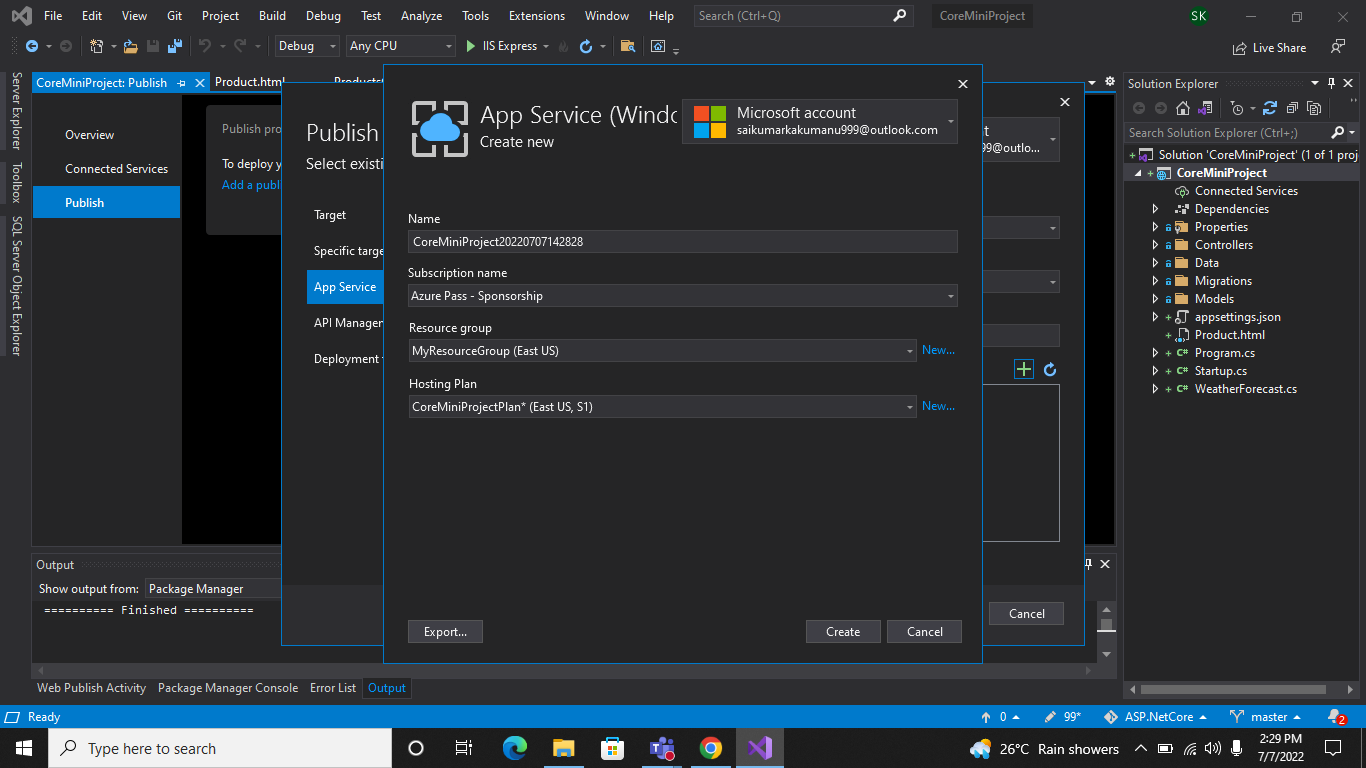
1. **Host the web api in azure and consume the same using JQuery Client :-**

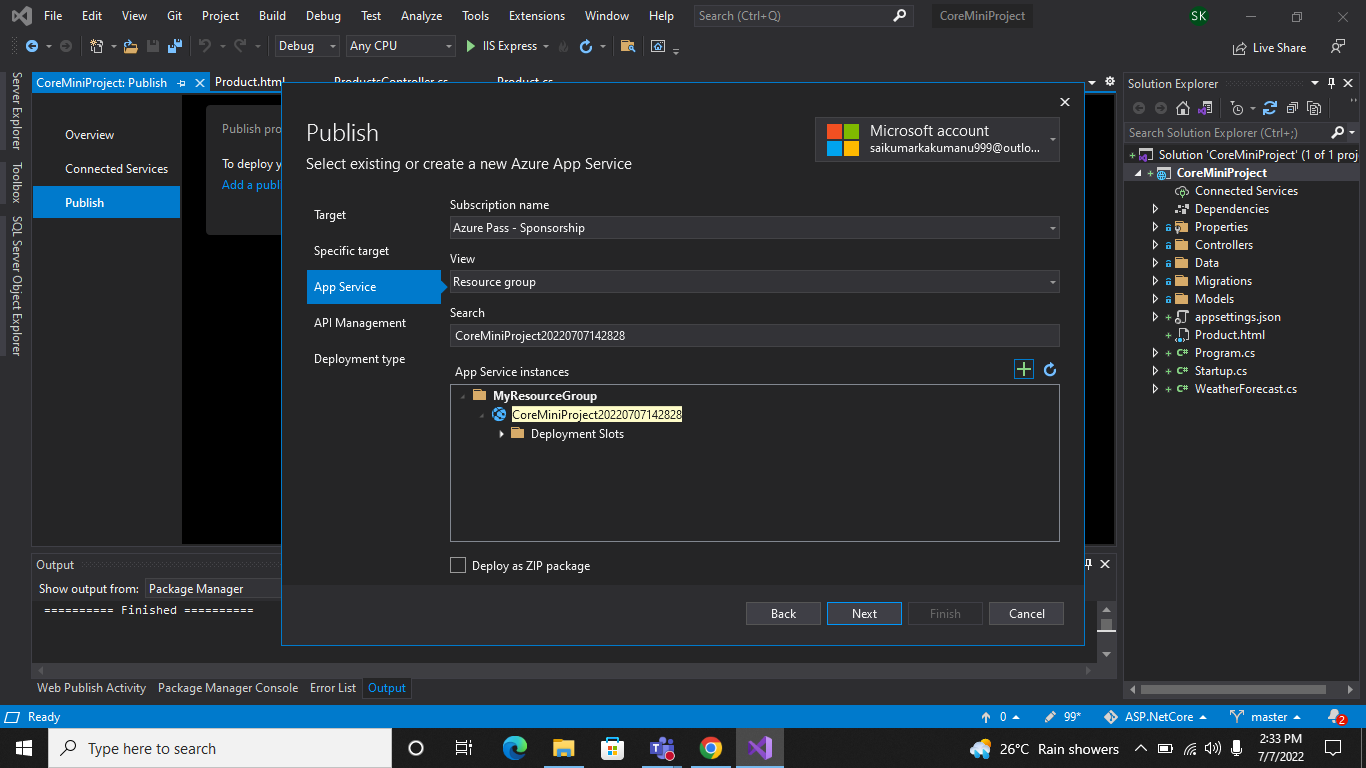


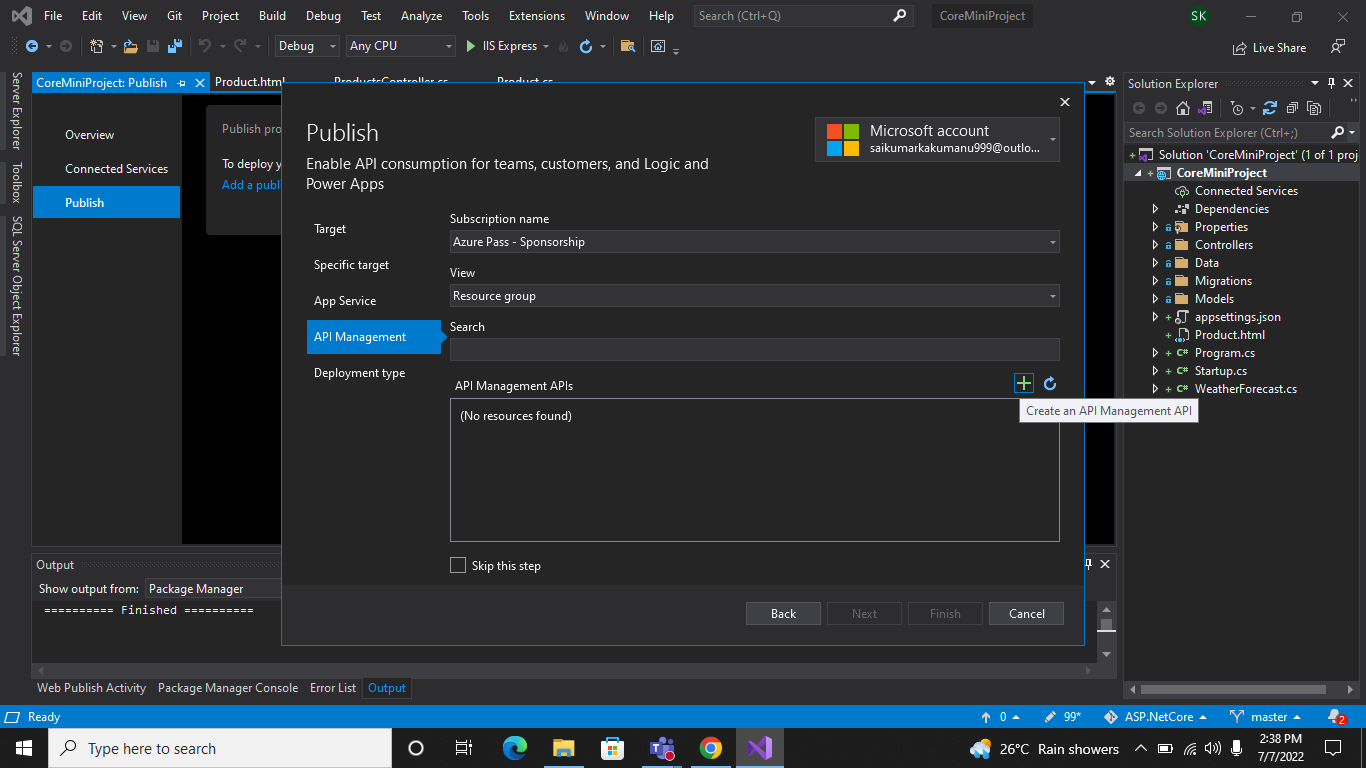


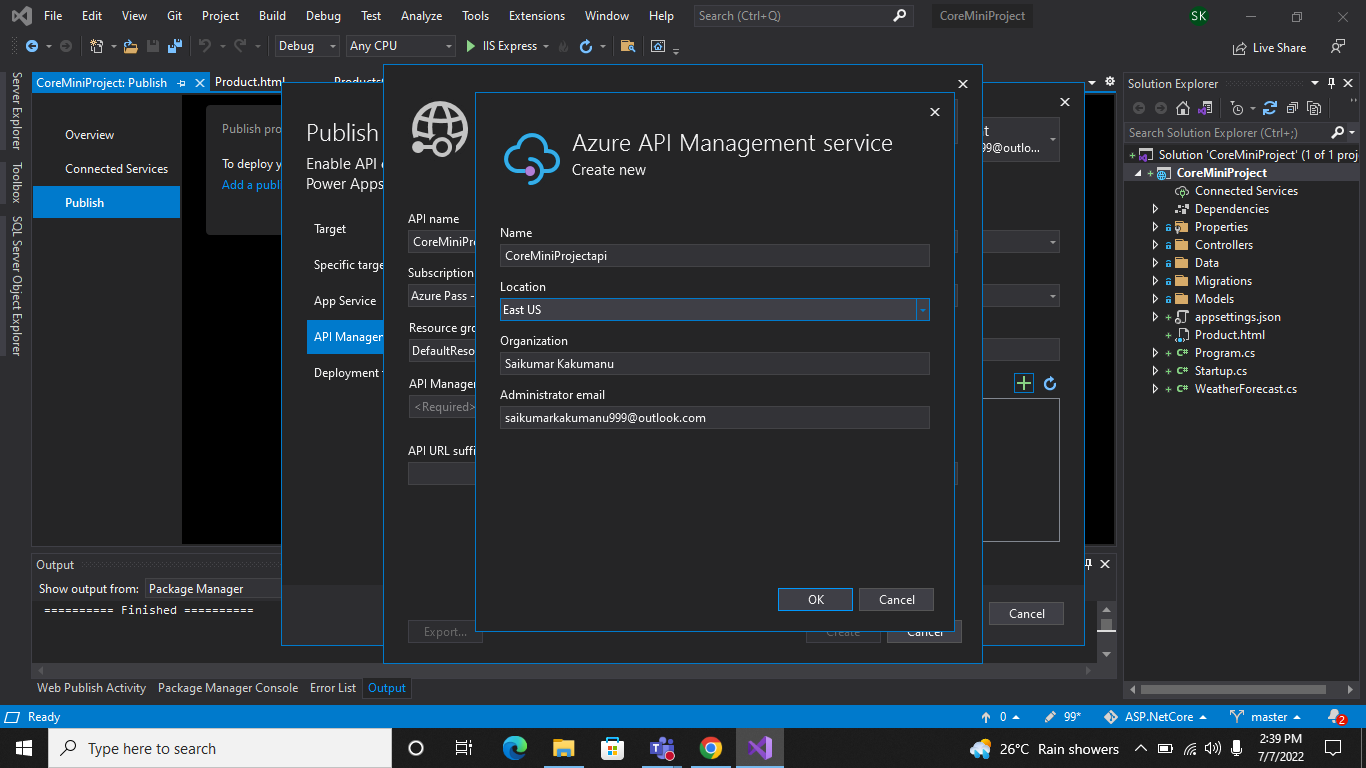


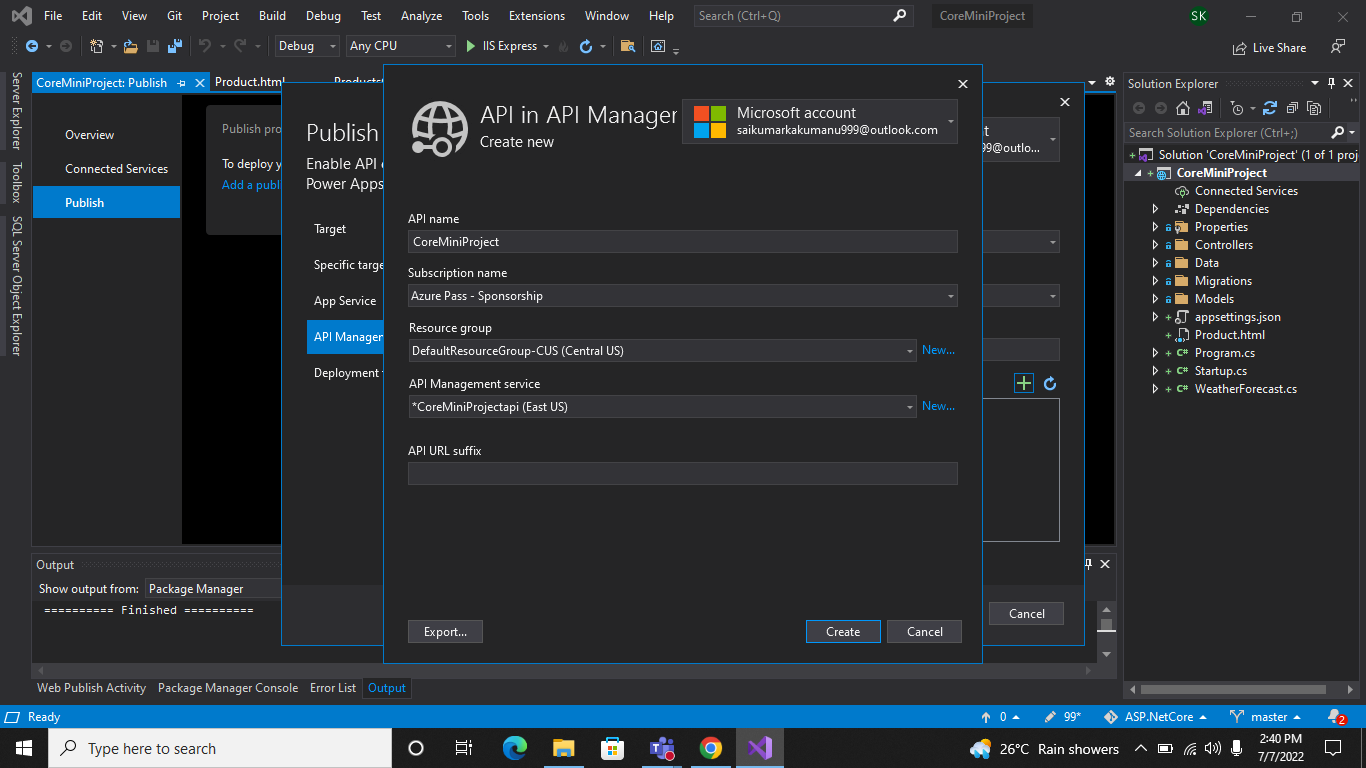


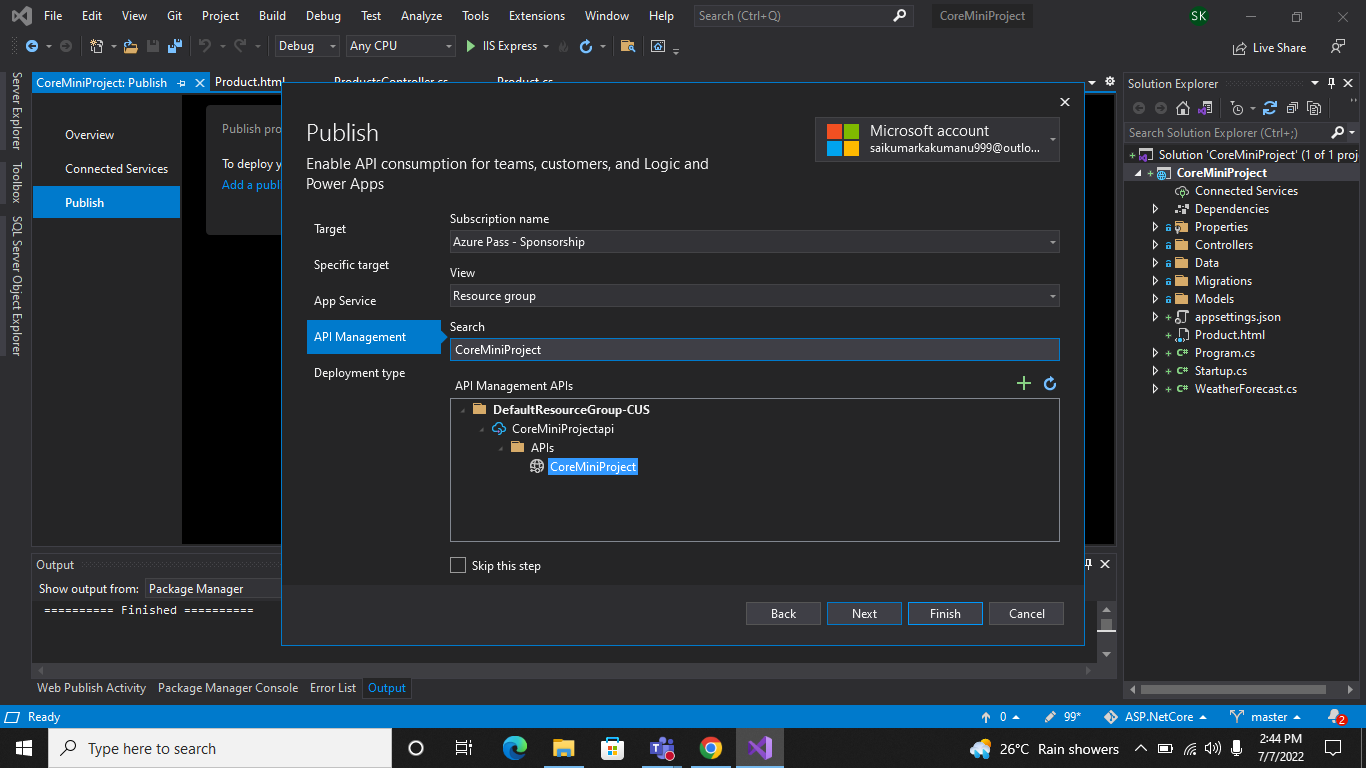


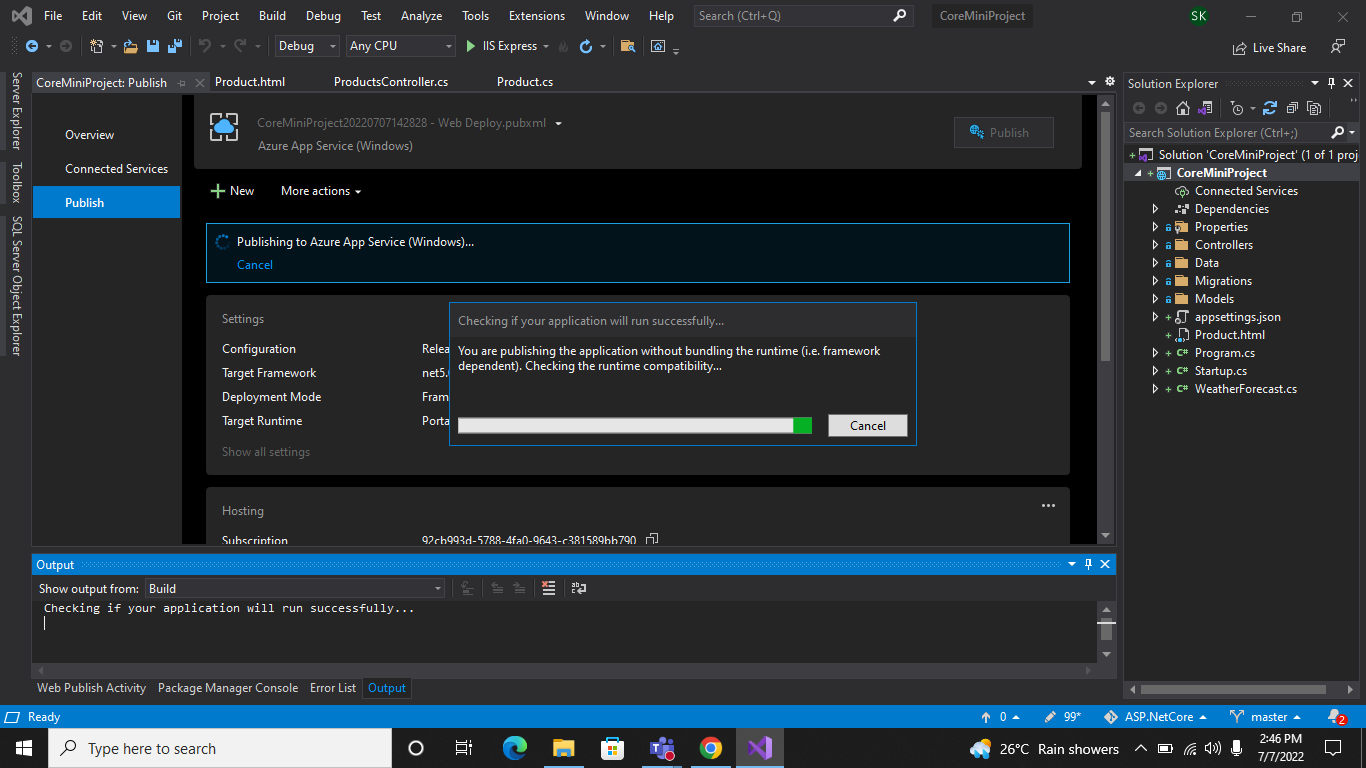












**2.Configure Scale out by adding rules for custom scaling :-**

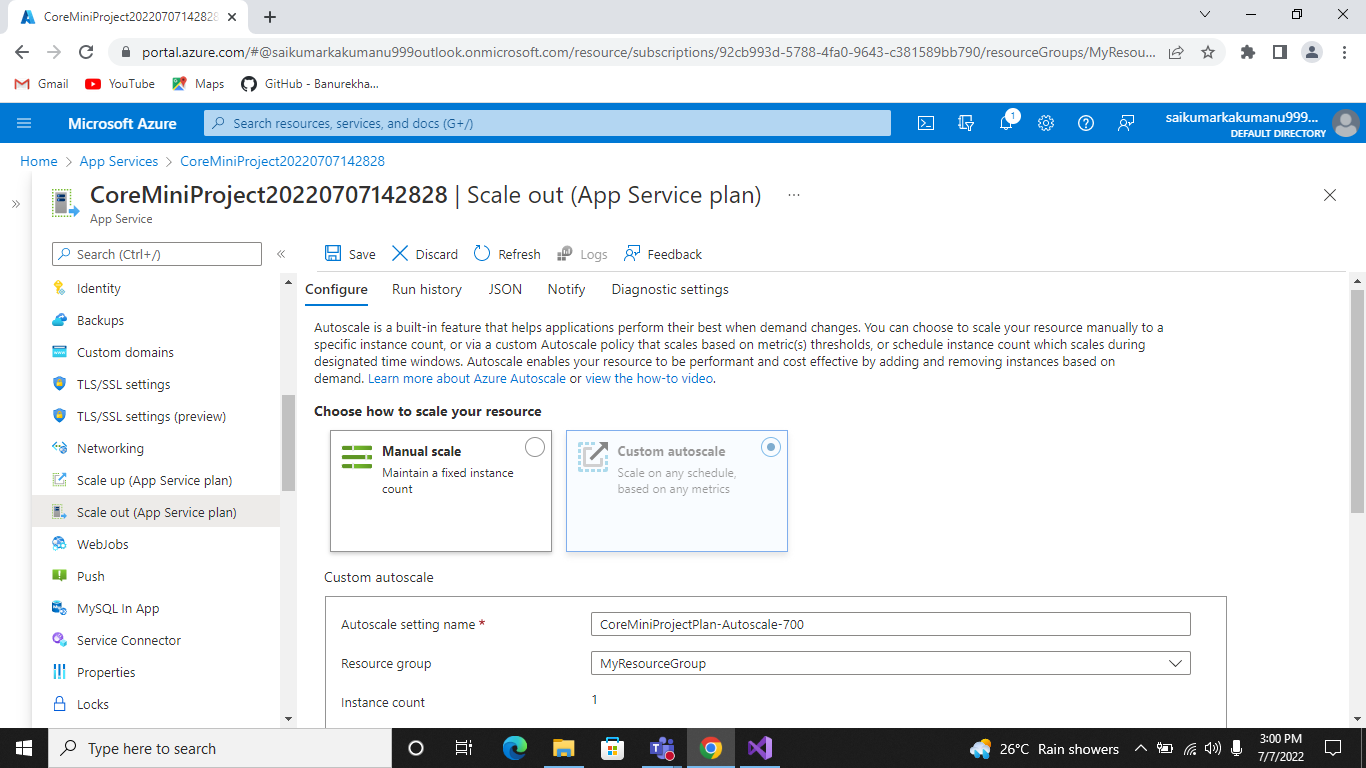
A scale out operation is the equivalent of creating multiple copies of your web site and adding a load balancer to distribute the demand between them. When you scale out a web site in Windows Azure Web Sites there is no need to configure load balancing separately since this is already provided by the platform.

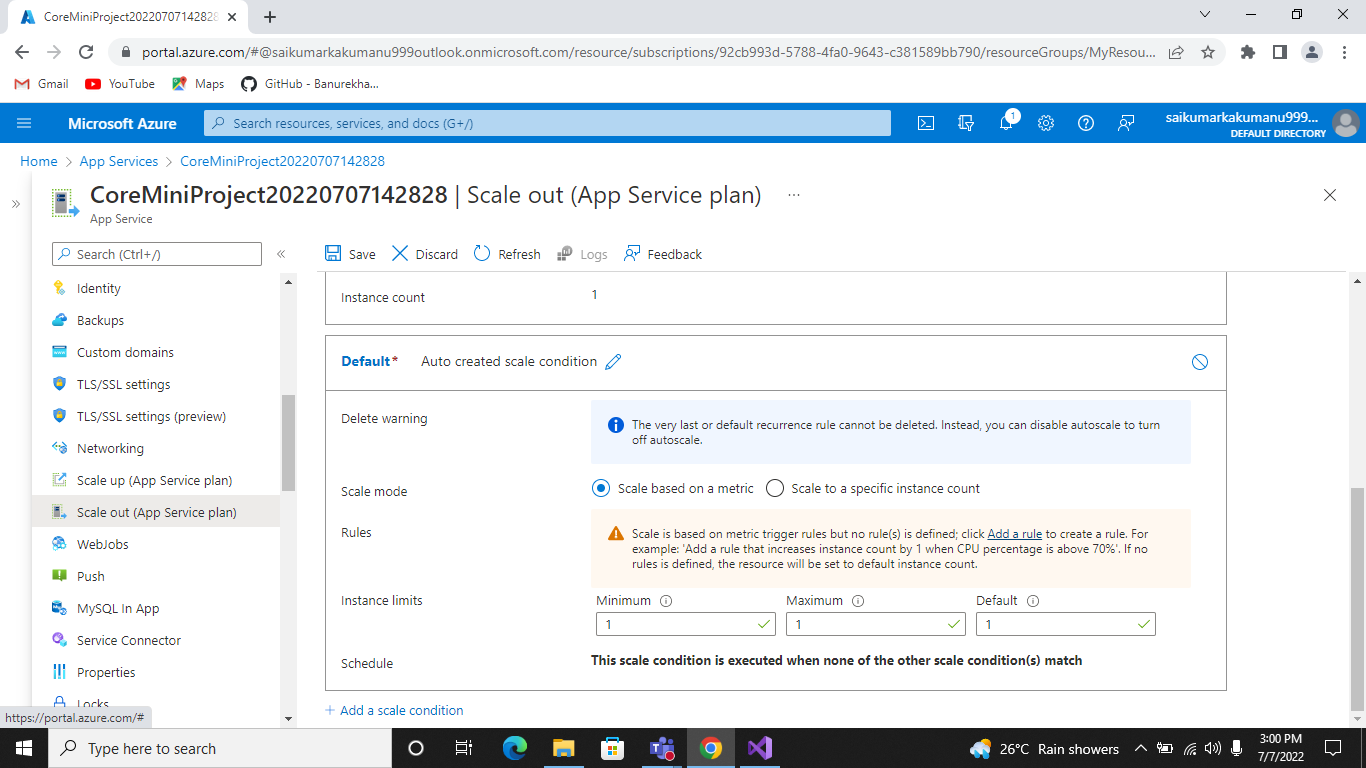
Follow the Bellow images to add the scale out to our web site.

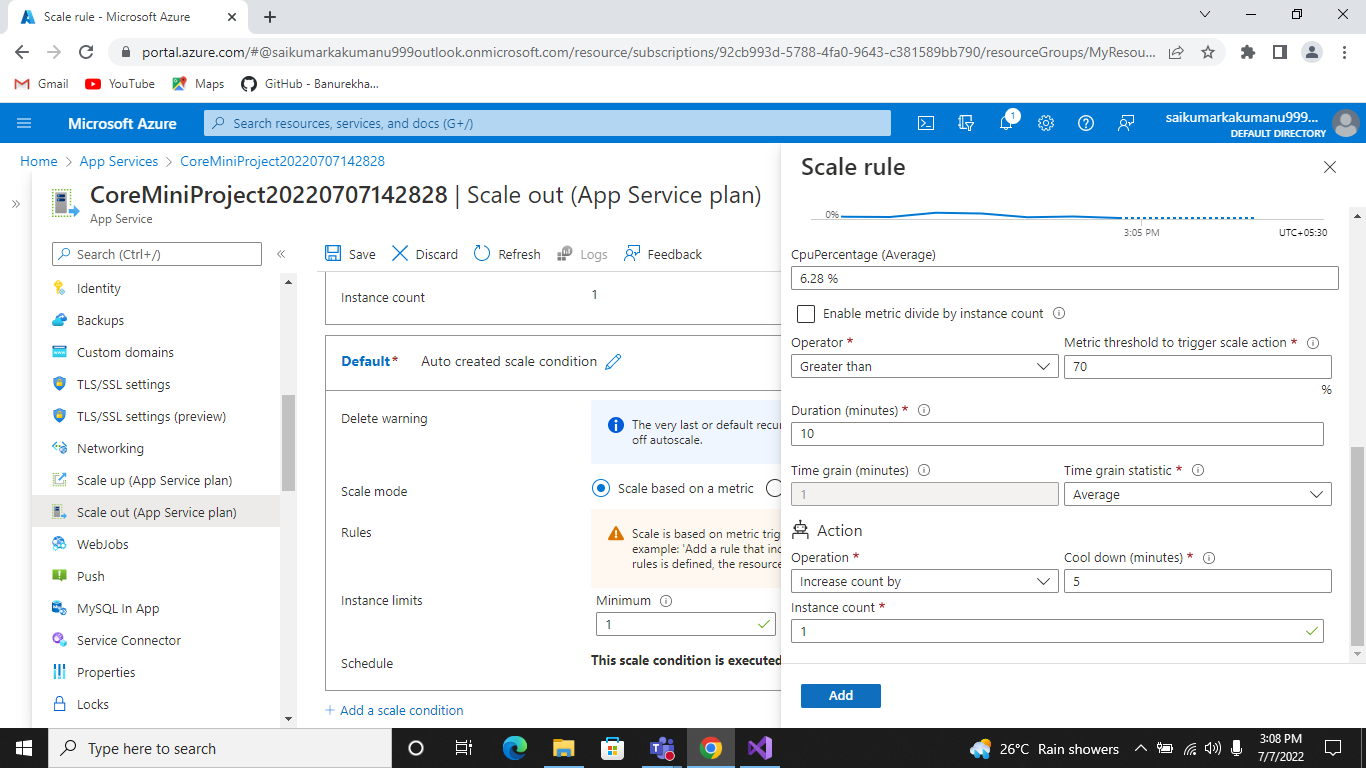
1.select Custom Auto Scale.

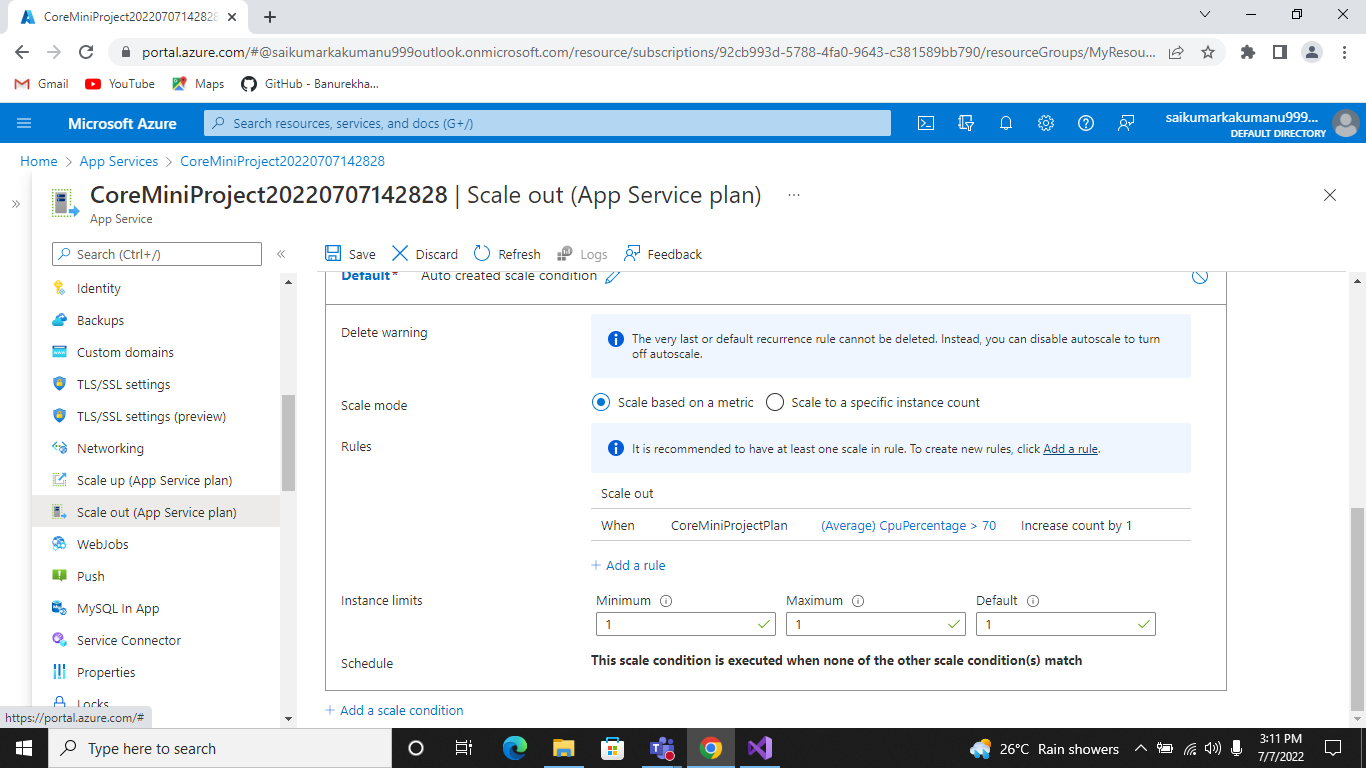
2.Click on Add a rule.

3.Add your rules and click on Add button and Save changes.









**3.Configure Deployment Slots for Staging and Production :-**

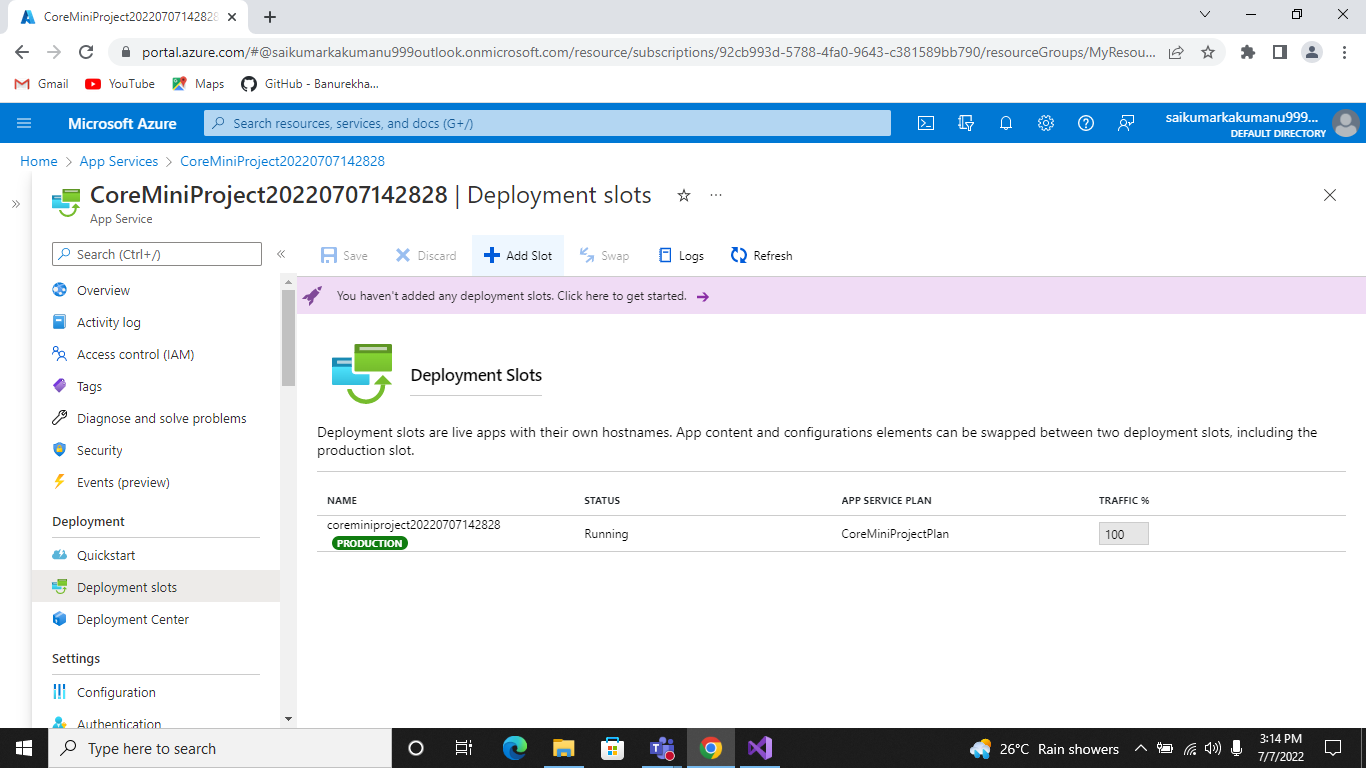
Azure Functions deployment slots allow your function app to run different instances called "slots". Slots are different environments exposed via a publicly available endpoint. One app instance is always mapped to the production slot, and you can swap instances assigned to a slot on demand. Function apps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

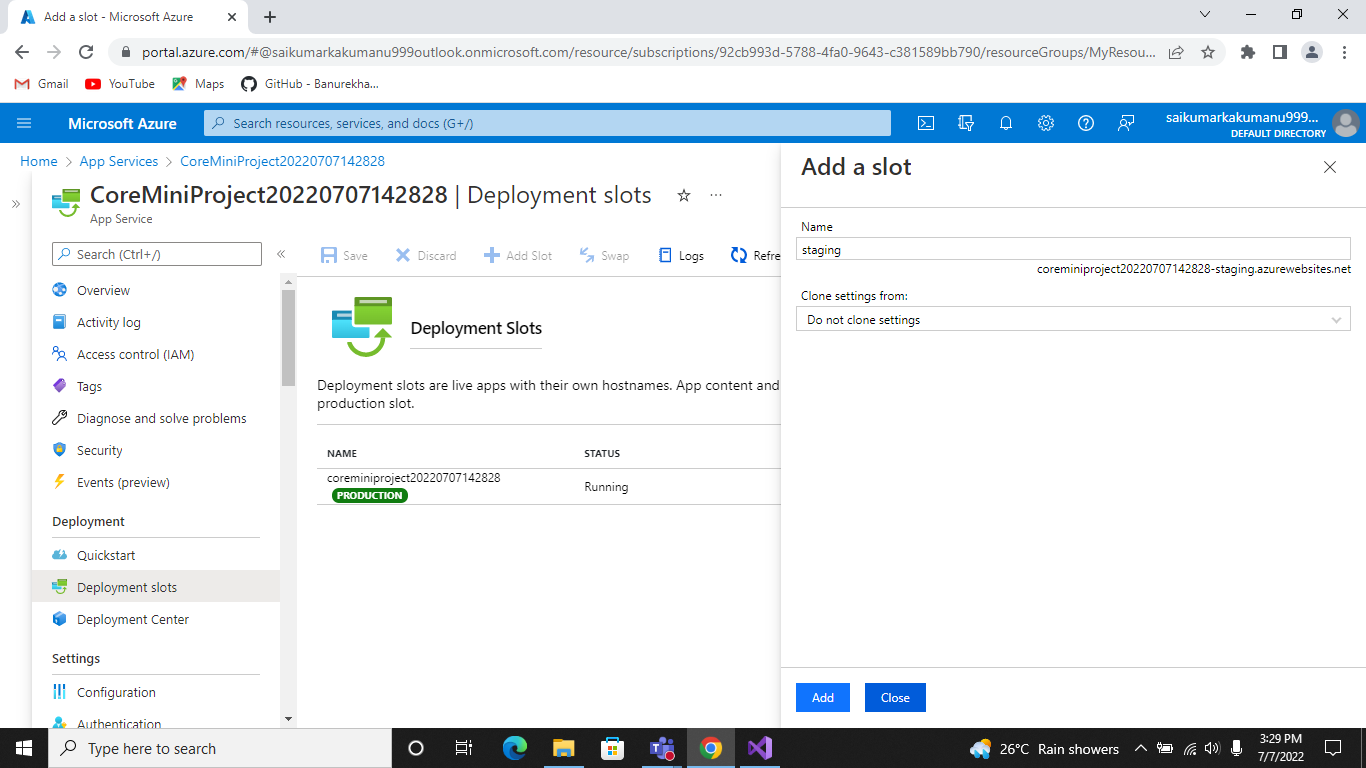
The following reflect how functions are affected by swapping slots:

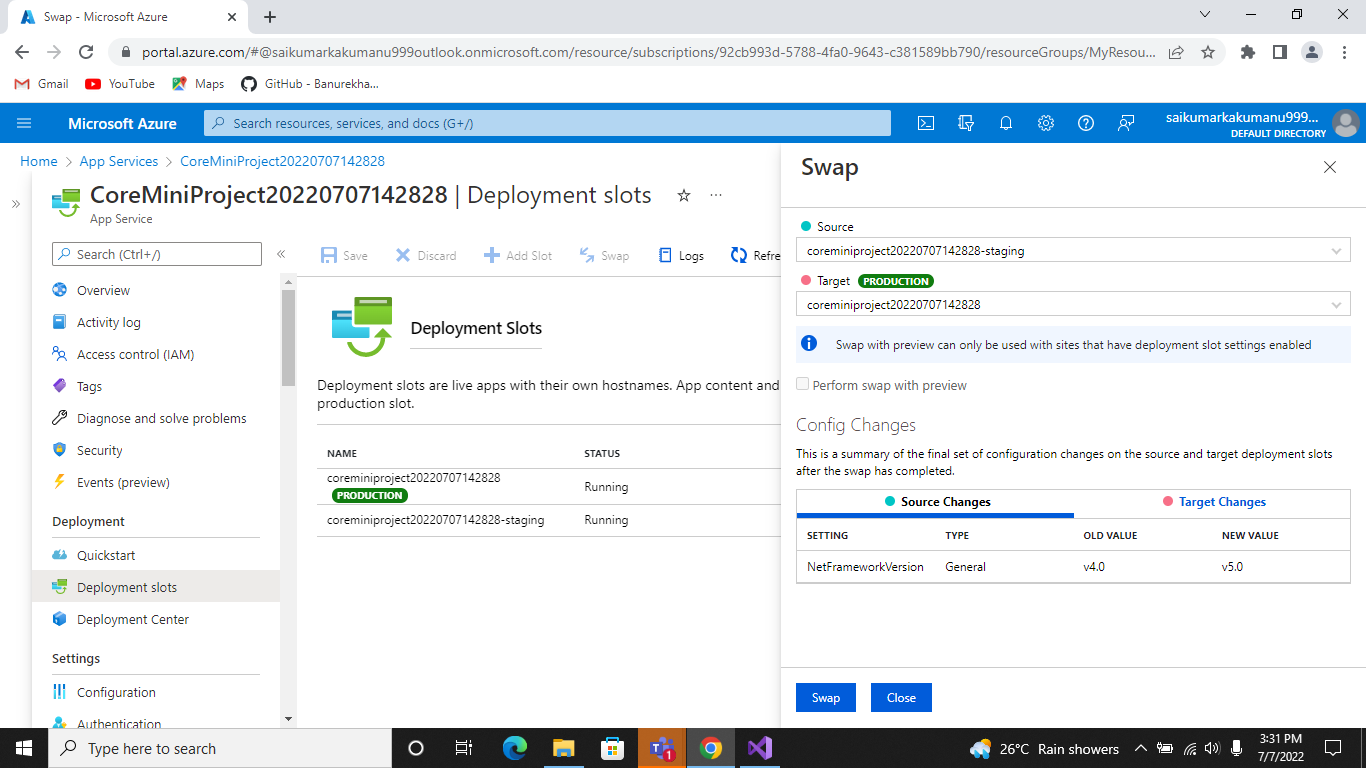
* Traffic redirection is seamless; no requests are dropped because of a swap. This seamless behavior is a result of the next function triggers being routed to the swapped slot.
* Currently executing function are terminated during the swap. Please review [Improve the performance and reliability of Azure Functions](https://docs.microsoft.com/en-us/azure/azure-functions/performance-reliability#write-functions-to-be-stateless) to learn how to write stateless and defensive functions.

Follow the bellow images to add deployment slots.

1. Click on Add Slot.
2. Enter The slot name and click on add button.
3. Finally swapping the slots.







**4.Configure Application Insights for the project :-**

Application Insights is a feature of [Azure Monitor](https://docs.microsoft.com/en-us/azure/azure-monitor/overview) that provides extensible application performance management (APM) and monitoring for live web apps. Developers and DevOps professionals can use Application Insights to:

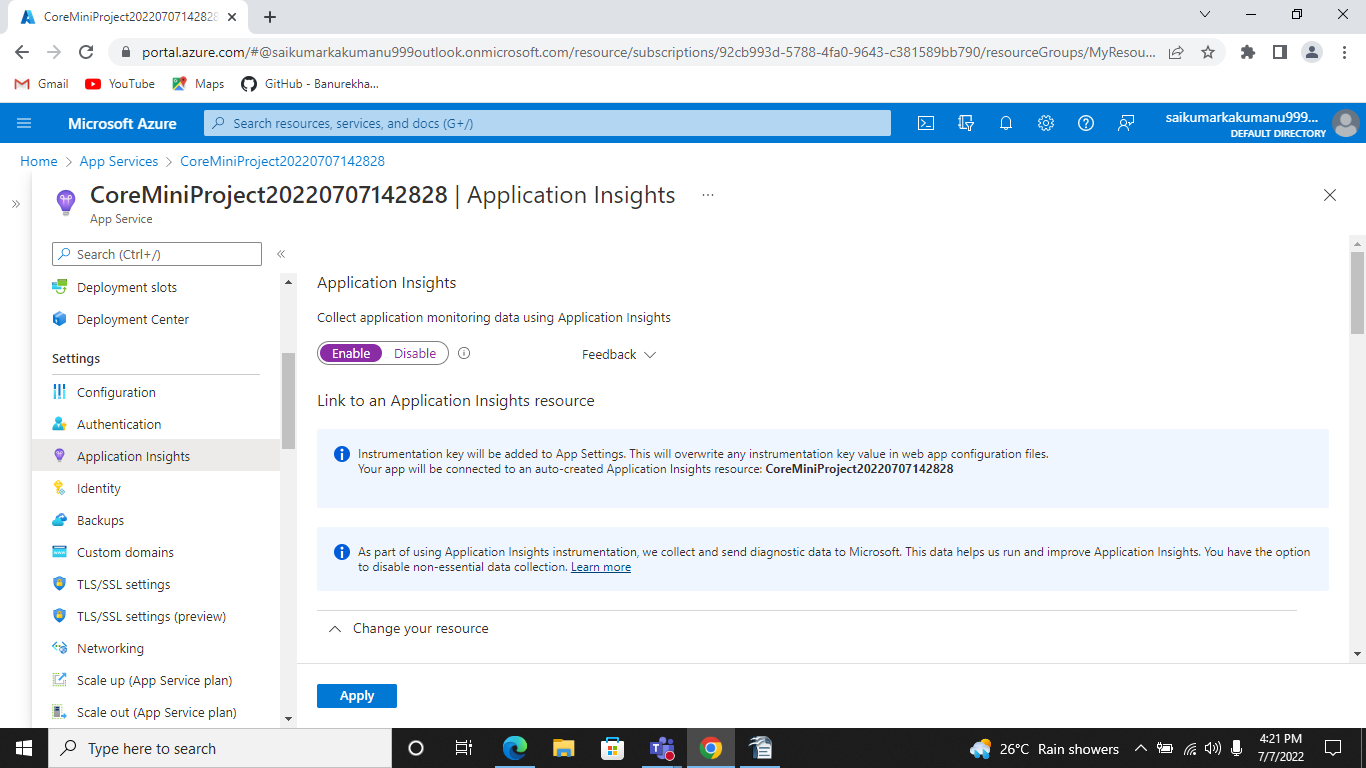
* Automatically detect performance anomalies.
* Help diagnose issues by using powerful analytics tools.
* See what users actually do with apps.
* Help continuously improve app performance and usability.

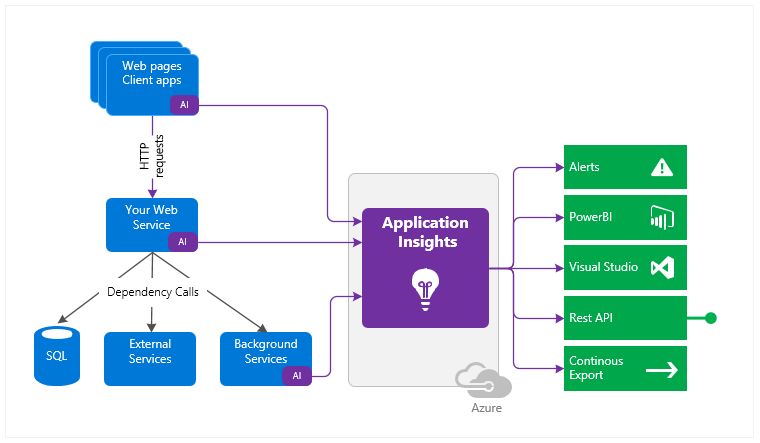
Application Insights:

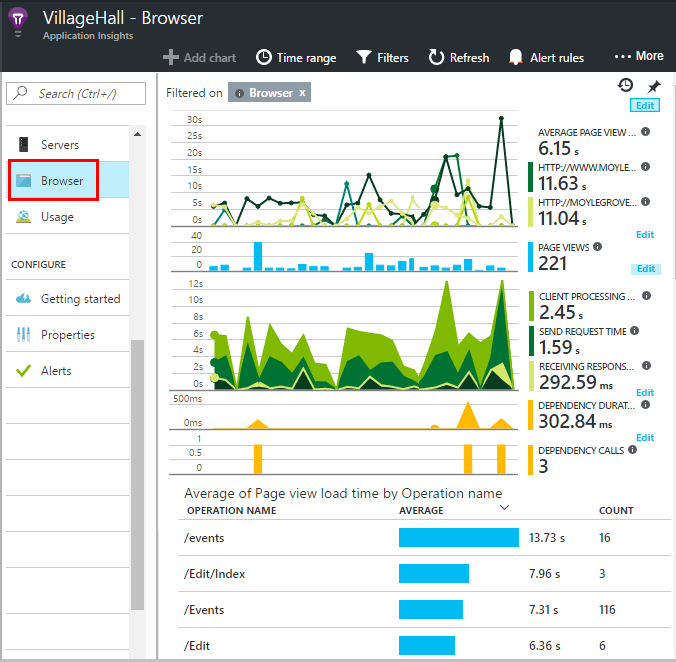
* Supports a wide variety of platforms, including .NET, Node.js, Java, and Python.
* Works for apps hosted on-premises, hybrid, or on any public cloud.
* Integrates with DevOps processes.
* Has connection points to many development tools.
* Can monitor and analyze telemetry from mobile apps by integrating with Visual Studio [App Center.](https://appcenter.ms/)

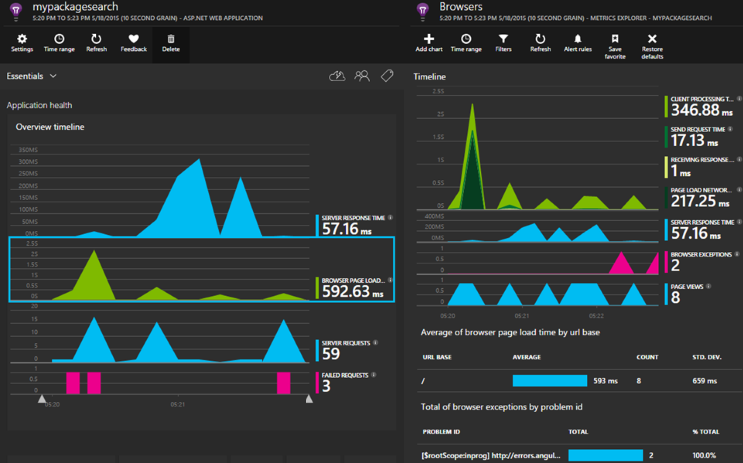
Follow the bellow images to add Application insights

1. Trun on Application Insights.
2. Select Existing Resource.
3. Click on the Apply button.









**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 OpenAPI (formerly known as Swagger) Specification, with the visual documentation making it easy for back end implementation and client side consumption.

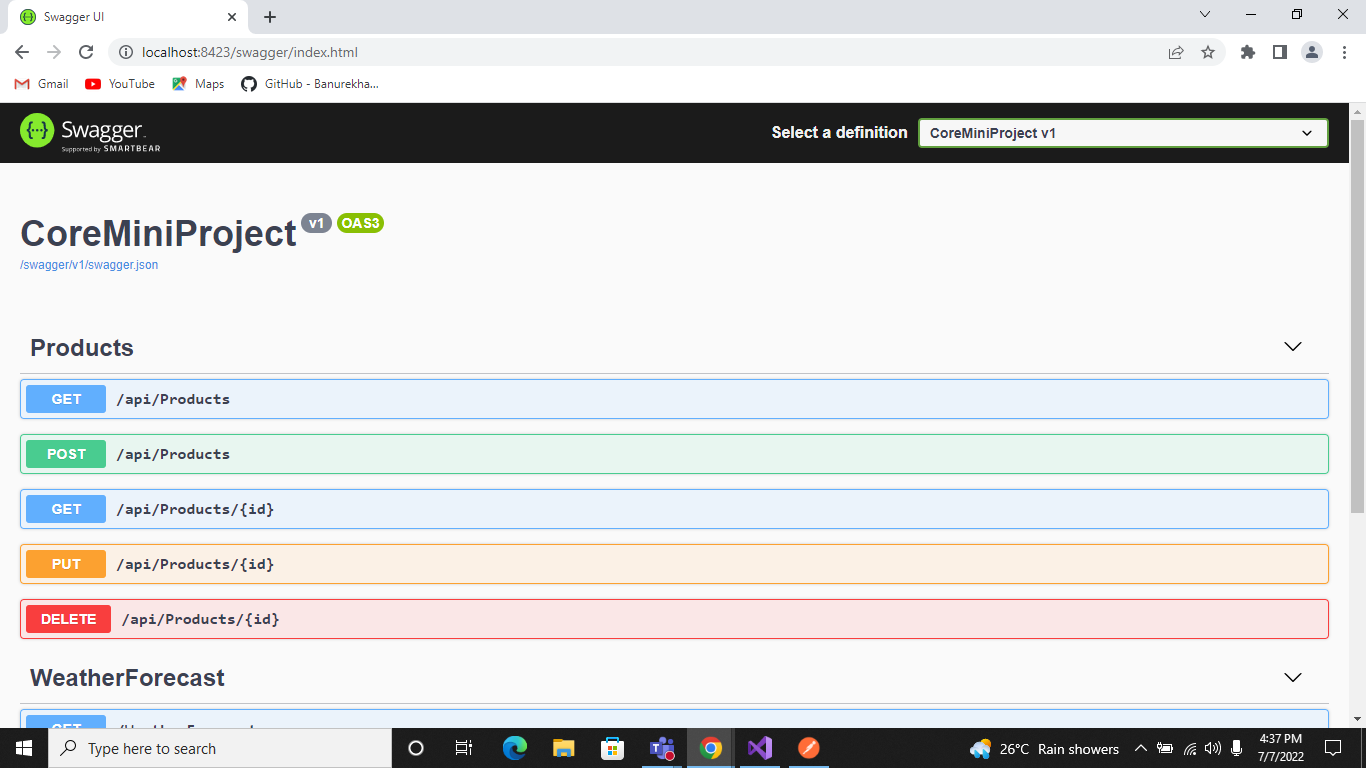
**Advantages of Swagger Api :-**

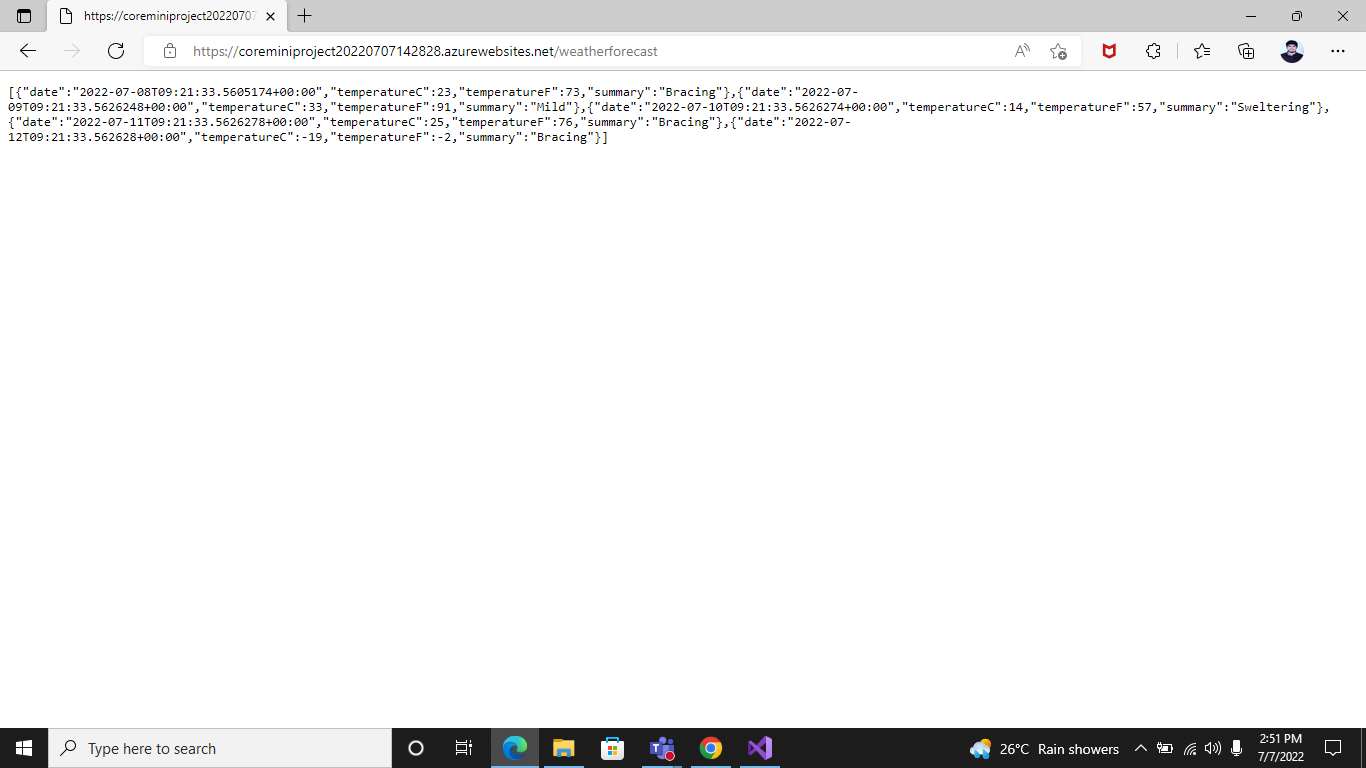
**1.** Testing is always crucial; on both the development and QA levels. Swagger provides a UI integrated page where all the API methods are listed and enables the user to test any method that is required from the UI.

**2.** Swagger does the documentation in a conventional way (OpenAPI) which means it is in a machine-readable language. If a user starts the documentation first, Swagger will write the structure of the API automatically based on the written documentation. The API logic relies on the developer and business requirements but the structure will be written by Swagger itself.

**3.**The user does not need a separate applications to test APIs. Just configure Swagger once in the project and access it through a URL to test the APIs.

These are the major benefits users will see once they start using Swagger. Swagger provides immense support for a wide range of platforms, languages, and domains.





**6.Work with Log Analytics with the sample logs available :-**

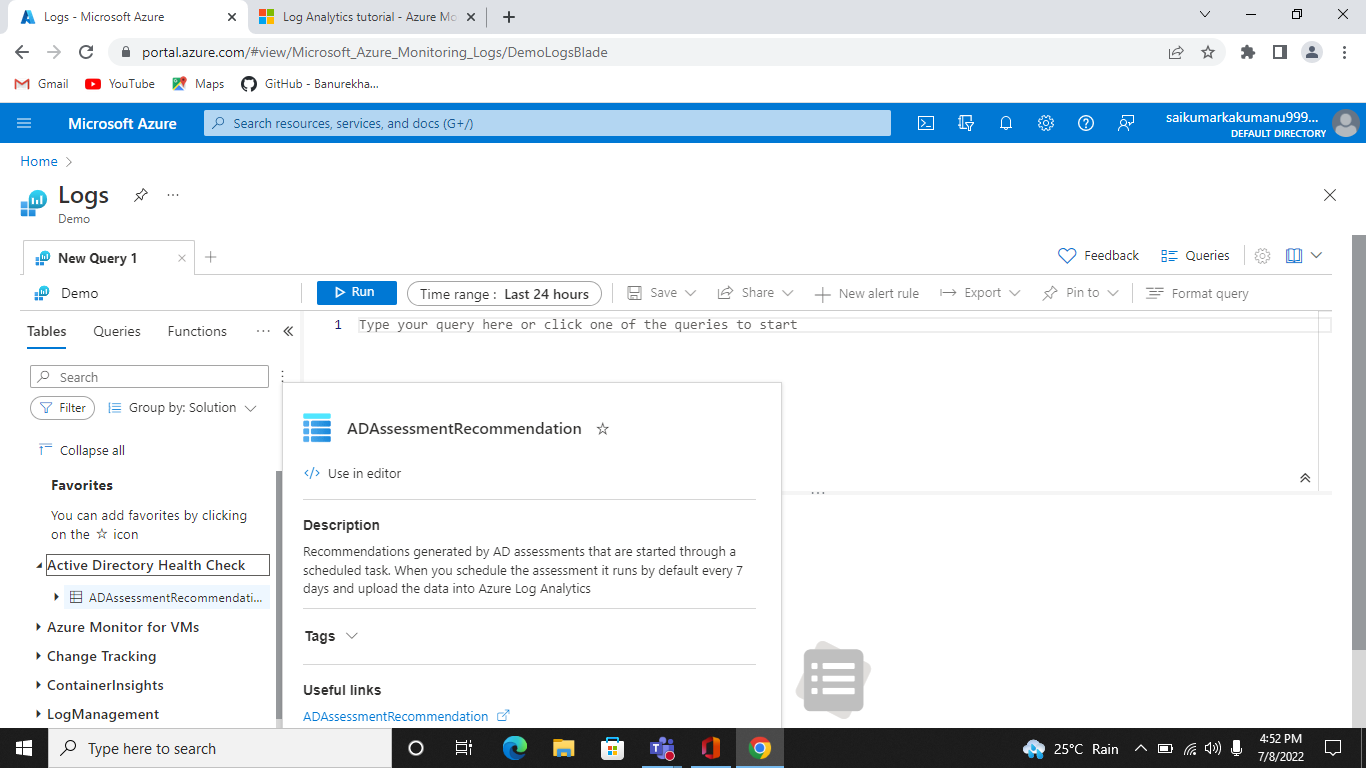
Log Analytics is a tool in the Azure portal to edit and run log queries from data collected by Azure Monitor logs and interactively analyze their results.

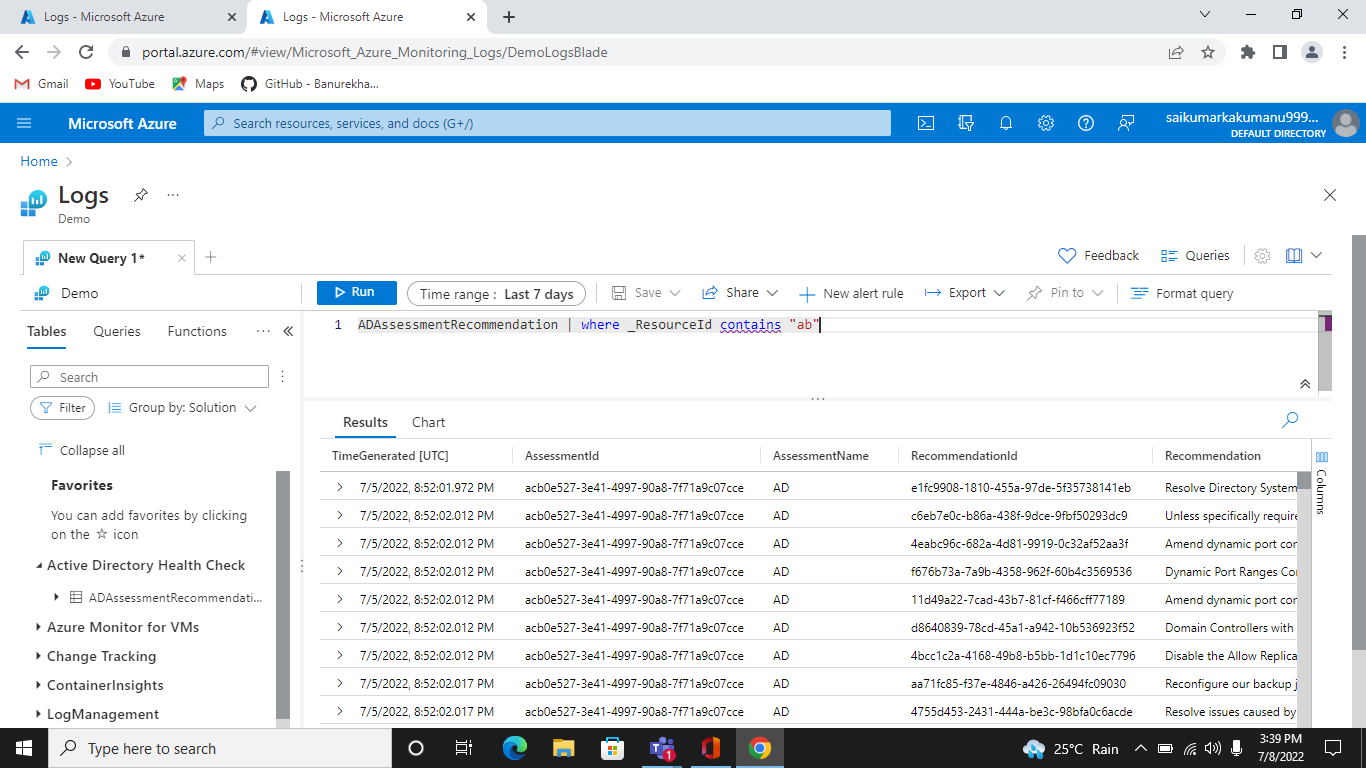
You can use Log Analytics queries to retrieve records that match particular criteria, identify trends, analyze patterns, and provide various insights into your data.

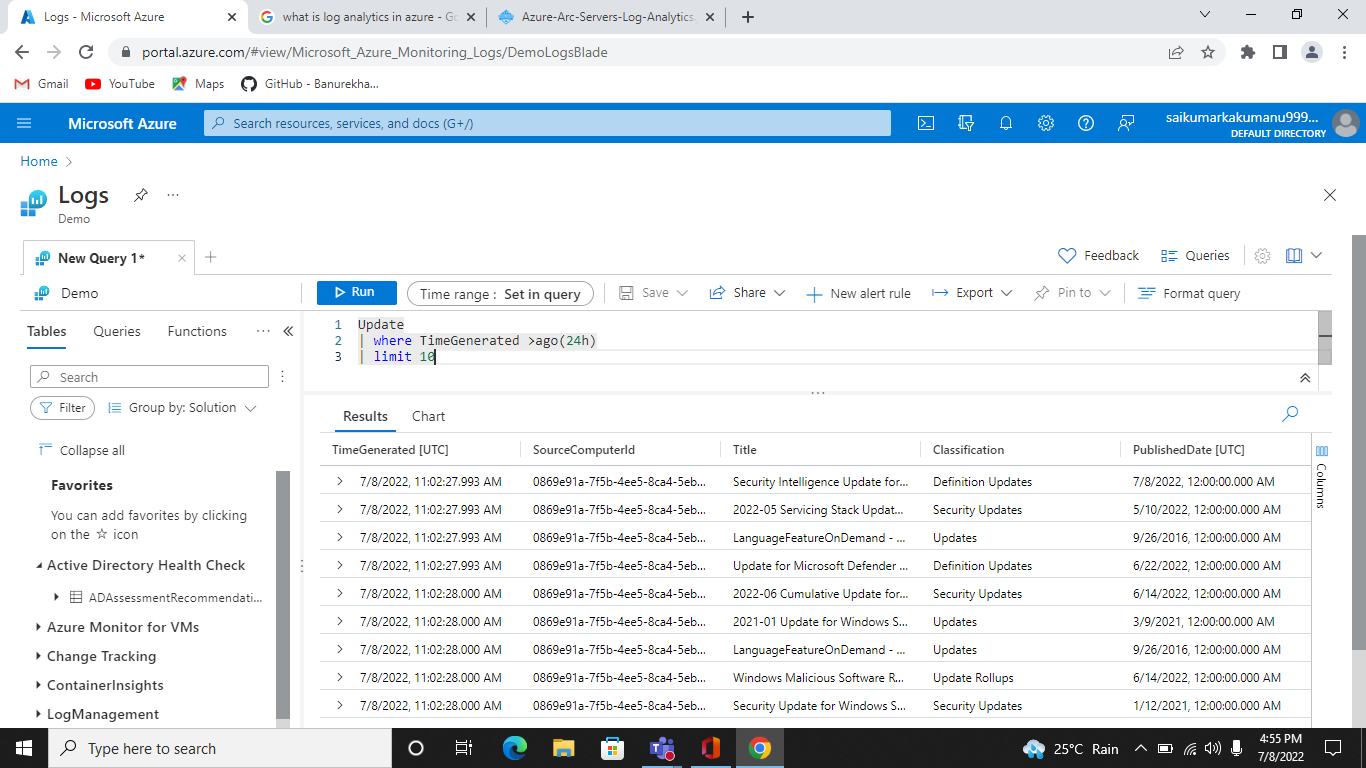
* You might write a simple query that returns a set of records and then use features of Log Analytics to sort, filter, and analyze them. Or you might write a more advanced query to perform statistical analysis and visualize the results in a chart to identify a particular trend.
* If you start Log Analytics from the **Azure Monitor** menu or the **Log Analytics workspaces** menu, you'll have access to all the records in a workspace. If you select **Logs** from another type of resource, your data will be limited to log data for that resource. For more information, see [Log query scope and time range in Azure Monitor Log Analytics](https://docs.microsoft.com/en-us/azure/azure-monitor/logs/scope).

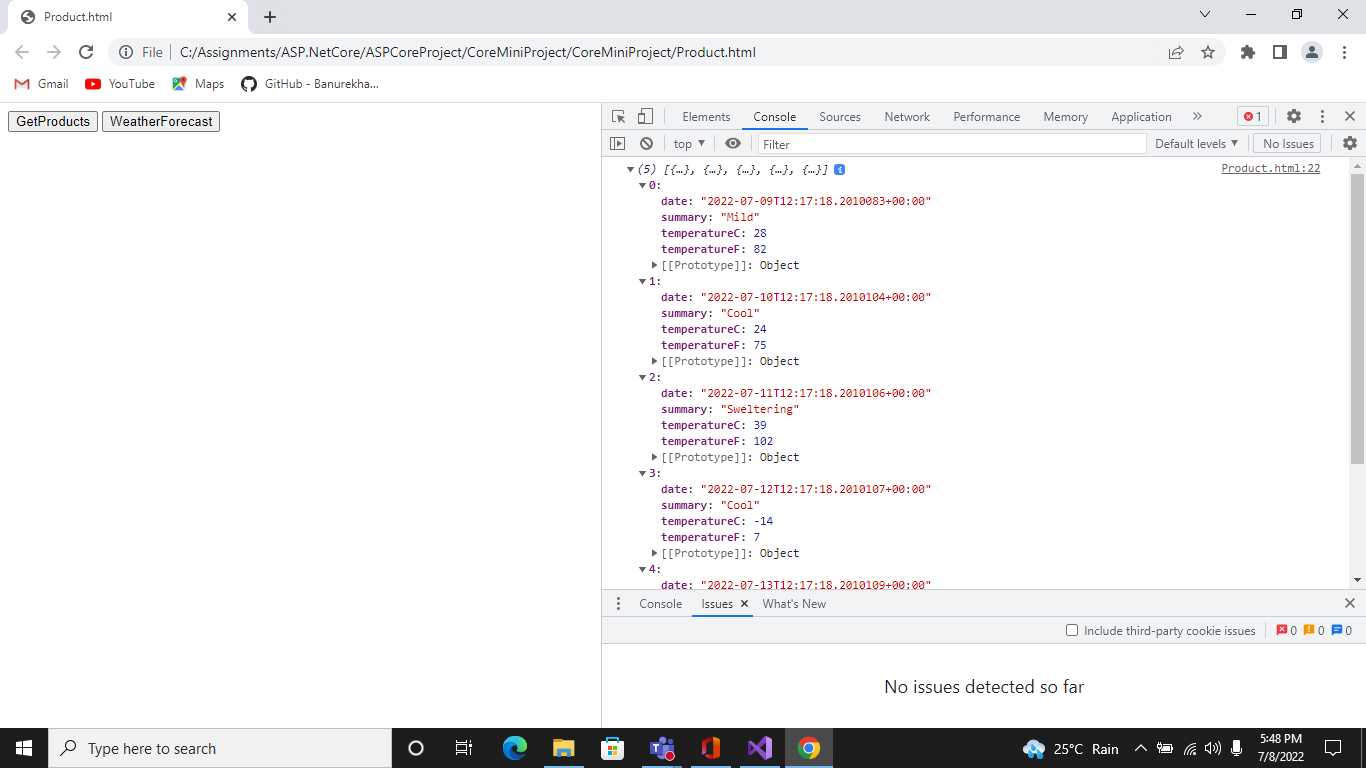
**How Logs work by using Query language as shown in bellow images.**

1. Open Azure portal and click on Logs.
2. Select the Tables what you check.
3. Write a query logic and click on Run.









**Final Project URL :-** <https://coreminiproject20220707142828.azurewebsites.net/weatherforecast>