**FINAL 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

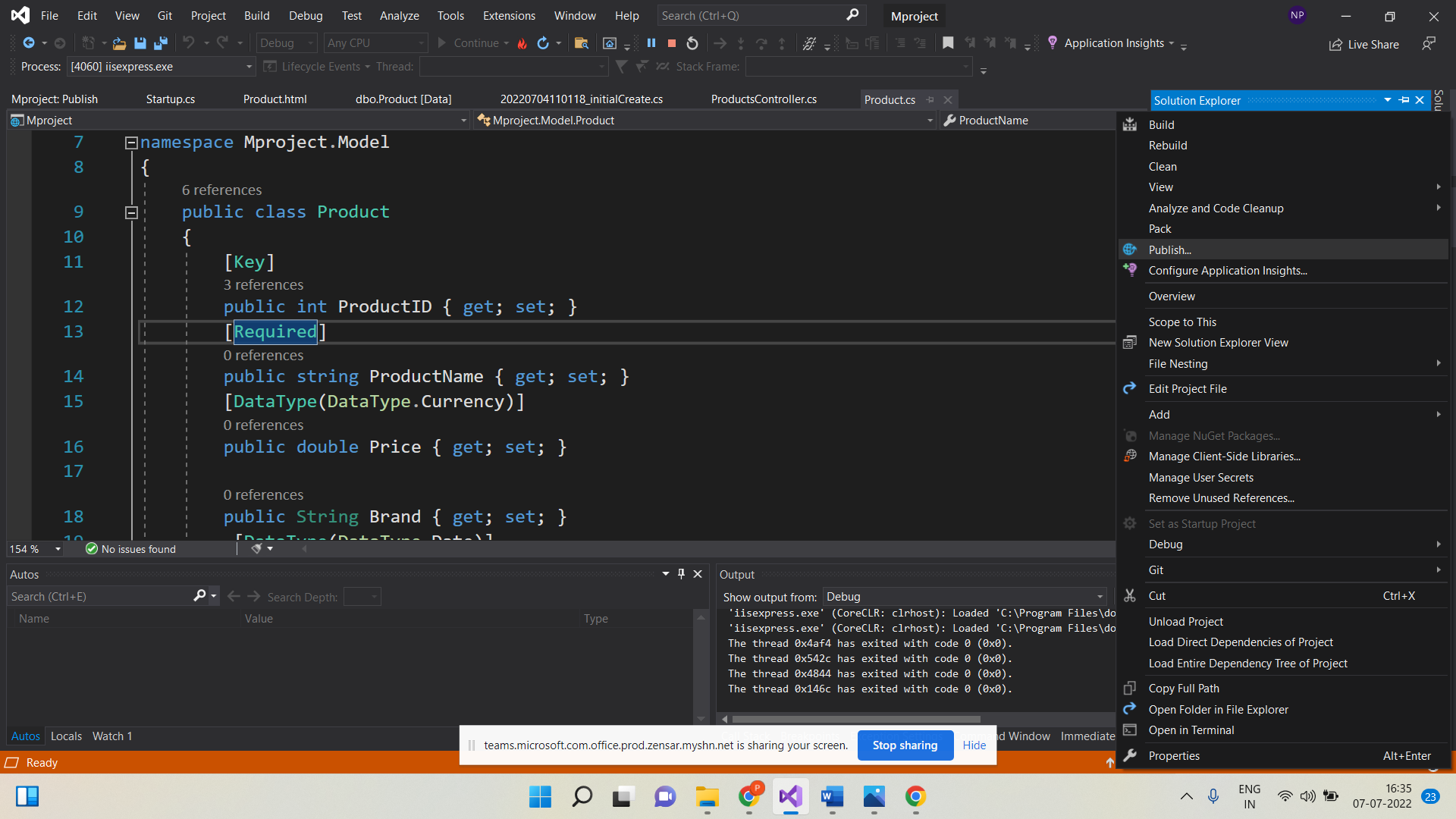
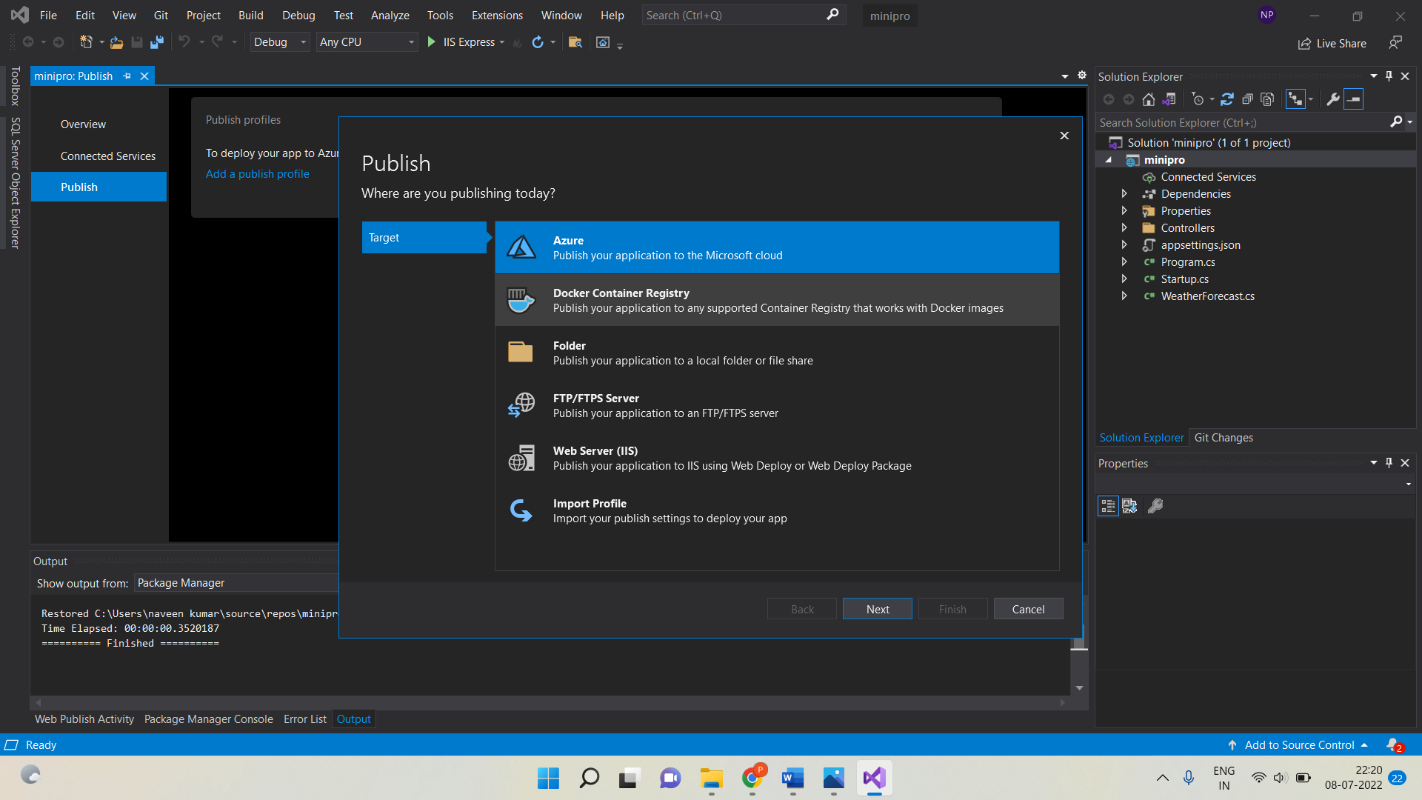
Create a JQuery and AJAX Client to consume the Web API and show the result.

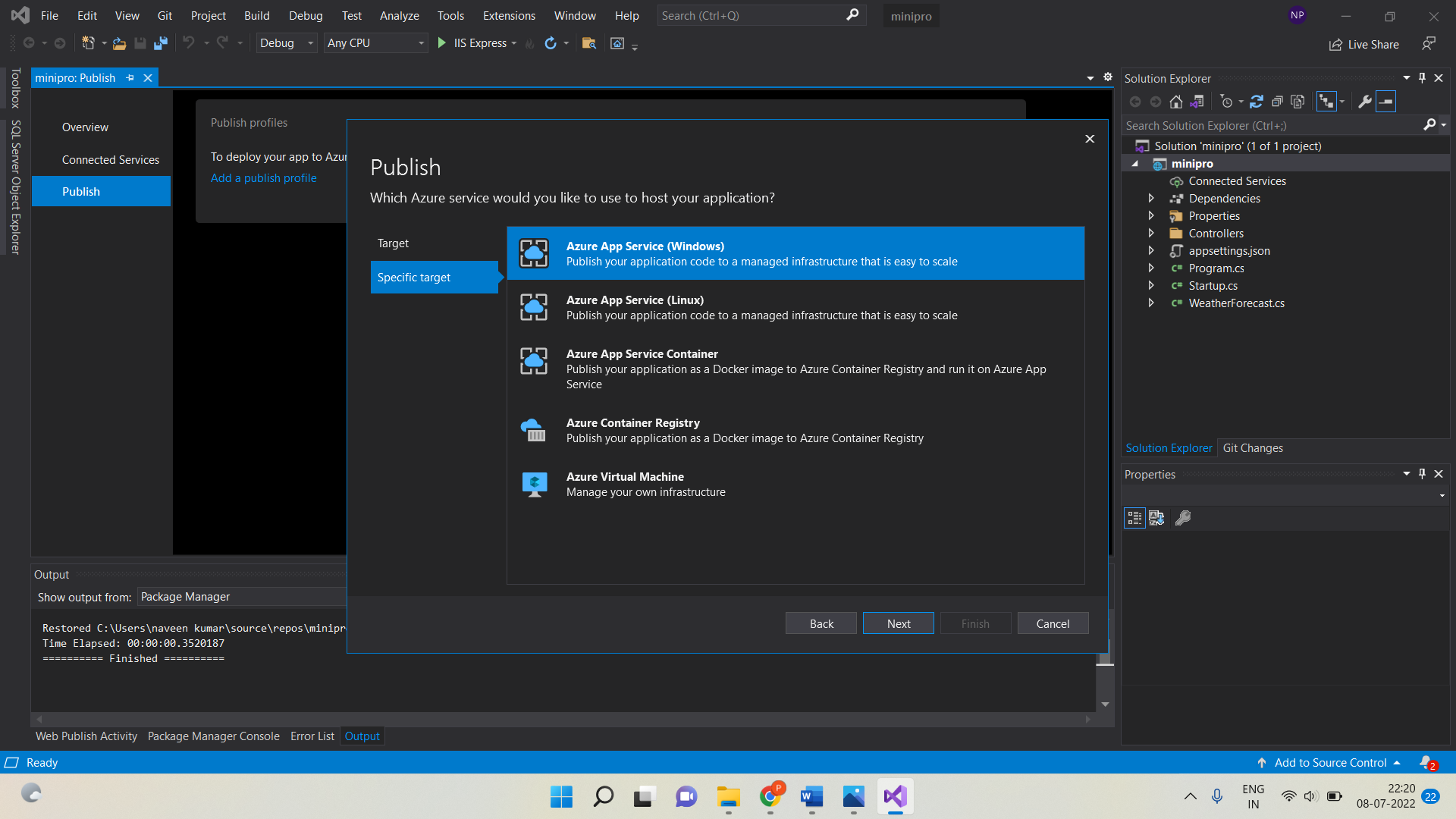
Azure Hosting:

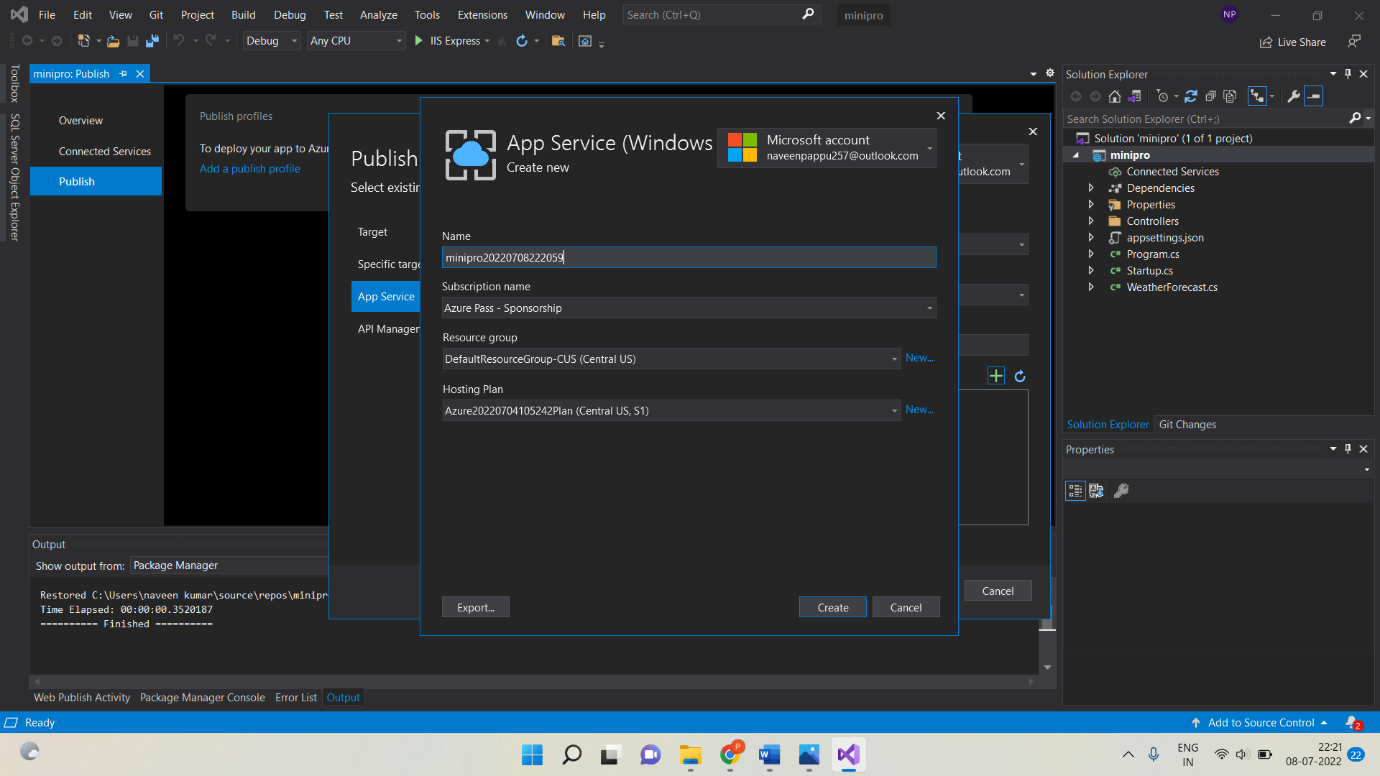
* Host the web api in azure and consume the same using JQuery Client.
* Configure Scale out by adding rules for custom scaling
* Configure Deployment slots for staging and production
* Configure Application Insights for the project
* Configure Swagger for the api
* Work with Log Analytics with the sample logs available

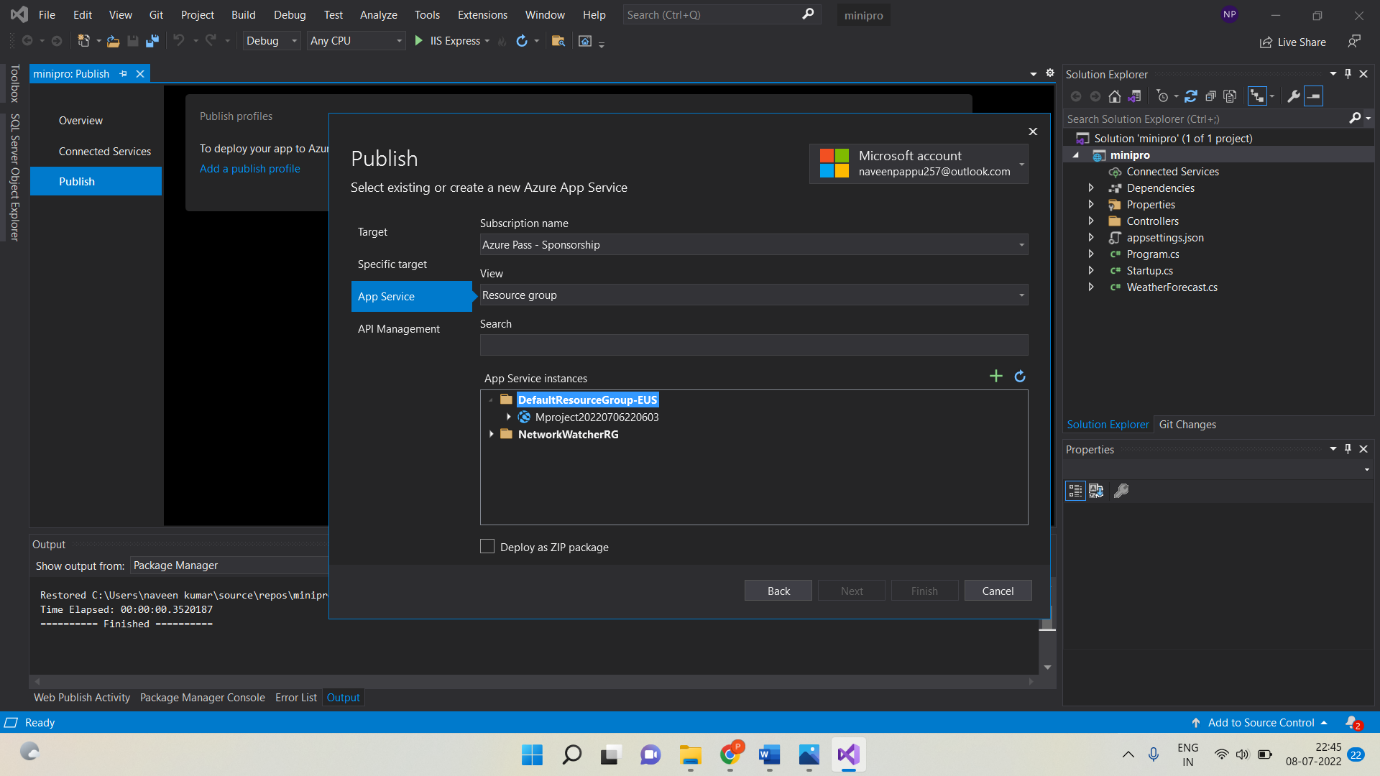
Q1.Host the web api in azure and consume the same using JQuery Client

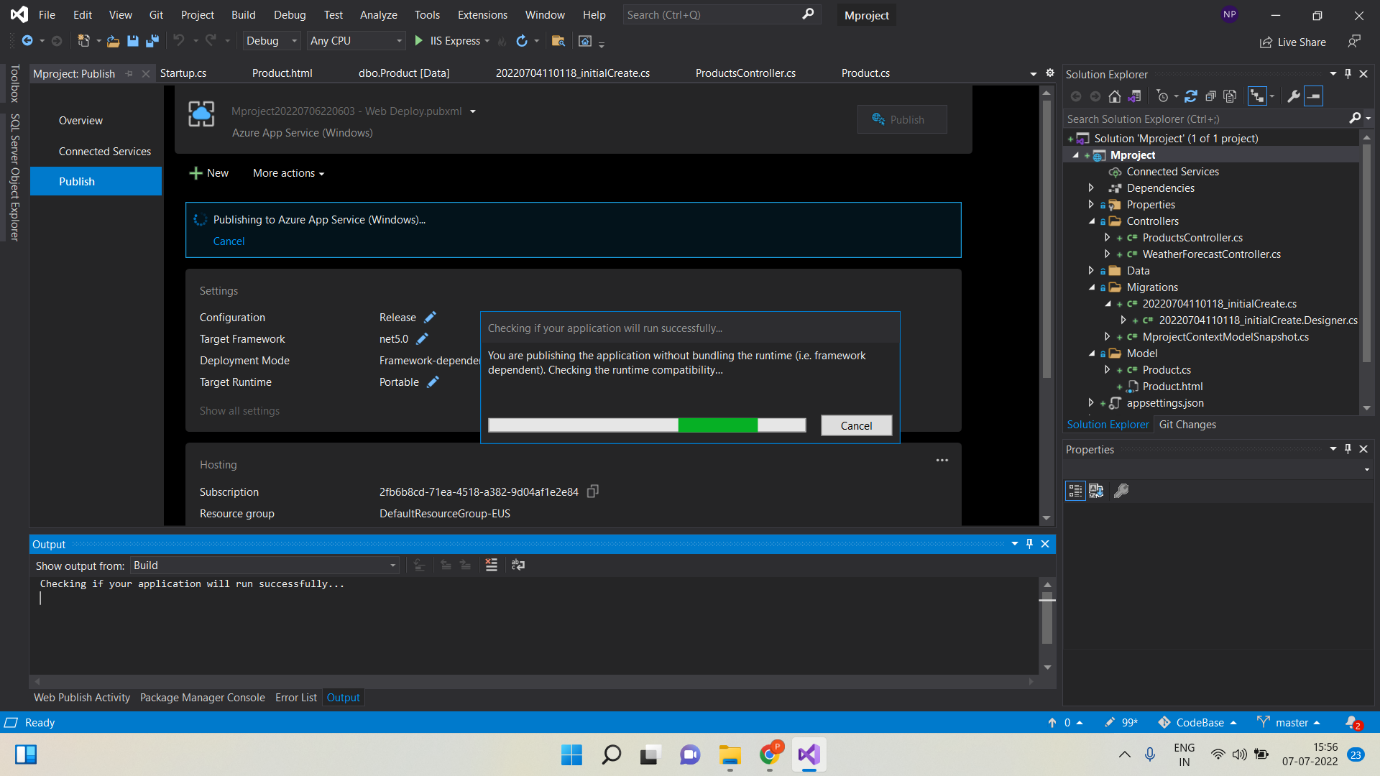
*PUBLISHING API ON AZURE: -*

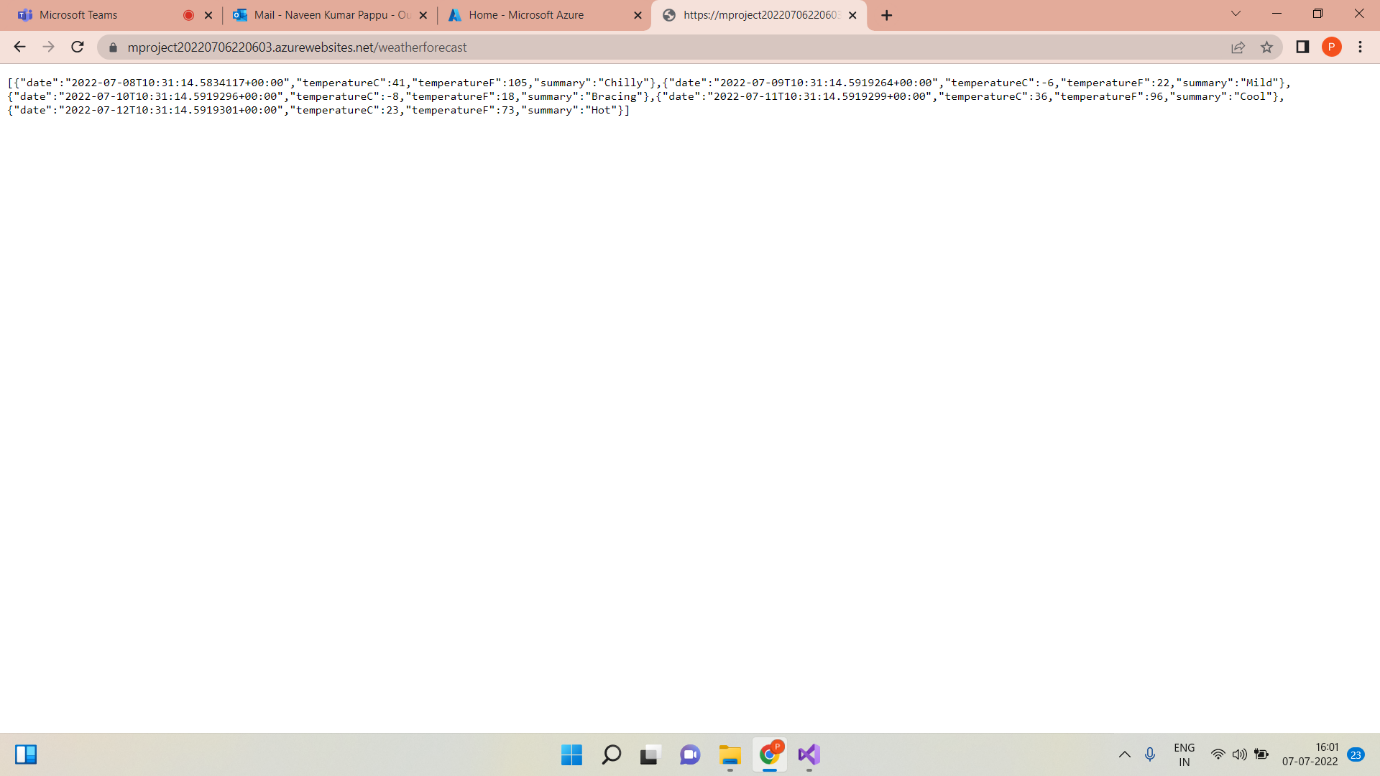
* **In Solution Explorer, right-Click the project and select publish*.*
* In the publish dialog, select Azure and select the Next button.
* Select azure App service (Windows) and select the Next button.



* Select create a new Azure App Service, Select the Create button.
* Click On Next.

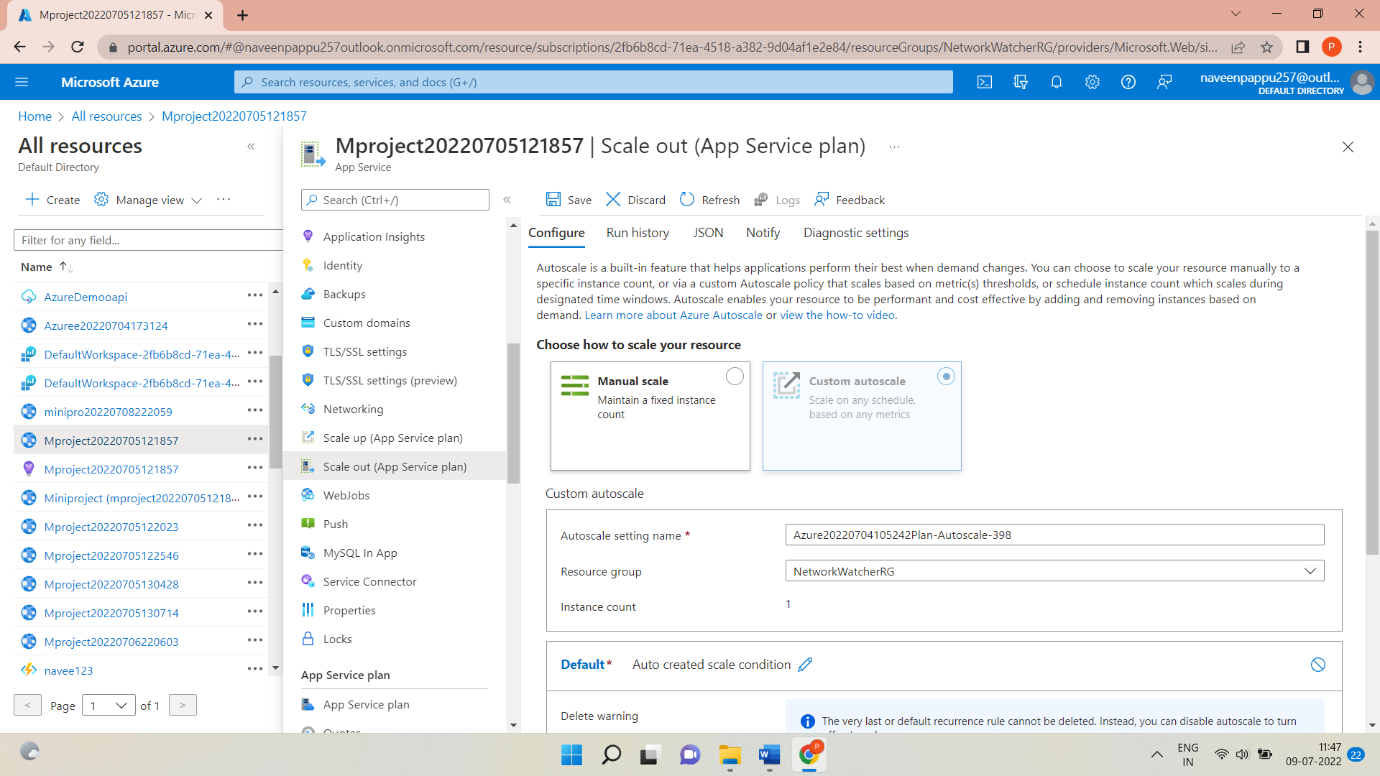


* After creation is completed, the dialog is automatically closed and the publish dialog gets focus again. The instance was created is automatically selected.
* Then if you click on publish button you will get JSON result.

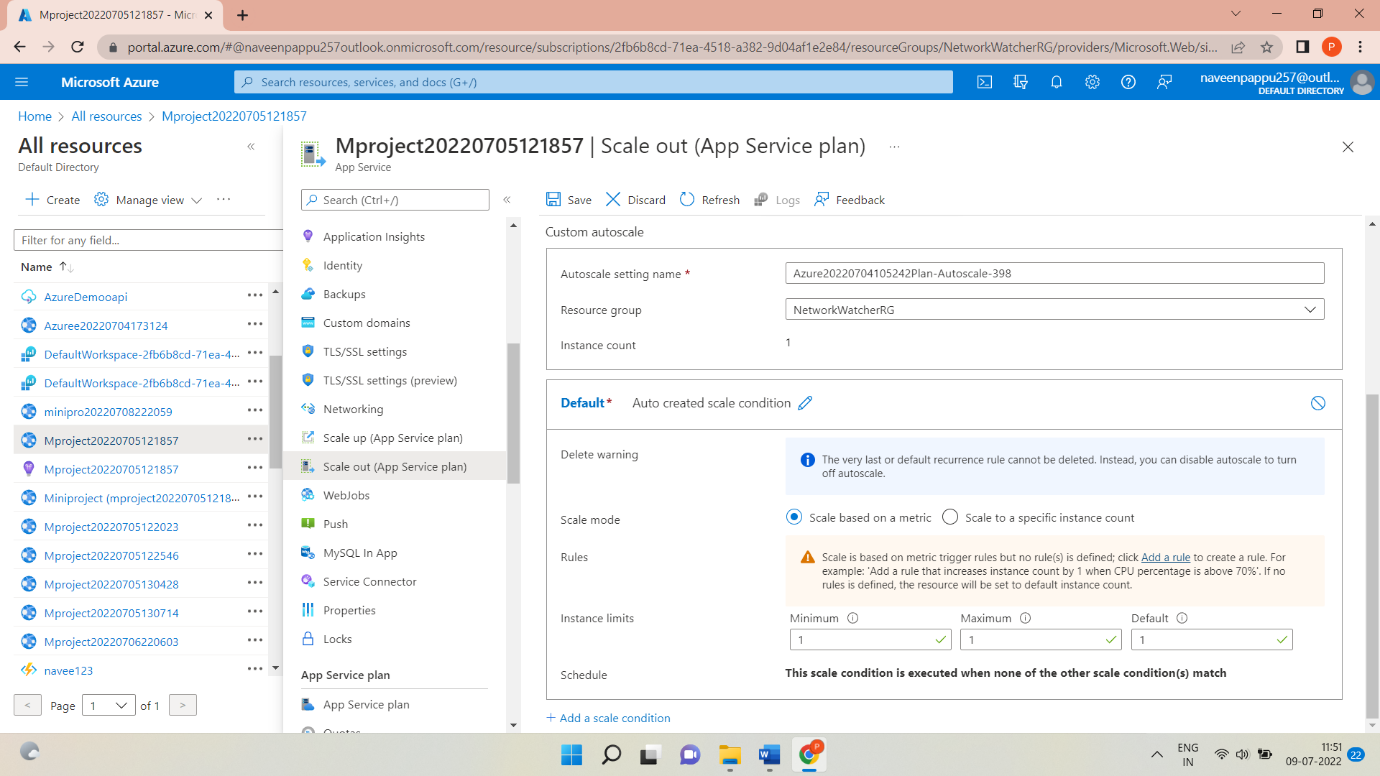


**Configuration of Scale Out :-**

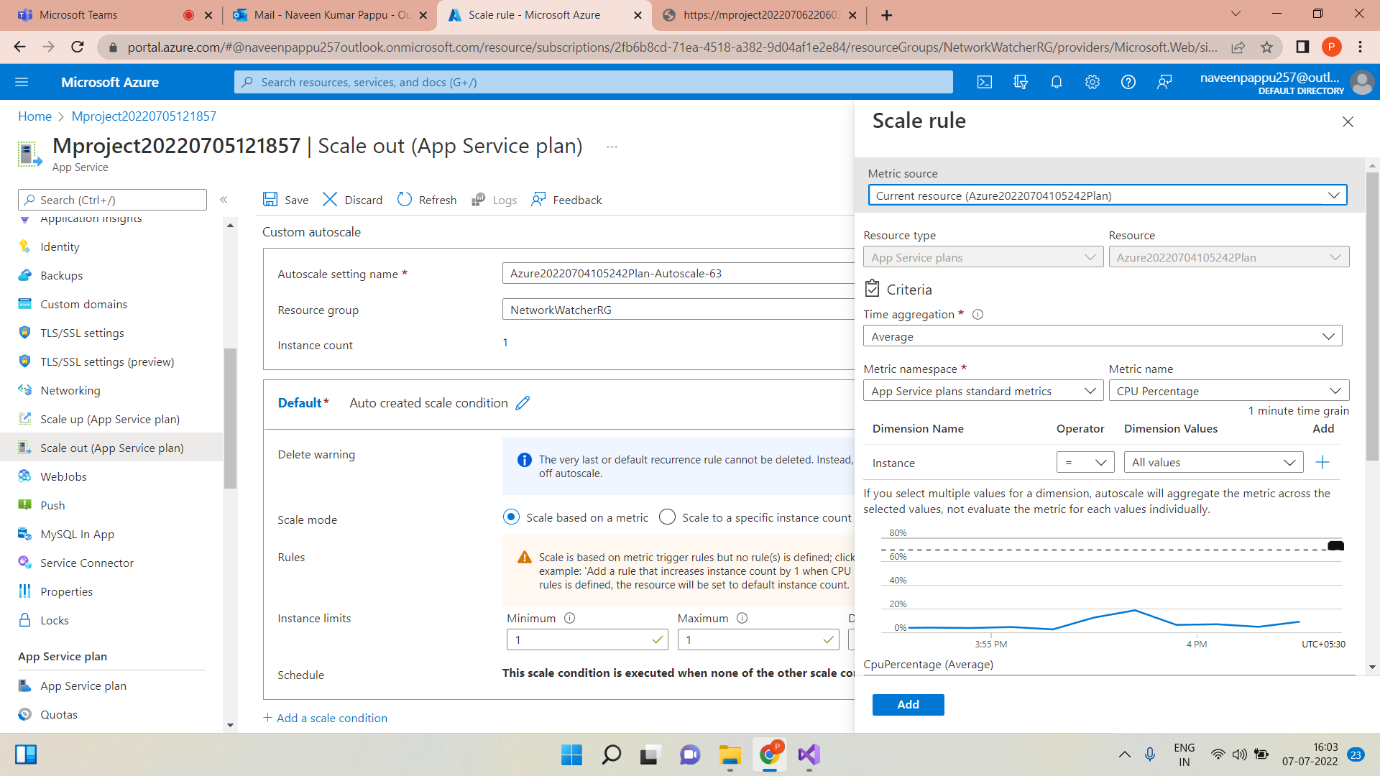
* Open Azure portal and open App service go to settings and select scale out(App service plan),select custom auto scale.



* And then click Add a rule. This opens as a context pane on the right side.



* By default ,this sets the option to scale your instance count by 1 if the CPU percentage of the resource exceeds 70 percent. Leave it at its default values and click Add.

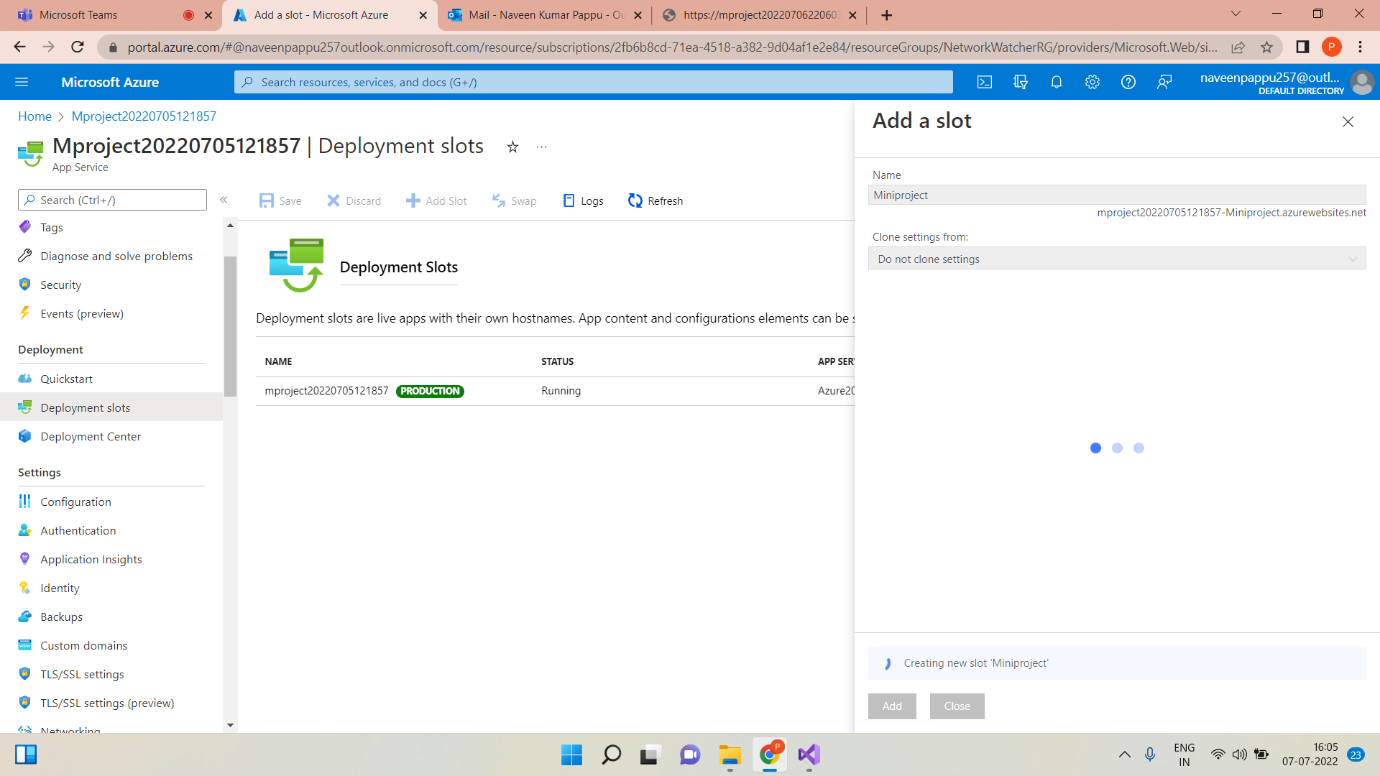


Scale-out:- A scale out operation is the equivalent of creating multiple copies of your web site and adding a load balancer to distribute the demand  between them. When you scale out a web site in Windows Azure Web Sites there is no need to configure load balancing separately since this is already provided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

* To scale out is the process of selling off portions of the total held shareswhile the price increases. To scale out (or scaling out) means to get out of a position (e.g., to sell) in increments as the price climbs.

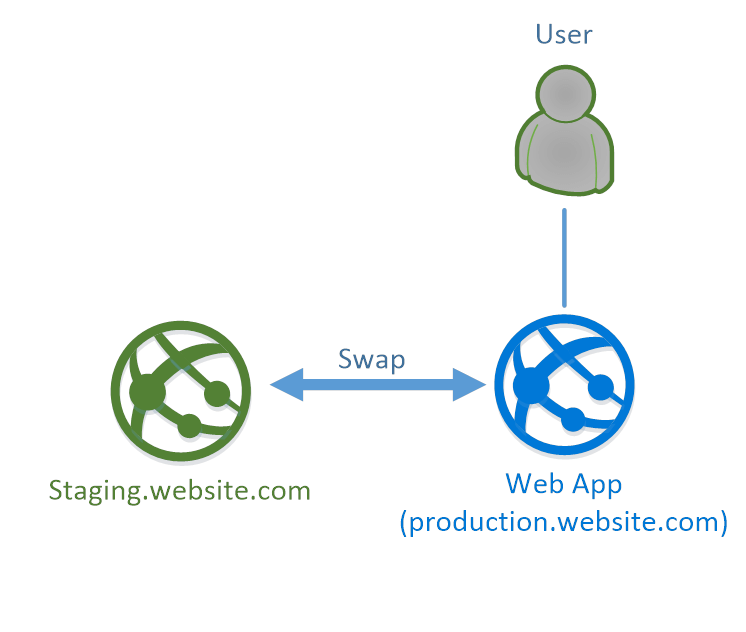
Configuration of Deployment Slots:-

* Select Deployment slots, and then select+ Add slot.



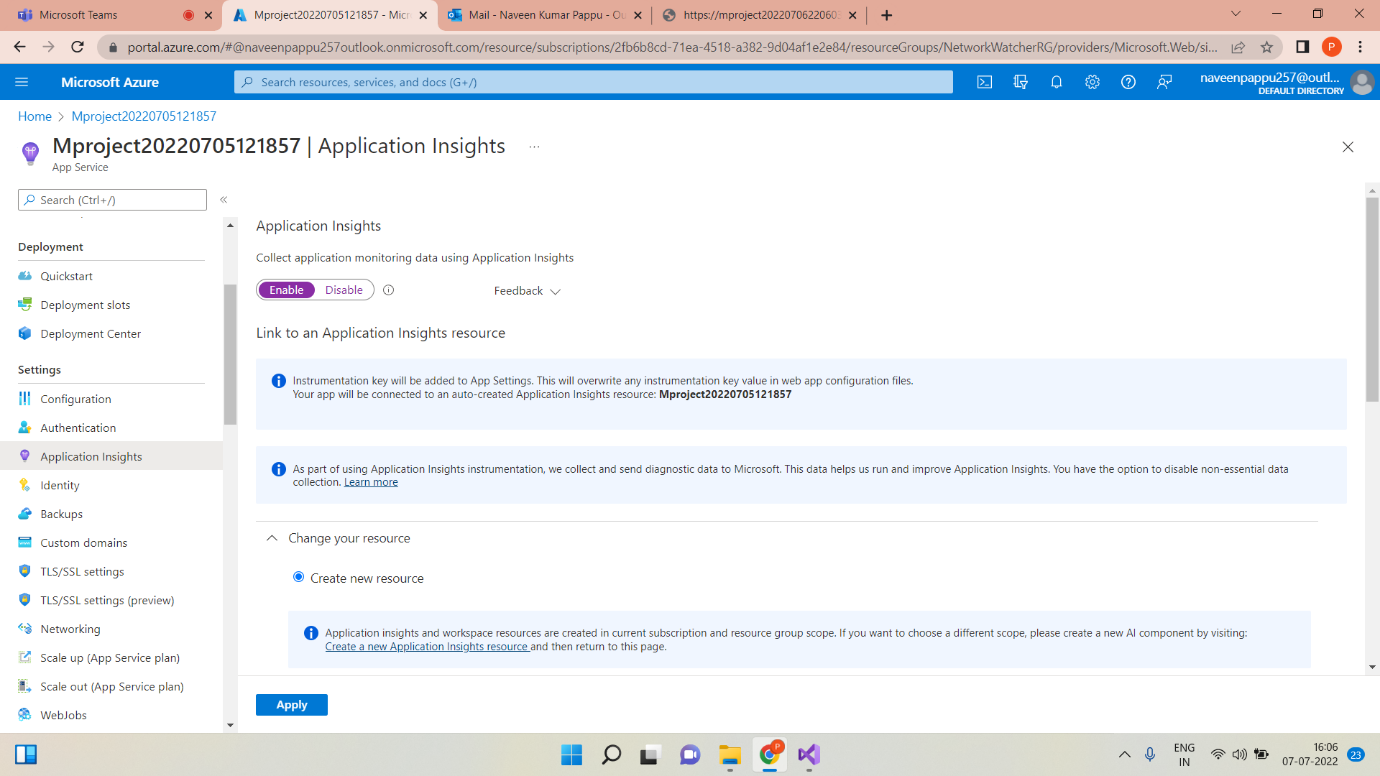
* Verify the configuration settings for your swap and select Swap.

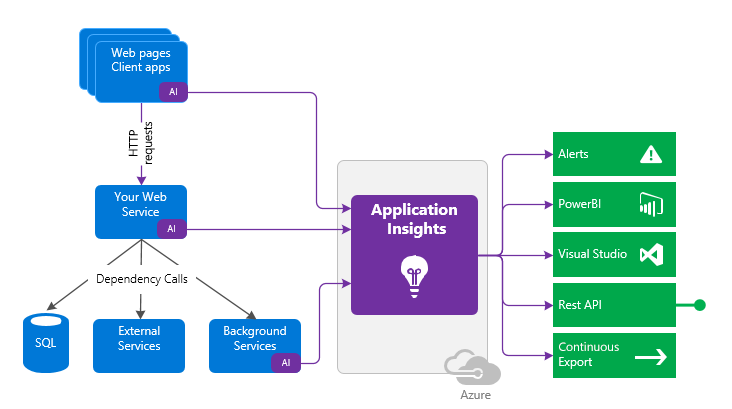
Deployment Slots: -Azure Functions deployment slots allow your function app to run different instances called "slots". Slots are different environments exposed via a publicly available endpoint. One app instance is always mapped to the production slot, and you can swap instances assigned to a slot on demand.

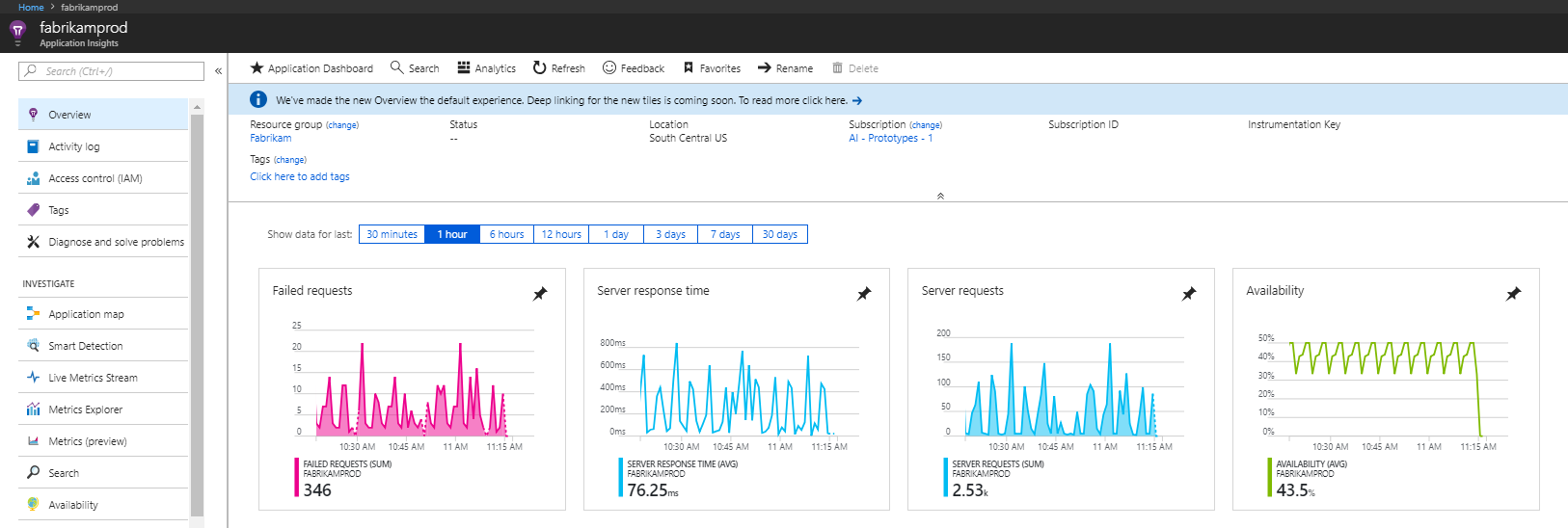


* Deployment slots are live apps with their own host names. App content and configurations elements can be swapped between two deployment slots, including the production slot.
* Testing in production is easy to set up. In the Azure Portal, choose the testing in production menu option in your App Service. Here, you can choose the deployment slots that you want to route traffic to (or add new ones) and assign a percentage of traffic to them. That's it!

Configuration of Application Insights: -







Application Insights: - Application Insights is a feature of Azure Monitor that provides extensible application performance management (APM) and monitoring forlive web apps. Developers and DevOps professionals can use Application Insights to: Automatically detect performance anomalies. Help diagnose issues by using powerful analytics tools.

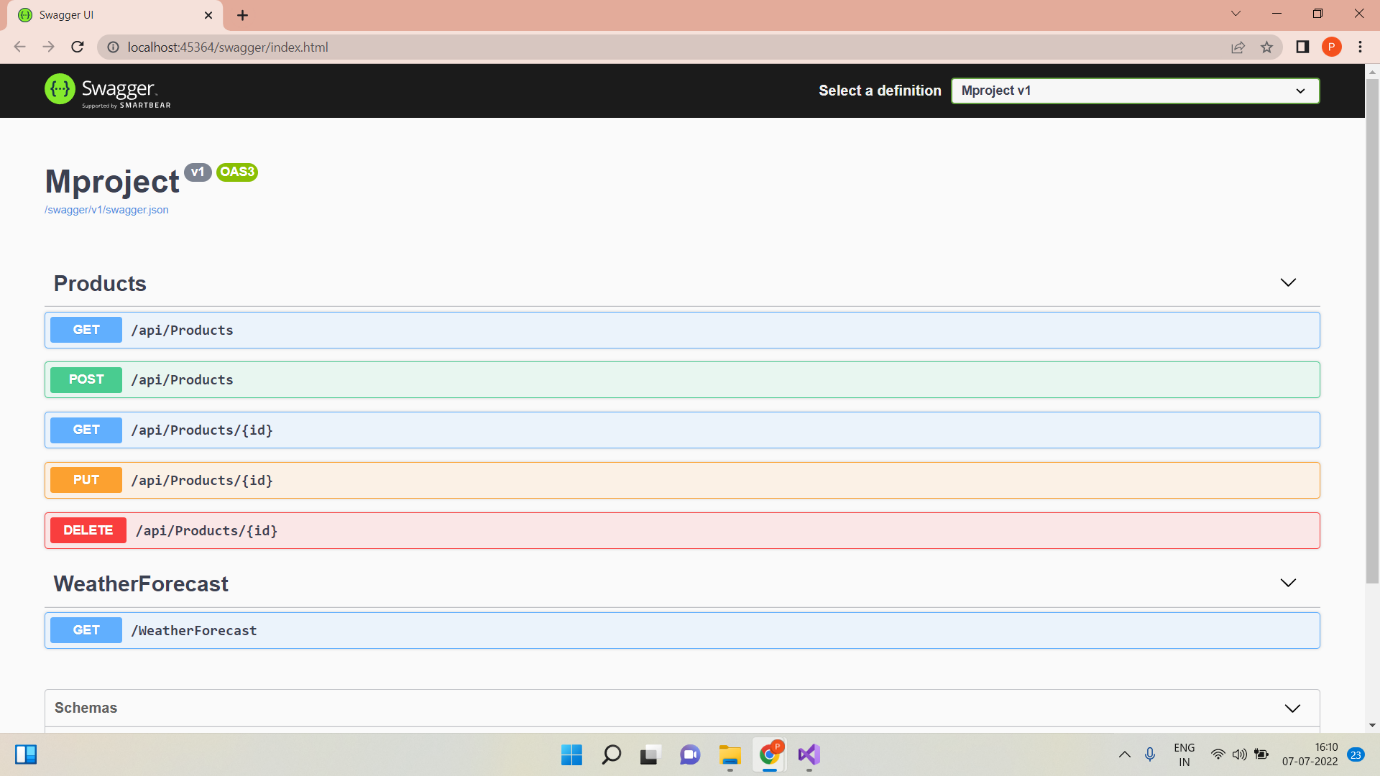
* Automatically detect performance anomalies.
* Help diagnose issues by using powerful analytics tools.
* See what users actually do with apps.
* Supports a wide variety of platforms ,including .NET, Node.js, java, and Python.
* Works for apps hosted on-premises, hybrid, or on any public cloud.
* Integrates with DevOps process.
* Has connection points to many development tools.
* Can monitor and analyses telemetry from mobile apps by integrating with visual studio app center.

Configuration Swagger for API:-

Swagger:- Swagger allows you to describe the structure of your APIs so that machines can read them. The ability of APIs to describe their own structure is the root of all awesomeness in Swagger. Why is it so great? Well, by reading your API’s structure, we can automatically build beautiful and interactive API documentation. We can also automatically generate client libraries for your API in many languages and explore other possibilities like automated testing. Swagger does this by asking your API to return a YAML or JSON that contains a detailed description of your entire API.

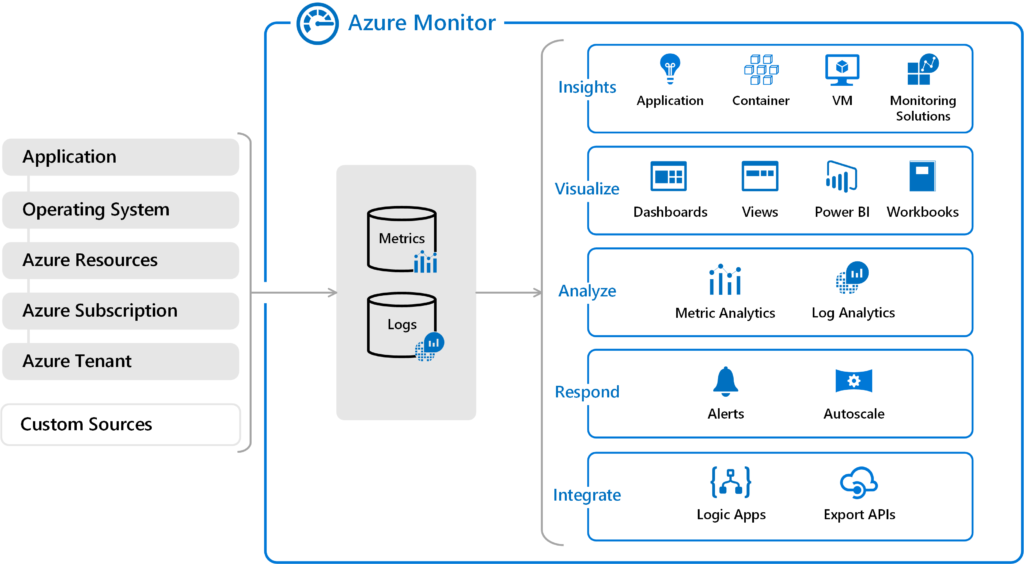
There are a few ways in which Swagger can help drive your API development further:

* Design-first users: use Swagger Codegen to generate a server stub for your API. The only thing left is to implement the server logic – and your API is ready to go live!
* Use [Swagger Codegen](https://swagger.io/swagger-codegen/) to generate client libraries for your API in over 40 languages.
* Use [Swagger UI](https://swagger.io/swagger-ui/) to generate interactive API documentation that lets your users try out the API calls directly in the browser.
* Use the spec to connect API-related tools to your API. For example, import the spec to [SoapUI](https://soapui.org/) to create automated tests for your API.



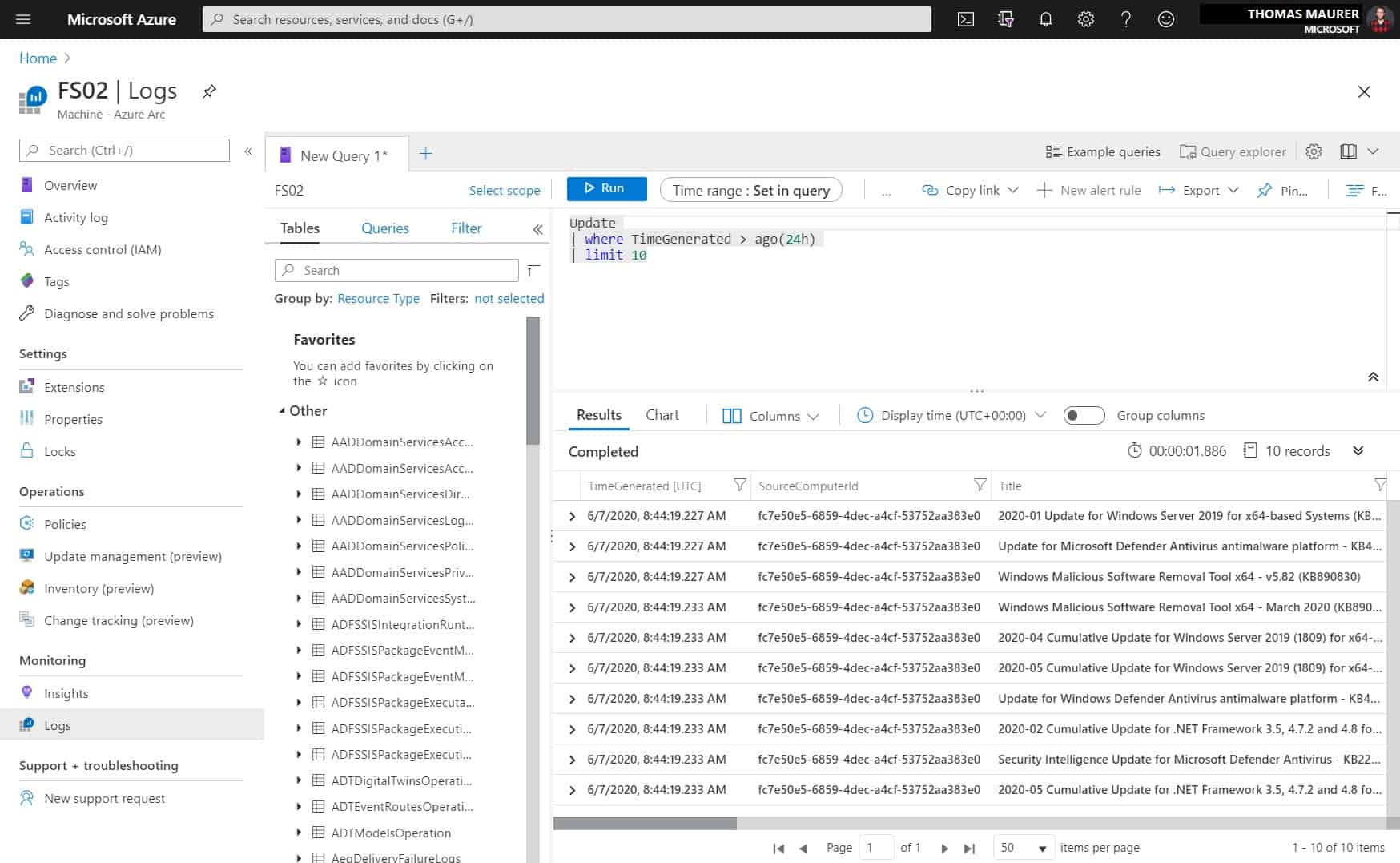
* Log Analytics: -

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

* Write a Query

In addition to helping, you write and run queries, Log Analytics provides features for working with the results. Start by expanding a record to view the values for all of its columns. Select the name of any column to sort the results by that column. Select the filter icon next to it to provide a filter condition.

This is the simplest query that we can write .it just returns all the records in a table. run it by selecting the run button or by selecting shift+ Enter with the cursor positioned anywhere in the query text.

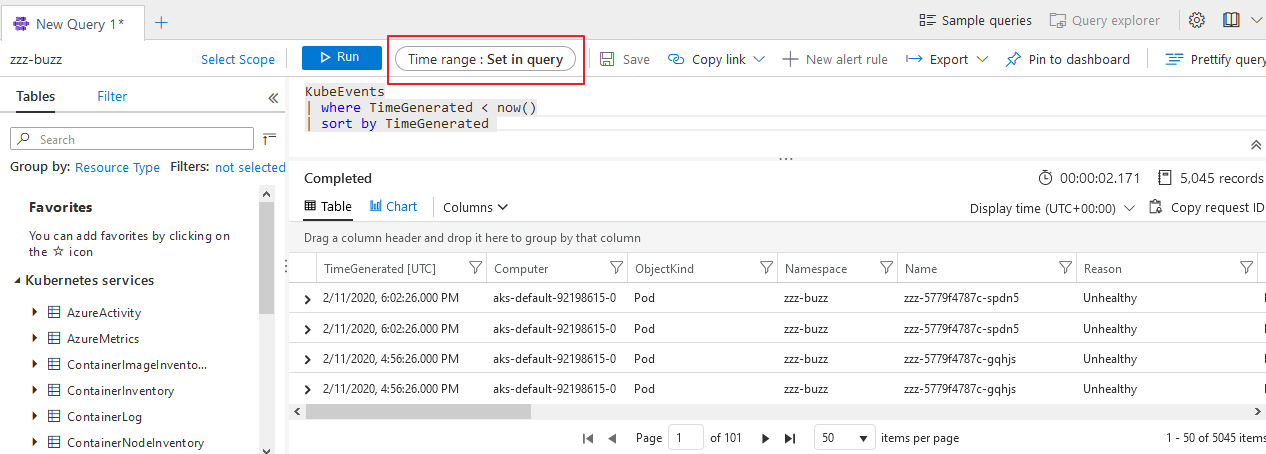


* TIME RANGE: -

All queries return records generated within a set time range. By default, the query returns records generated in the last 24 hours.

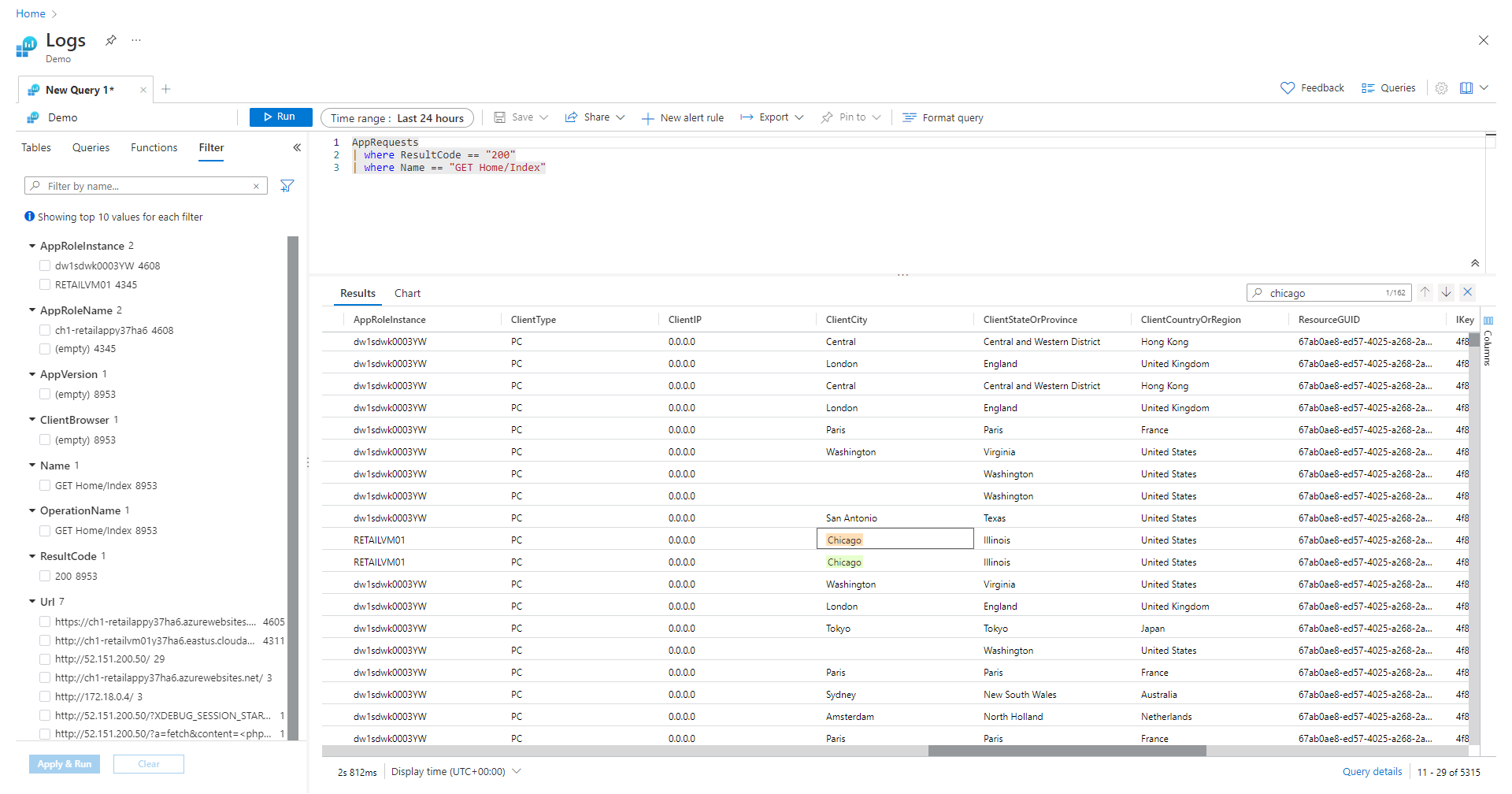
You can set a different time range by using the Where Operator in the query. You can also use the time range dropdown list in the top of the screen.

Let’s change the time range of the query by selecting last 12 hours from time range dropdown. Select Run to returns the results.



* Analyze results

In Addition to helping you write and run queries, Log analytics provides features for working with the results .start by expanding a record to view the values for all of its columns.



* THANKS&REGARDS

Naveen kumar pappu