

DOTNET CORE AZURE

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:

- Product ID
- ProductName
- Price
- Brand
- Manufacture Date
- Expiration Date

UseDataAnnotations:

- 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:

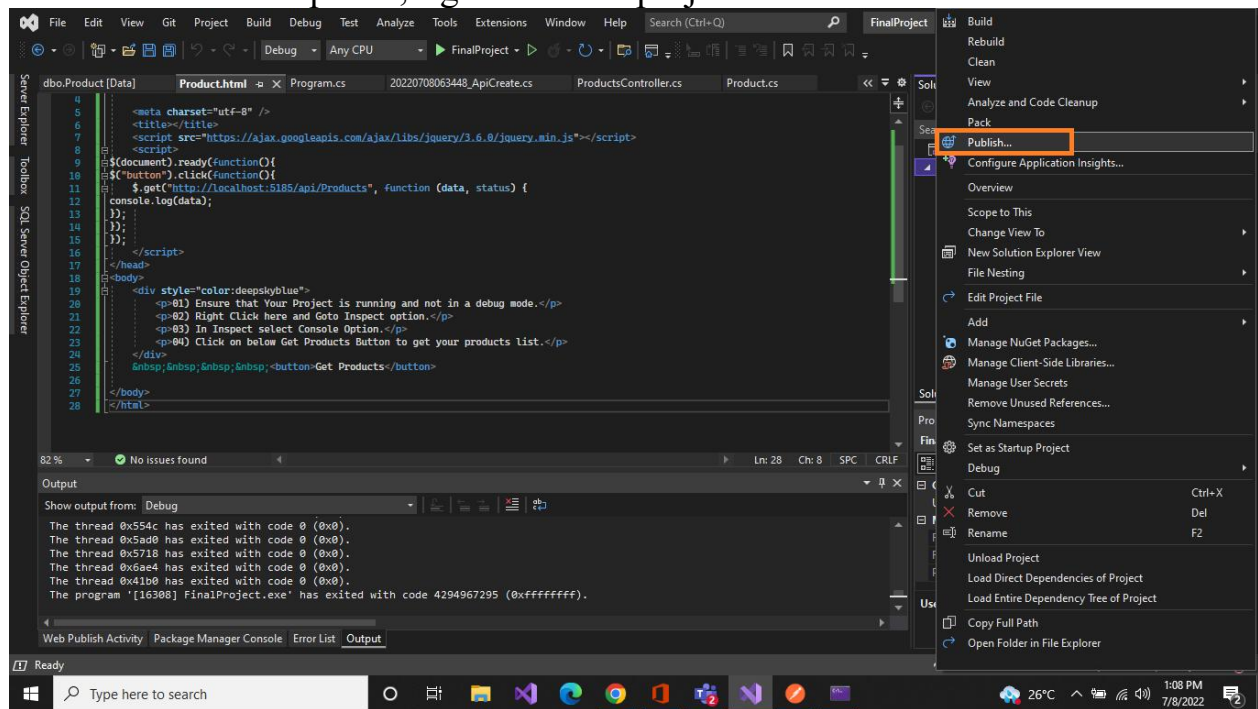
- Host the web api in azure and consume the same using jQuery Client.
- Configure Scale out by adding rules for custom scaling
- Configure Deployment slots for staging and production
- Configure Application Insights for the project
- Configure Swagger for the api
- Work with Log Analytics with the sample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

CREATING WEB API PROJECT : -

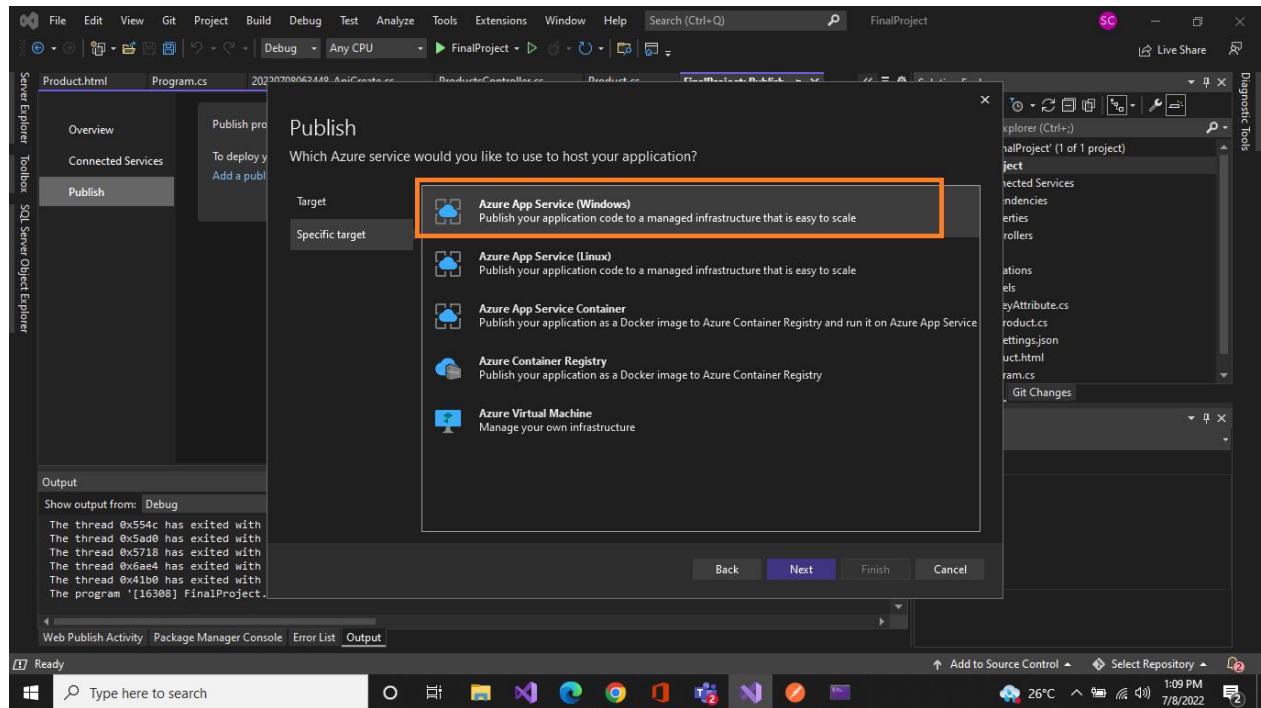
1. Start with a new project on visual studio, with project type as **ASP.NET Core Web API**.
2. Create a new **Folder** called **Models** with class **Products** with the given specifications, added all the **validation attributes** on to the fields of the Product class.
3. Added required **entity framework** 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 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.

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

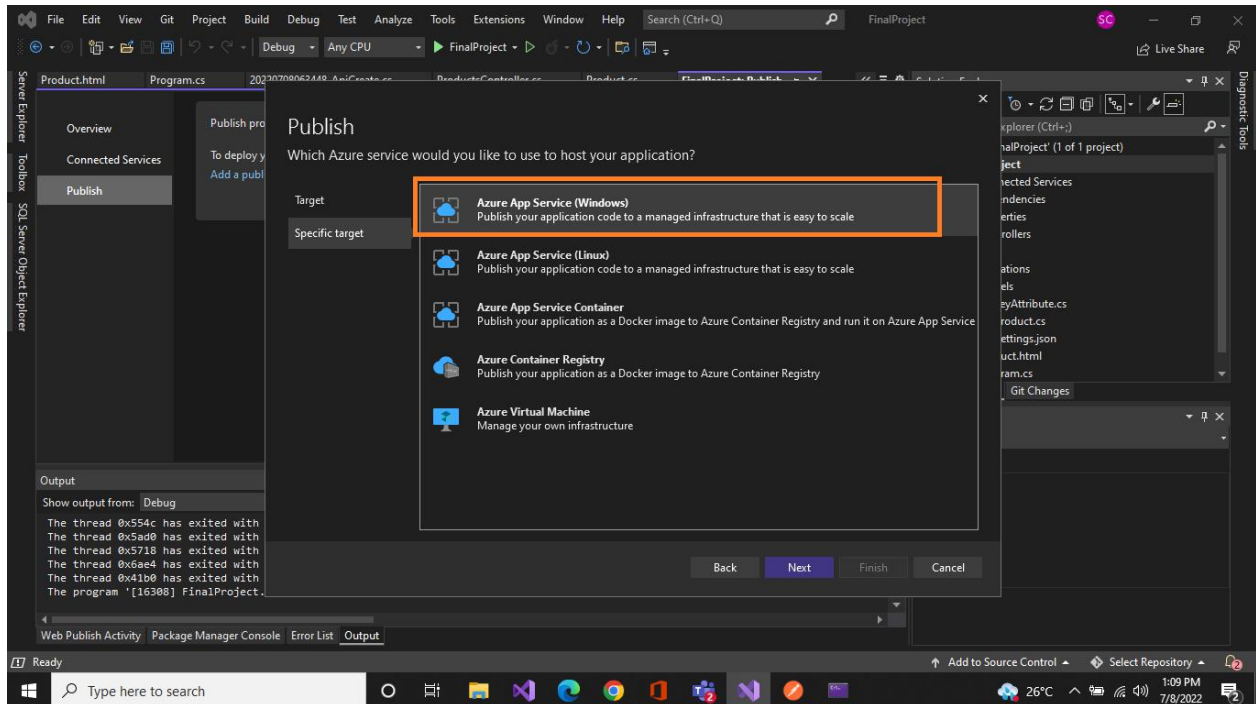
✧ In Solution Explorer, right-click the project and select Publish .



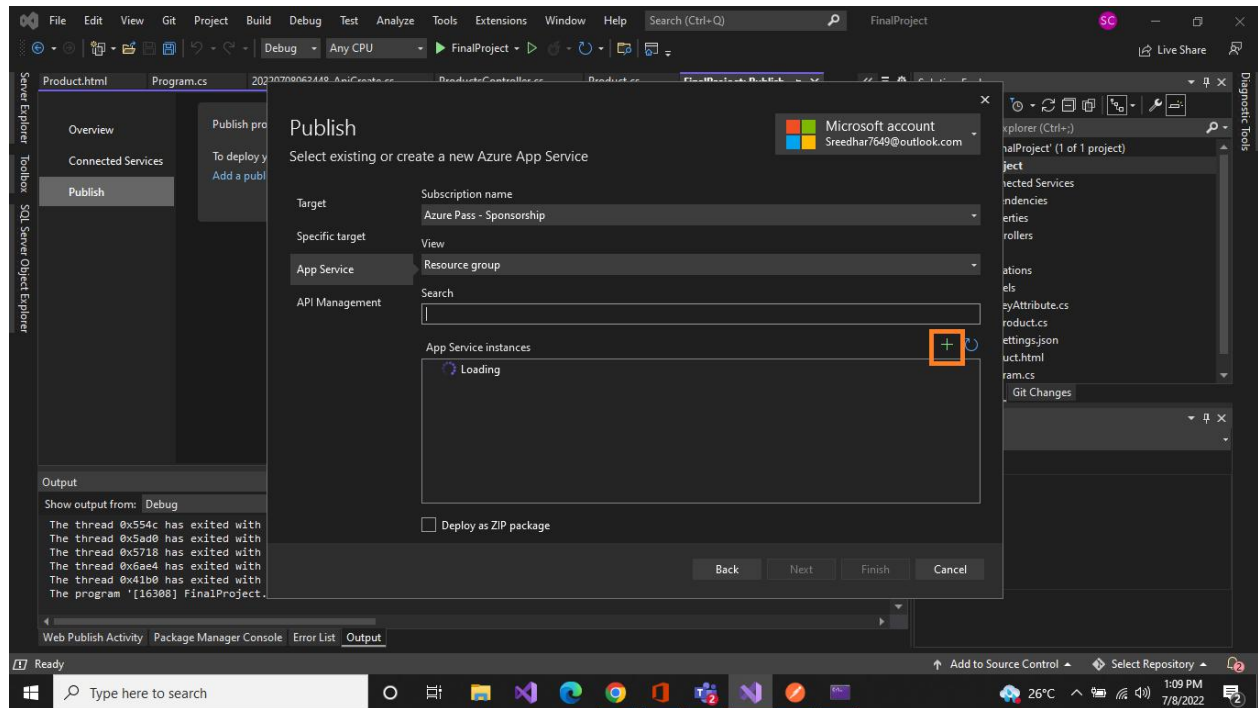
✧ In the Publish dialog, select Azure and select the Next button.



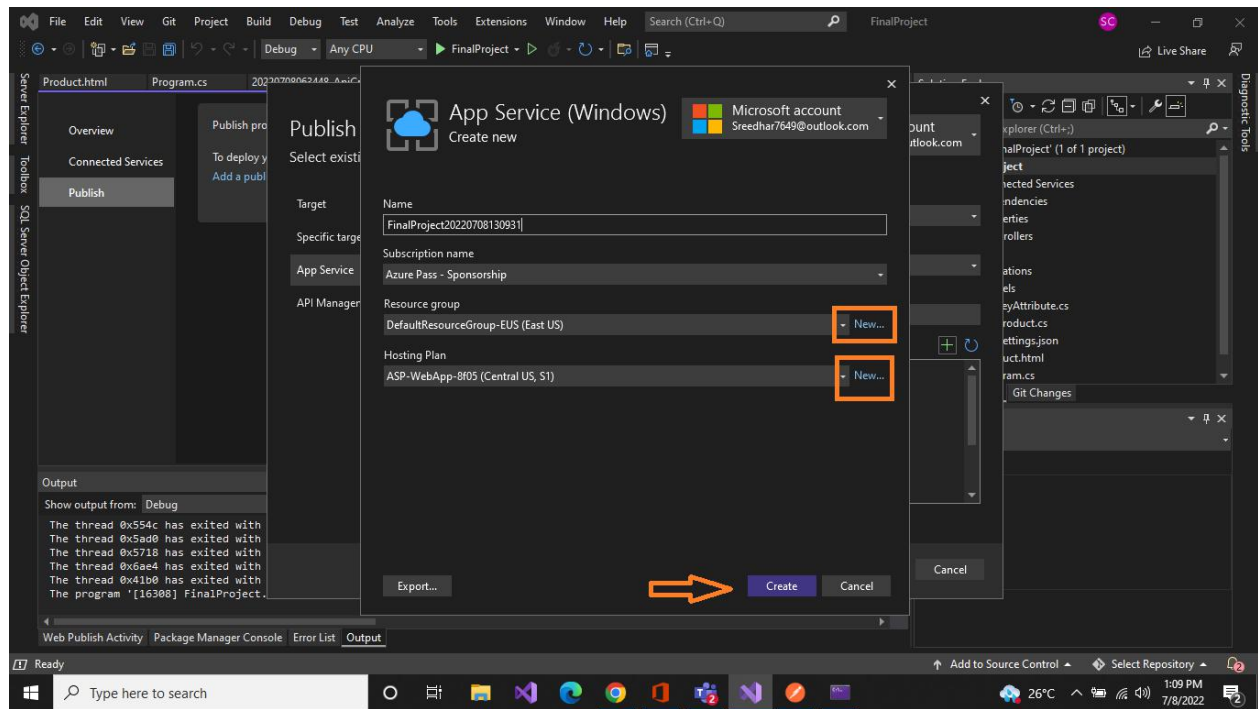
✧ Select Azure App Service (Windows) and select the Next button .



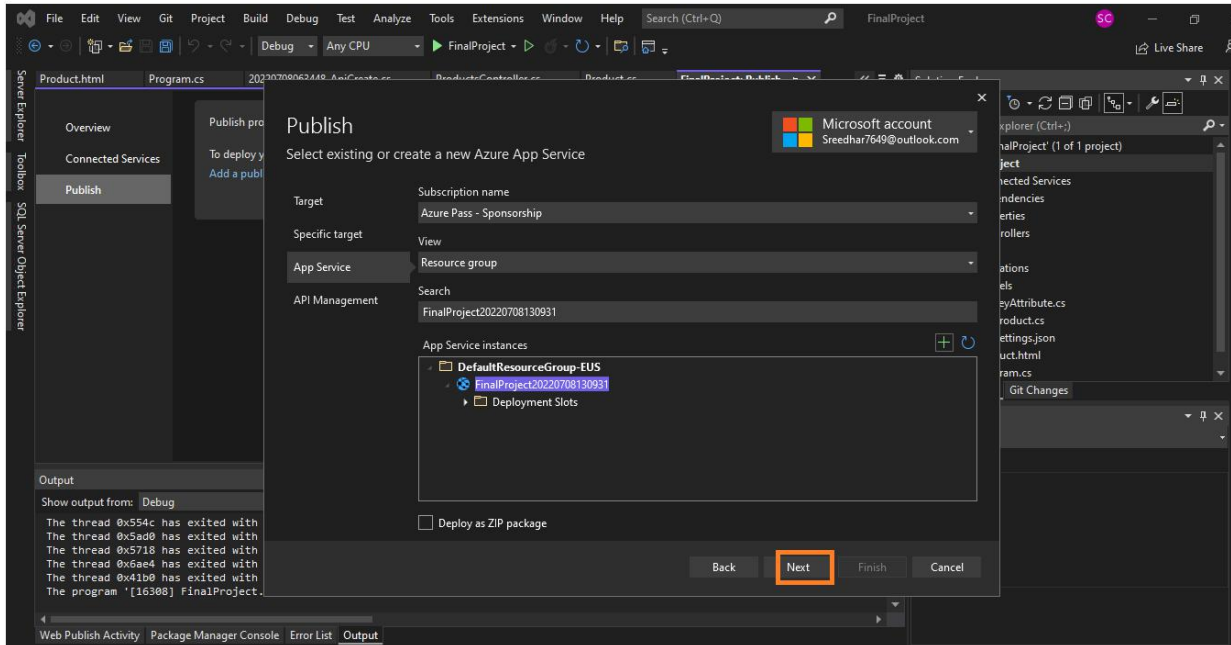
❖ Select Create a new Azure App Service.



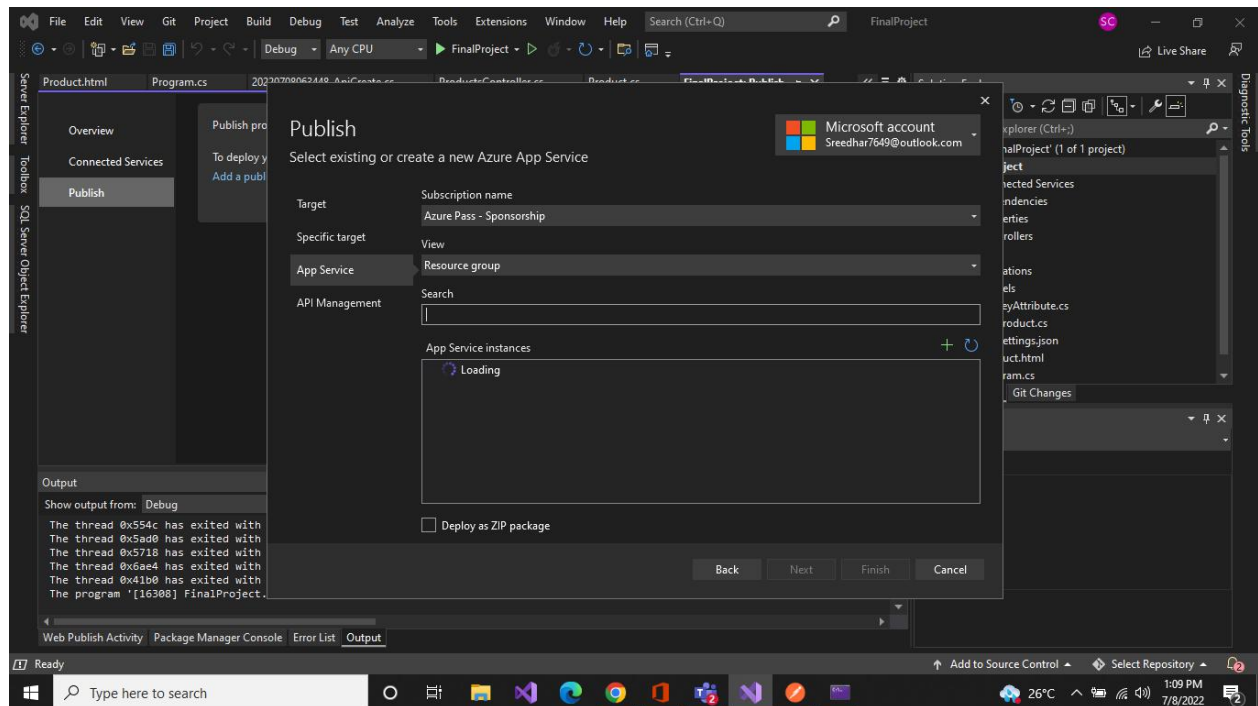
- ✧ The Create App Service dialog appears. The App Name, Resource Group, and App Service Plan entry fields are populated. You can keep these names or change them. Select the Create button.



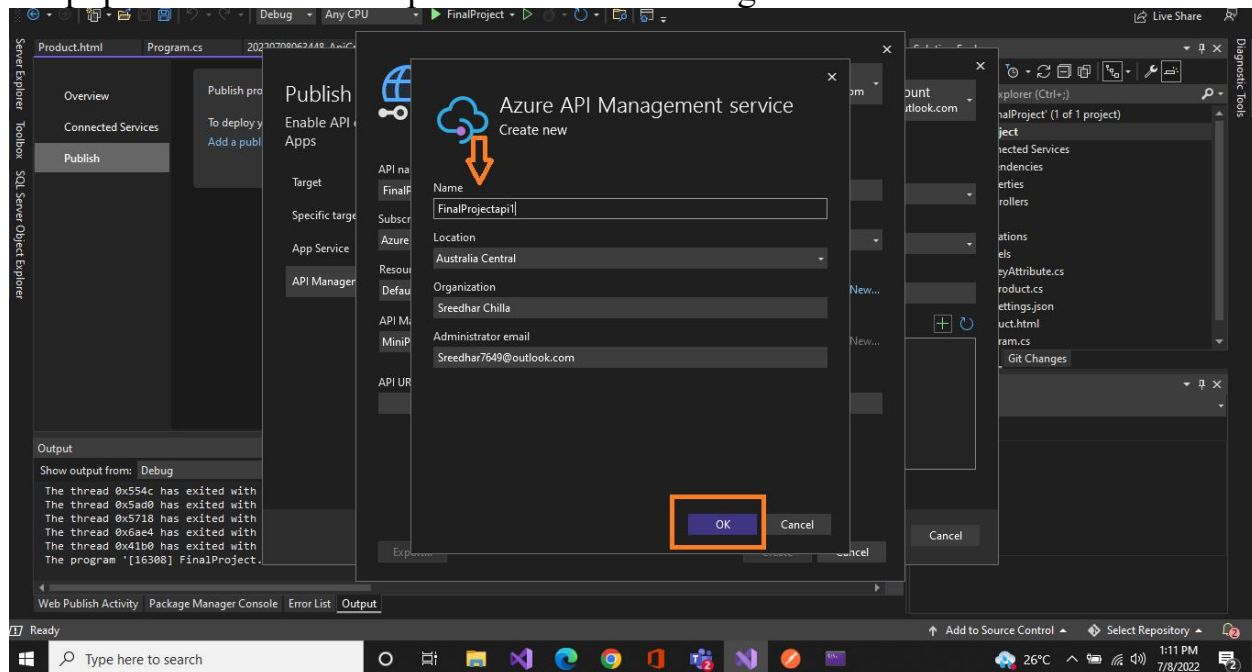
- ✧ After creation is completed, the dialog is automatically closed and the Publish dialog gets focus again. The instance that was created is automatically selected.



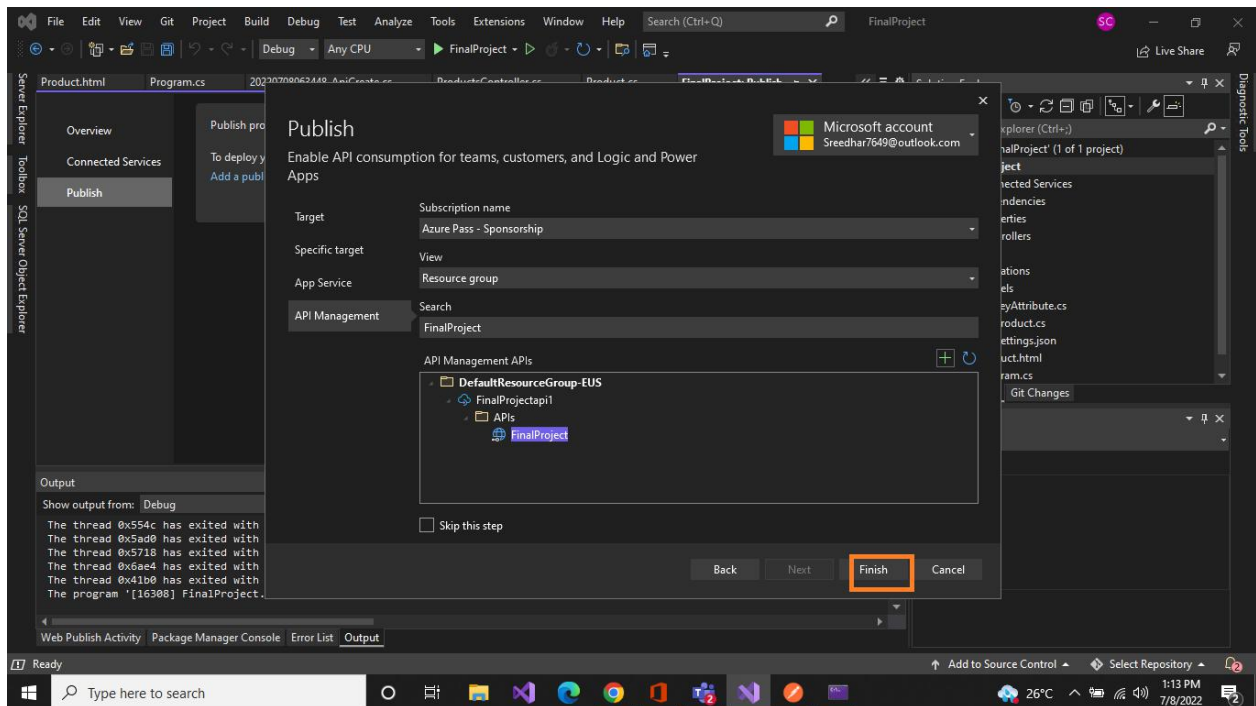
✧ The dialog now shows the Azure API Management service to create.



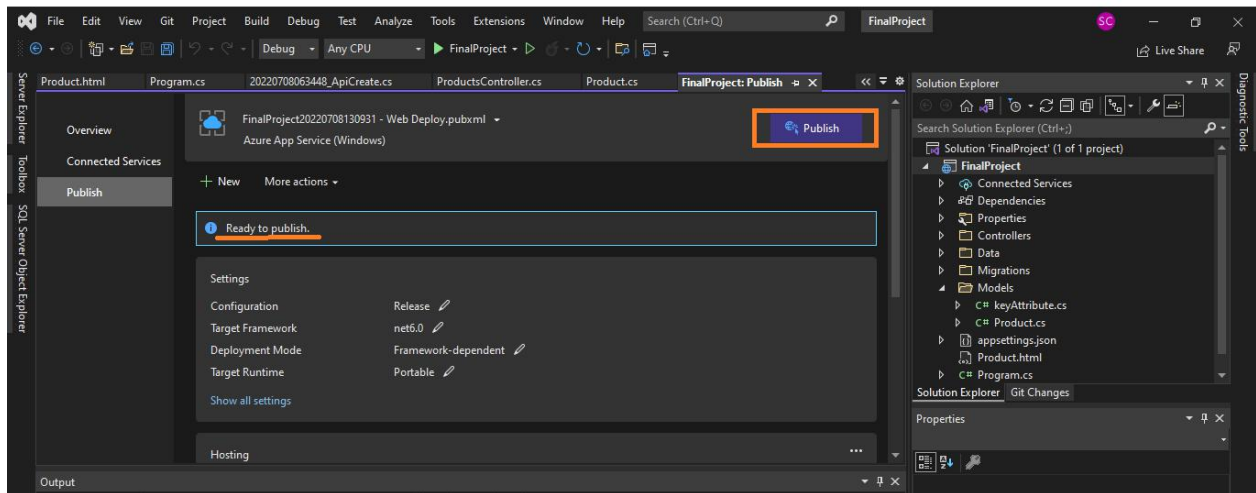
- ✧ The Create the Azure API Management service dialog appears. The App Name, Resource Group, and API Management service entry fields are populated. You can keep these names or change them. Select the Create button.



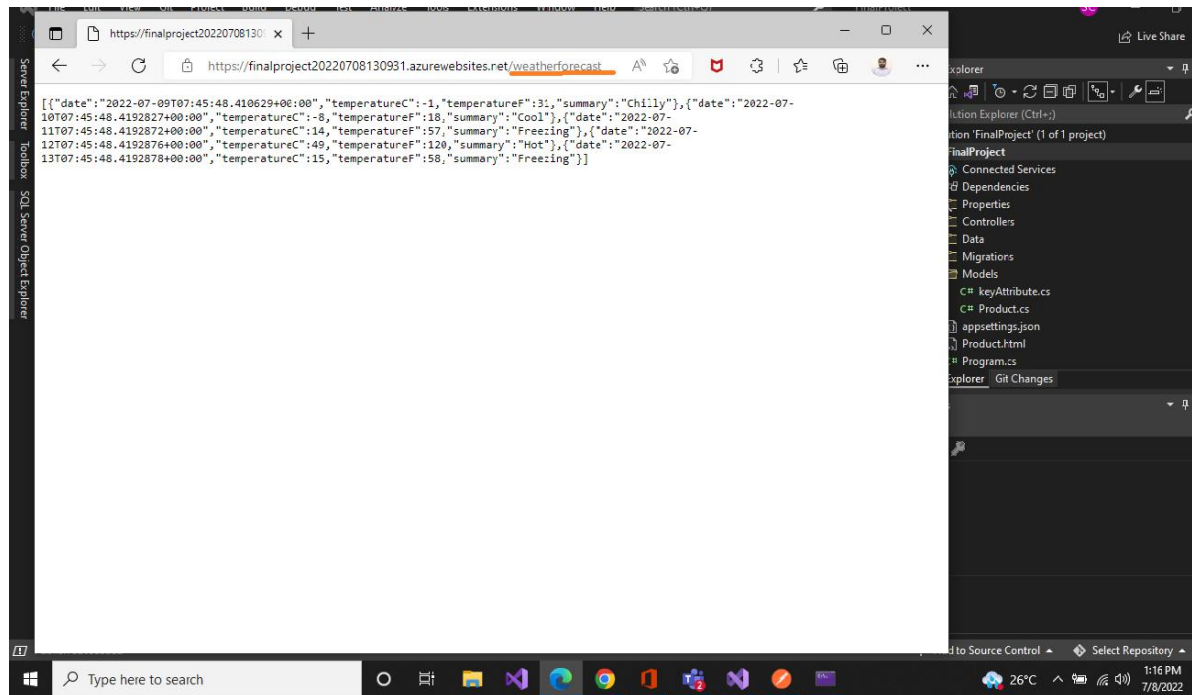
- ✧ After creation is completed, the dialog is automatically closed and the Publish dialog gets focus again. The instance that was created is automatically selected.



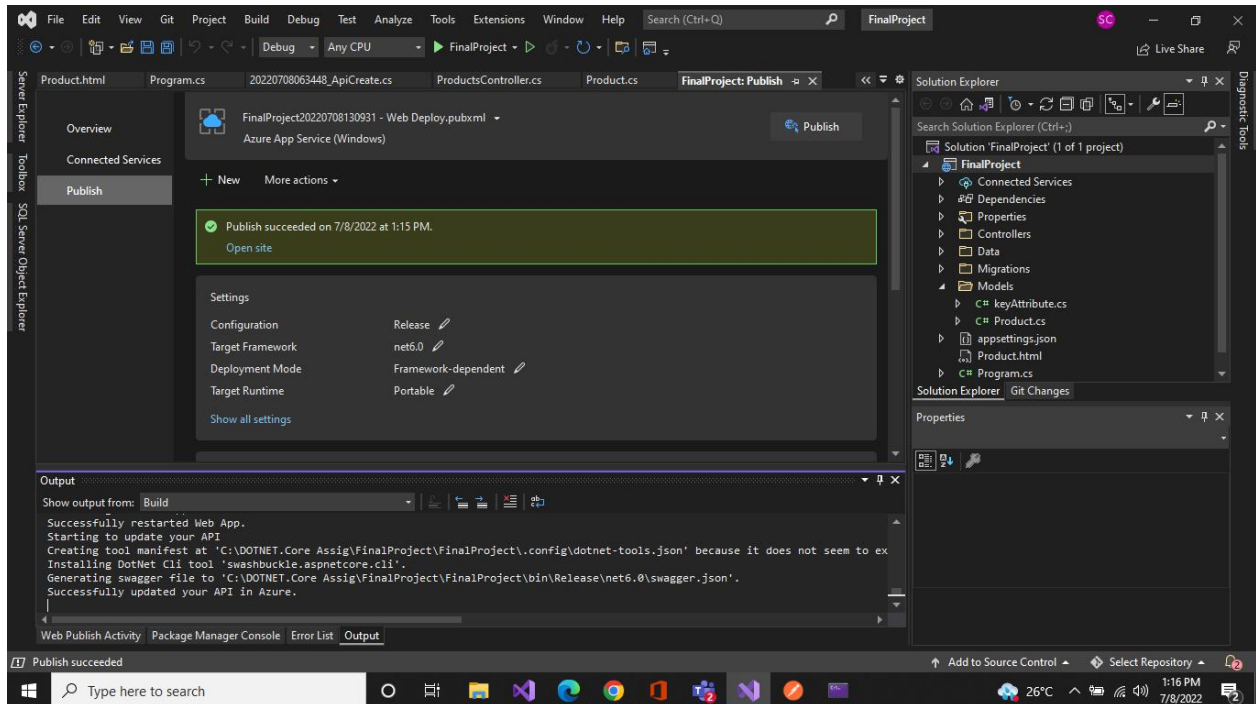
- ✧ The dialog closes and a summary screen appears with information about the publish. Select the Publish button.



- ✧ Select the Publish button after that it will Publishing to Azure App Service(Window) and checking your application will run Successfully..
- ✧ 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. You can close that window.

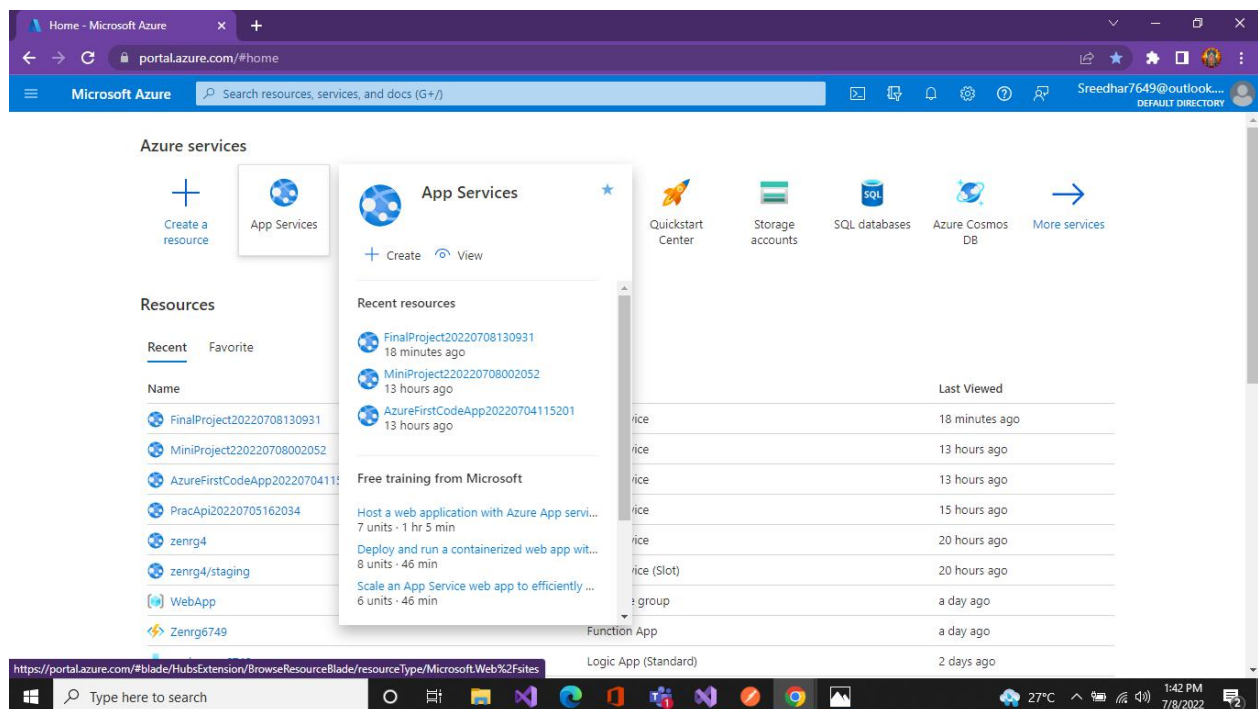


✧ Select the Publish button on site to open.

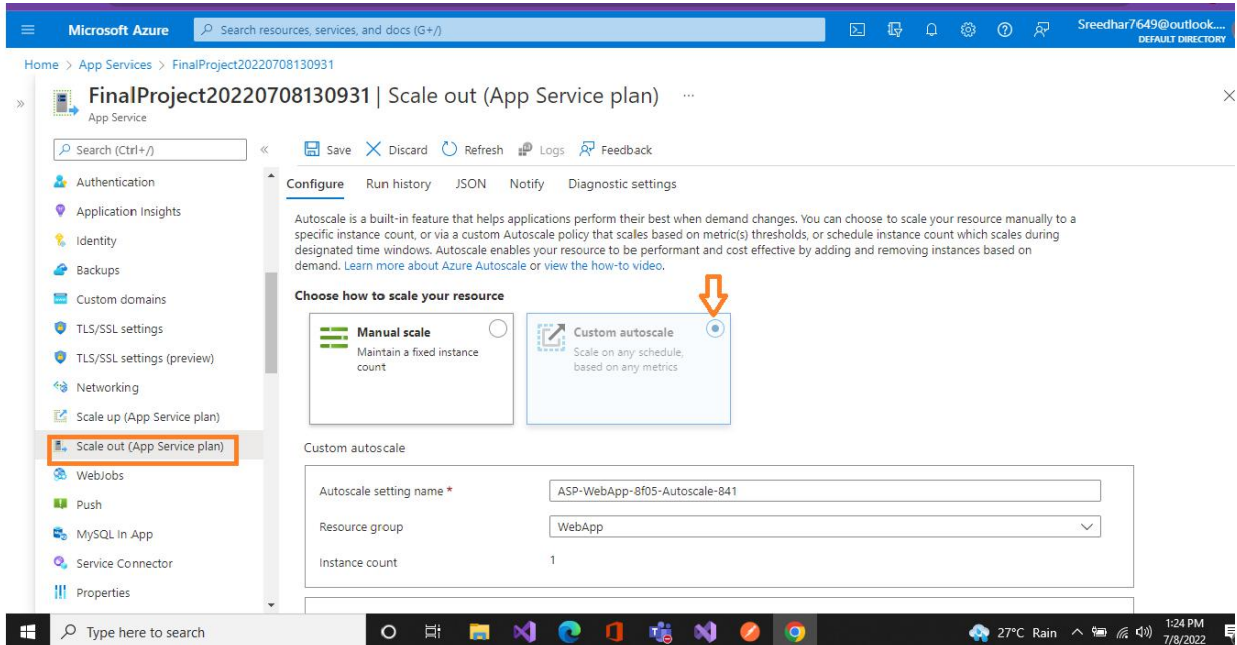


2. Configure Scale out by adding rules for custom scaling

- ✧ 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.

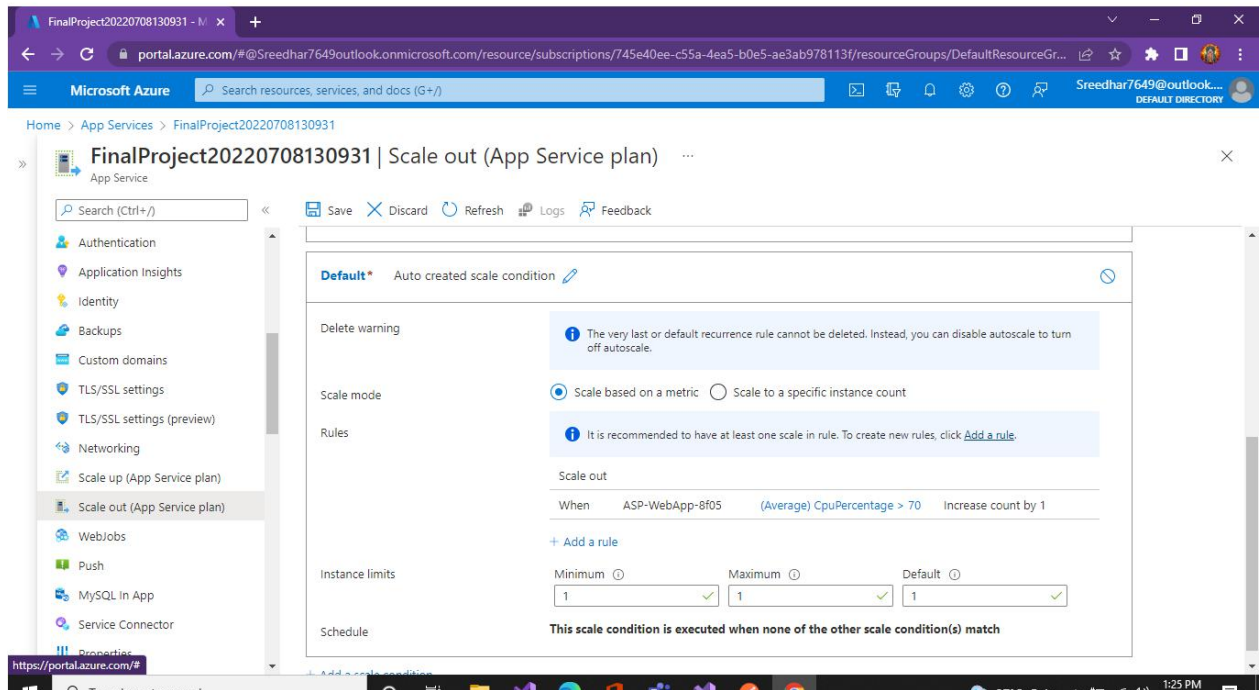
The screenshot displays the Microsoft Azure portal interface for configuring a scale rule. The main pane shows the 'Scale out (App Service plan)' configuration for the resource 'FinalProject20220708130931'. The 'Scale rule' panel on the right is open, showing the configuration for the 'Default' auto-created scale condition. The 'Metric source' is set to 'Current resource (ASP-WebApp-8f05)'. The 'Resource type' is 'App Service plans' and the 'Resource' is 'ASP-WebApp-8f05'. The 'Criteria' section shows 'Time aggregation' as 'Average', 'Metric namespace' as 'App Service plans standard metrics', and 'Metric name' as 'CPU Percentage'. The 'Dimension Name' is 'Instance', the 'Operator' is 'Greater than', and the 'Dimension Values' are 'All values'. The 'Instance limits' section shows a 'Minimum' of 1. The 'Add' button is highlighted with an orange box.

- ✧ 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 5 minutes 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.

The screenshot displays the Microsoft Azure portal interface for configuring a scale rule for an App Service plan. The left sidebar shows the navigation menu with options like Authentication, Application Insights, Identity, Backups, Custom domains, TLS/SSL settings, Networking, Scale up (App Service plan), Scale out (App Service plan), WebJobs, Push, MySQL In App, Service Connector, and Properties. The main content area is titled 'FinalProject20220708130931 | Scale out (App Service plan)'. The 'Scale rule' configuration panel is open, showing a graph of CPU percentage over time. The configuration details are as follows:

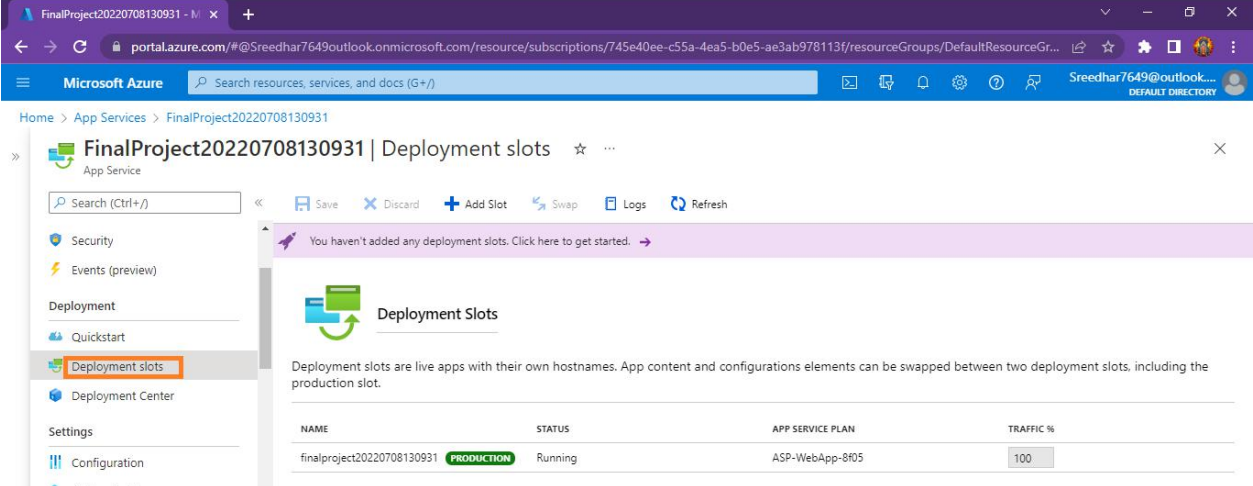
- Resource group:** WebApp
- Instance count:** 1
- Default*** Auto created scale condition
- Delete warning:** The very last or default rule will turn off autoscale.
- Scale mode:** Scale based on a metric
- Rules:** Scale is based on metric trigger. Example: Add a rule that triggers scaling when the resource is scaled.
- Instance limits:** Minimum 1
- Schedule:** This scale condition is executed
- Scale rule graph:** Shows CPU percentage (Average) over time, with a peak around 1:20 PM.
- Configuration details:**
 - Operator:** Greater than
 - Metric threshold to trigger scale action:** 70%
 - Duration (minutes):** 10
 - Time grain (minutes):** 1
 - Time grain statistic:** Average
 - Action:** Increase count by 1
 - Cool down (minutes):** 5
- Add button:** Highlighted in orange.

✧ After adding the scale it show Rules.



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 screenshot shows the Azure Portal interface for a Function App named 'FinalProject20220708130931'. The left-hand navigation pane is open, and the 'Deployment slots' option is highlighted with an orange box. The main content area displays the 'Deployment slots' configuration page. At the top, there is a message: 'You haven't added any deployment slots. Click here to get started.' Below this, there is a section titled 'Deployment Slots' with a brief description: 'Deployment slots are live apps with their own hostnames. App content and configurations elements can be swapped between two deployment slots, including the production slot.' A table lists the current deployment slots:

NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
finalproject20220708130931	PRODUCTION Running	ASP-WebApp-8R05	100

✧ Add the Slot and Name as Staging and then Add the Slot .

The screenshot shows the Microsoft Azure portal interface. The main content area displays the 'Deployment slots' page for the application 'FinalProject20220708130931'. A table lists the current deployment slots:

NAME	STATUS
finalproject20220708130931	PRODUCTION Running

An 'Add a slot' dialog box is open on the right side of the screen. It contains the following fields and controls:

- Name:** A text input field containing the value 'Stag'. An orange arrow points to this field.
- Clone settings from:** A dropdown menu with the selected option 'Do not clone settings'.
- Buttons:** 'Add' and 'Close' buttons are located at the bottom of the dialog box. The 'Add' button is highlighted with an orange box.

The background page also shows a left-hand navigation pane with various settings options, and a top toolbar with actions like 'Save', 'Discard', 'Add Slot', 'Swap', 'Logs', and 'Refresh'. The 'Add Slot' button in the toolbar is also highlighted with an orange box.

✧ After Add Slot Successful both the Production and Staging is Displayed .

The screenshot shows the Azure portal interface for an App Service named 'FinalProject20220708130931'. The 'Deployment slots' page is active, showing a list of deployment slots. The left sidebar contains navigation options like Security, Events, Deployment, Quickstart, Deployment slots, Deployment Center, and Settings. The main content area shows a table of deployment slots with columns for NAME, STATUS, APP SERVICE PLAN, and TRAFFIC %.

NAME	STATUS	APP SERVICE PLAN	TRAFFIC %
finalproject20220708130931	PRODUCTION Running	ASP-WebApp-8f05	100
finalproject20220708130931-Stag	Running	ASP-WebApp-8f05	0

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

The screenshot shows the Microsoft Azure portal interface. On the left, the 'Deployment slots' option is highlighted in the sidebar. The main pane displays the 'Deployment Slots' section for the app 'FinalProject20220708130931'. It lists two slots: 'finalproject20220708130931' (Production) and 'finalproject20220708130931-Stag' (Running). On the right, the 'Swap' dialog is open. It shows the 'Source' slot as 'finalproject20220708130931-Stag' and the 'Target' slot as 'finalproject20220708130931' (Production). Below this, there's a 'Config Changes' section with a table showing the 'NetFrameworkVersion' changing from 'v4.0' to 'v6.0'. At the bottom of the dialog, the 'Swap' button is highlighted with an orange box.

Deployment Slots

Deployment slots are live apps with their own hostnames. App content and production slot.

NAME	STATUS
finalproject20220708130931	PRODUCTION Running
finalproject20220708130931-Stag	Running

Swap

Source: finalproject20220708130931-Stag

Target: PRODUCTION finalproject20220708130931

Swap with preview can only be used with sites that have deployment slot settings enabled

Perform swap with preview

Config Changes

This is a summary of the final set of configuration changes on the source and target deployment slots after the swap has completed.

Source Changes		Target Changes	
SETTING	TYPE	OLD VALUE	NEW VALUE
NetFrameworkVersion	General	v4.0	v6.0

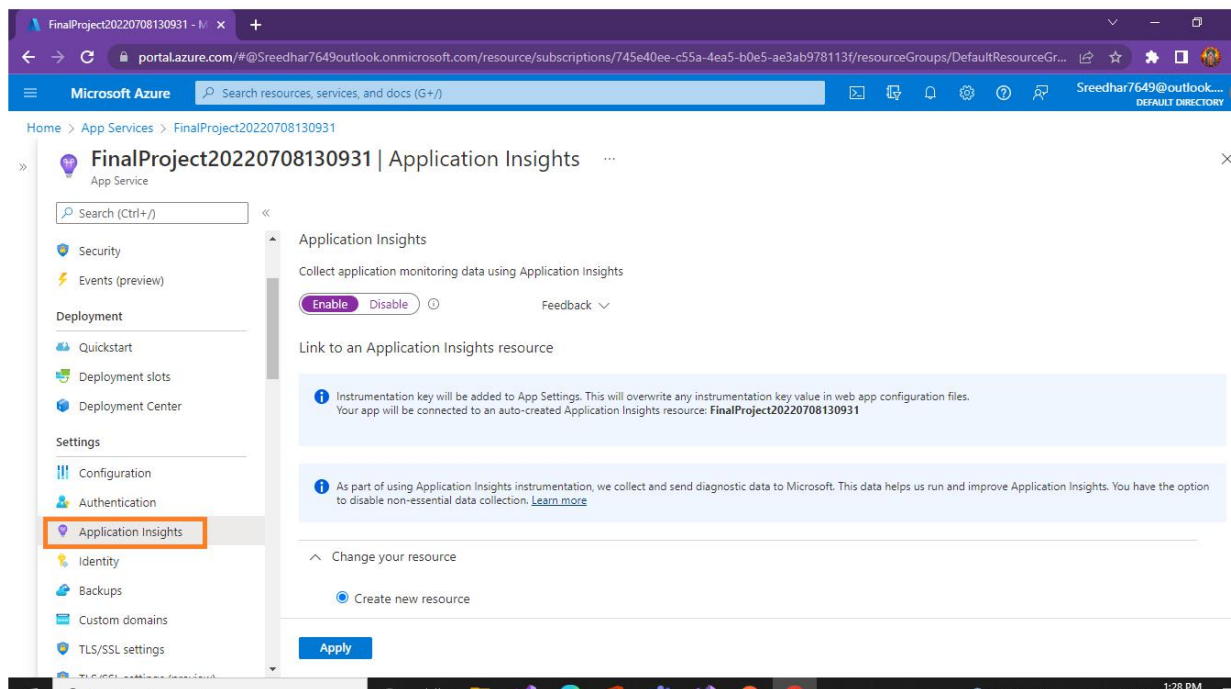
Swap Close

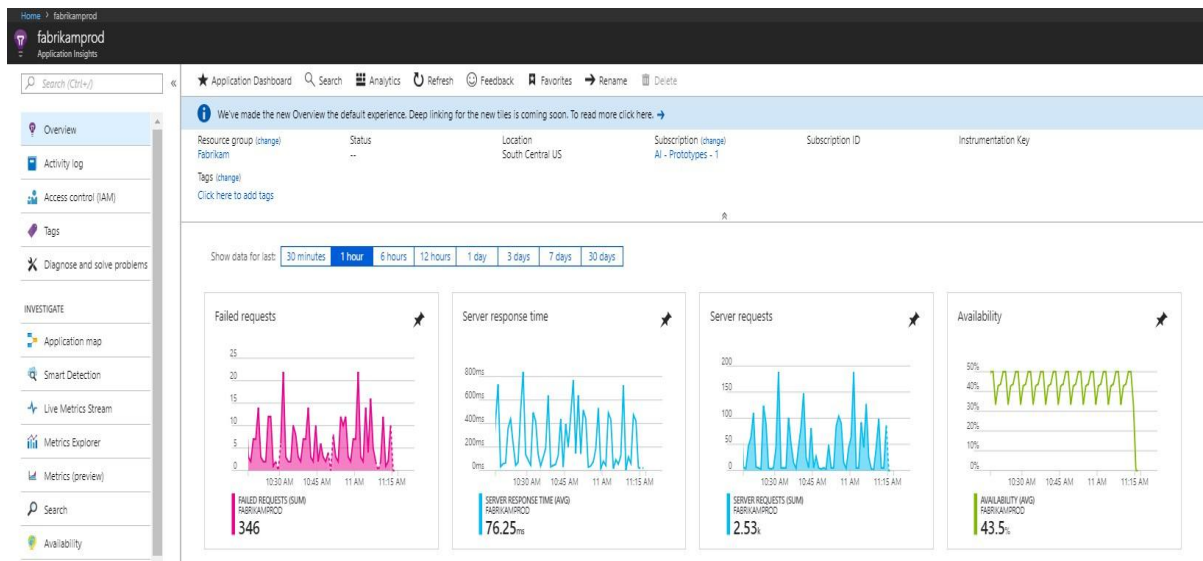
4. Configure Application Insights for the project

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

✧ Select the Application Insights





Type=ApplicationInsights ApplicationName=fabrikamprod TelemetryType = Availability

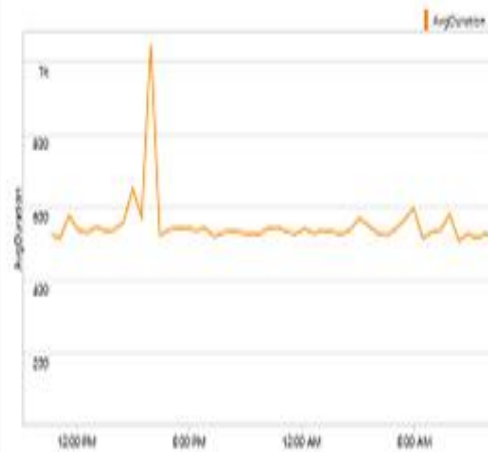
7K Results [List](#) [Table](#) [Application Insights](#)

Failed webtests

NAME	COUNT
DashTest21	5K
Fabrikam Homepage	2K

[See all...](#)

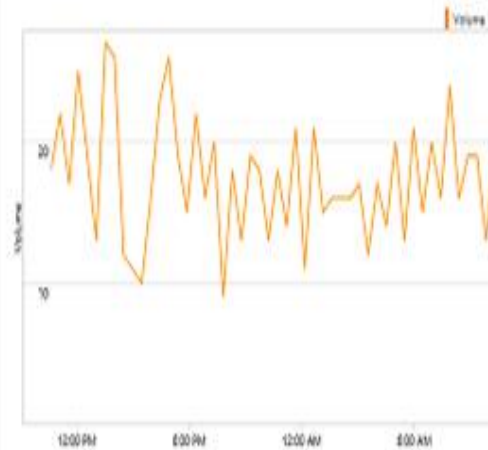
Test duration



Webtest results



Failed requests

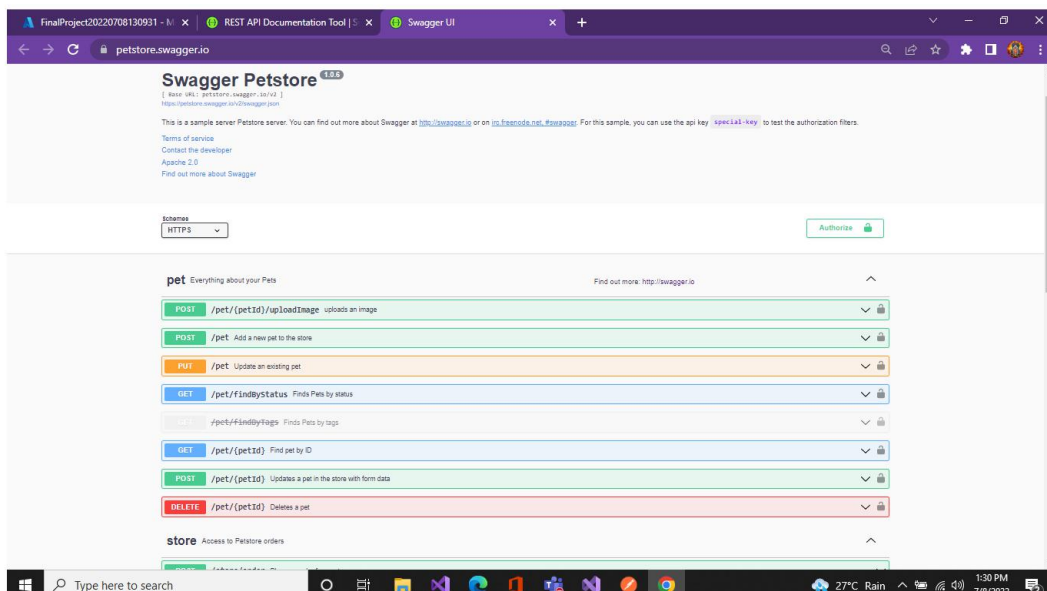


5. Configure Swagger for the API

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 back end implementation and client side consumption.

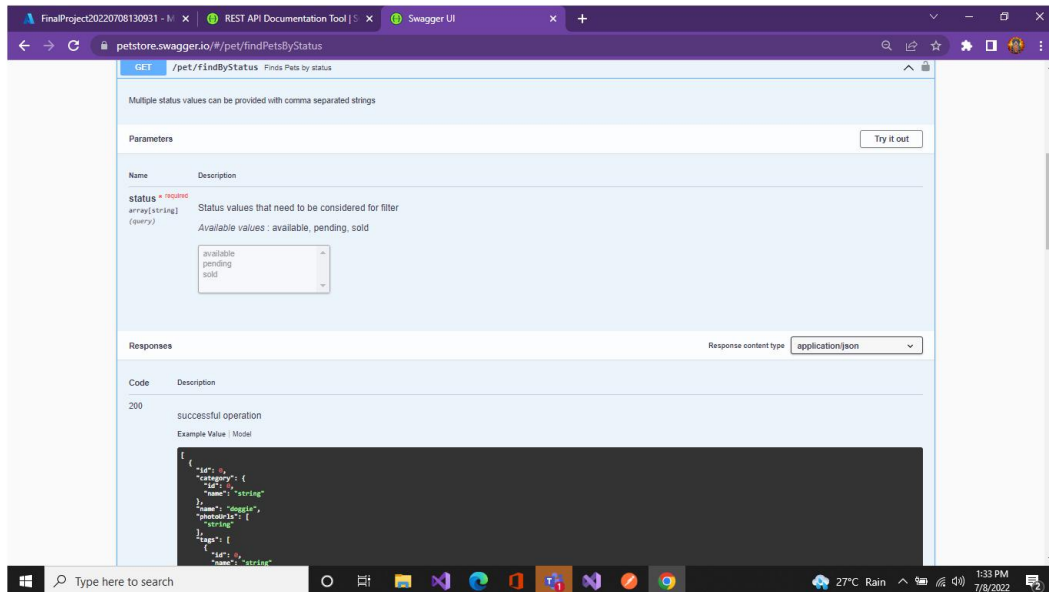
Advantages :-

- **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



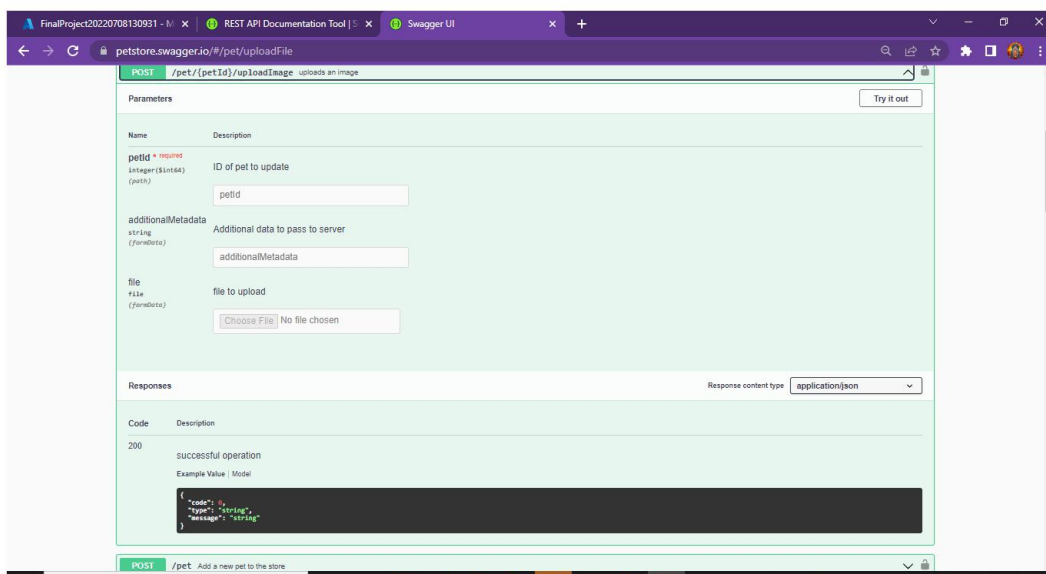
This image is showing swagger documentation for the created Products Web API project.

GET:-



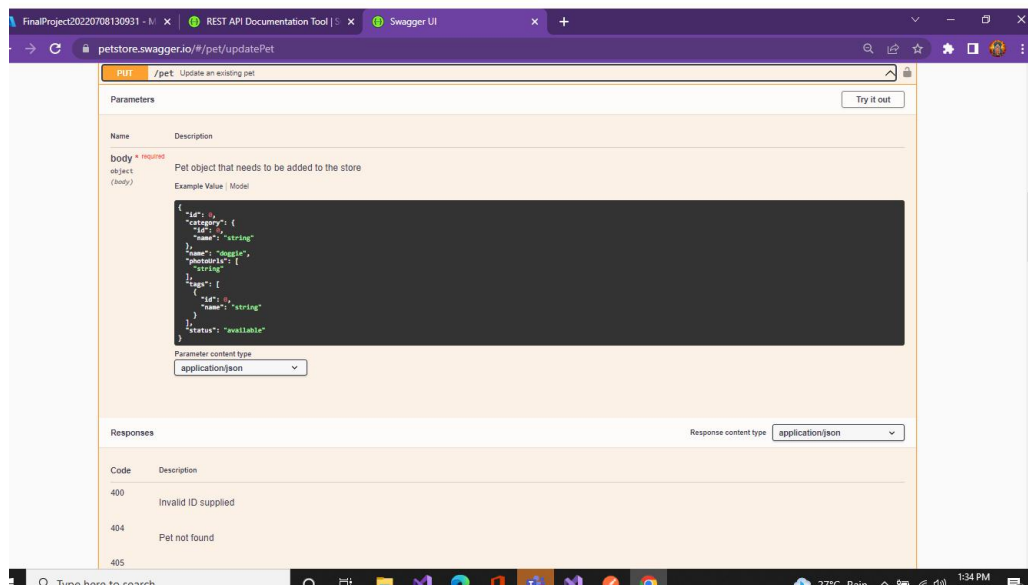
This Image showing the get call for a product with specified product id passed as a path variable.

POST:-



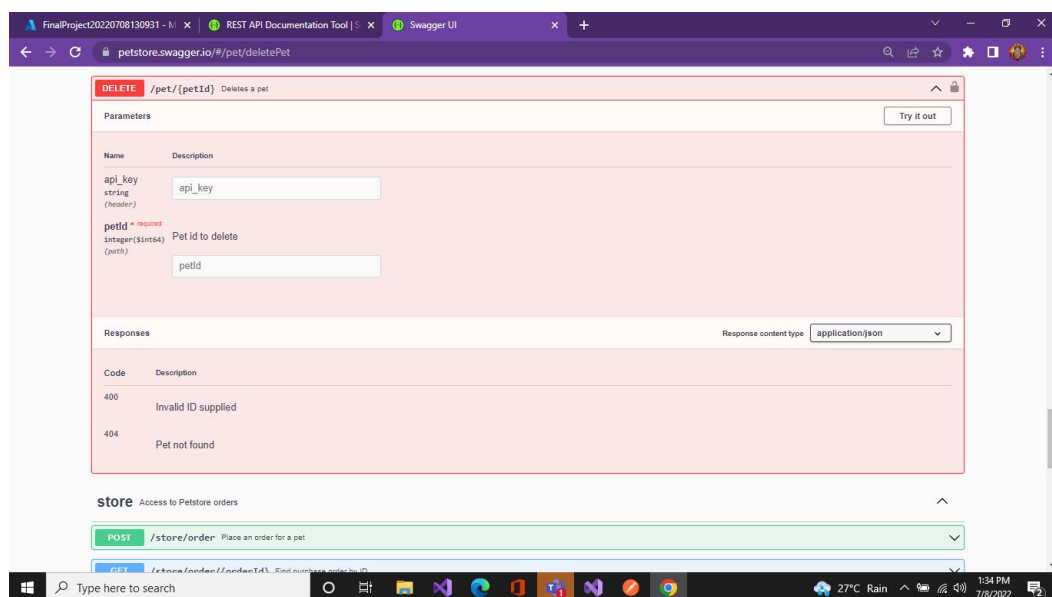
This Image showing the post call for creating a new product.

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.

DELETE:-

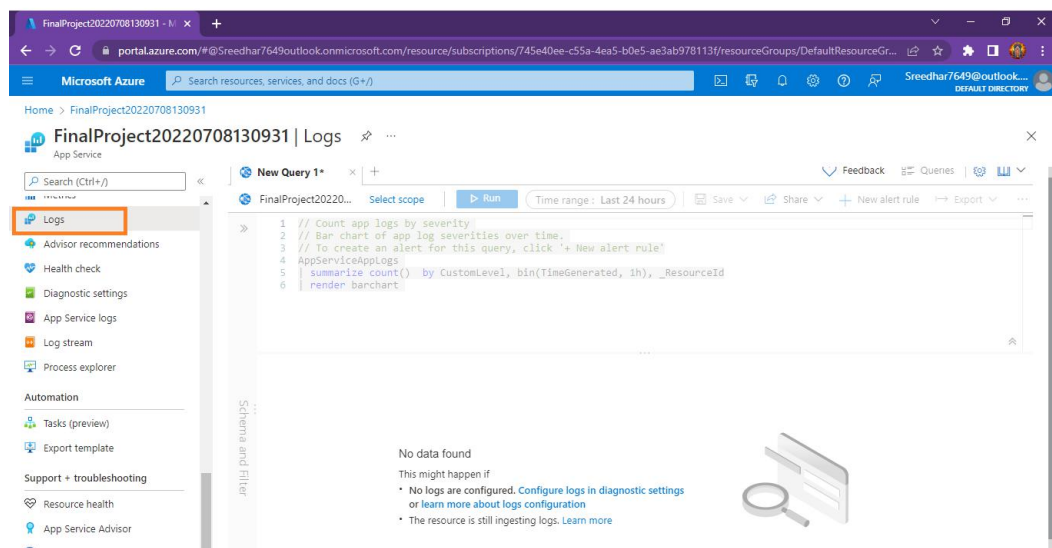


This Image showing the delete call for deleting an existing product with the specified id passed as a path variable.

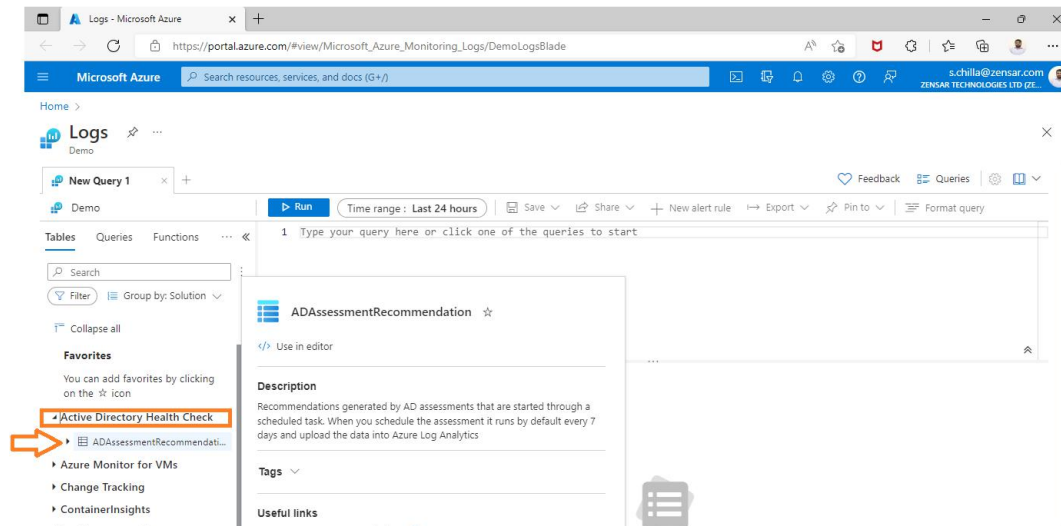
6. Work with Log Analytics with the sample logs available

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

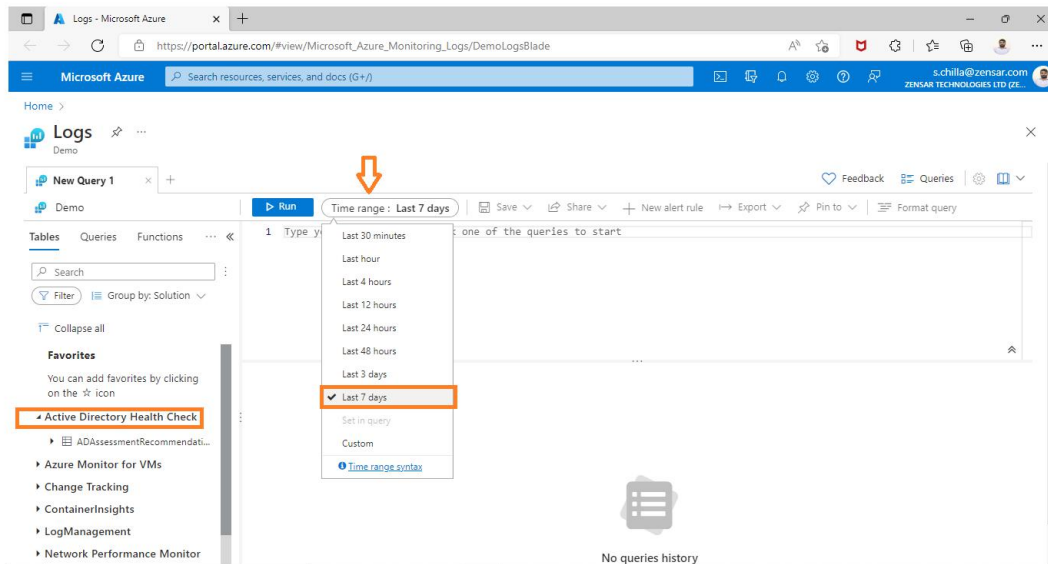
✧ Select the Logs in Azure Portal.



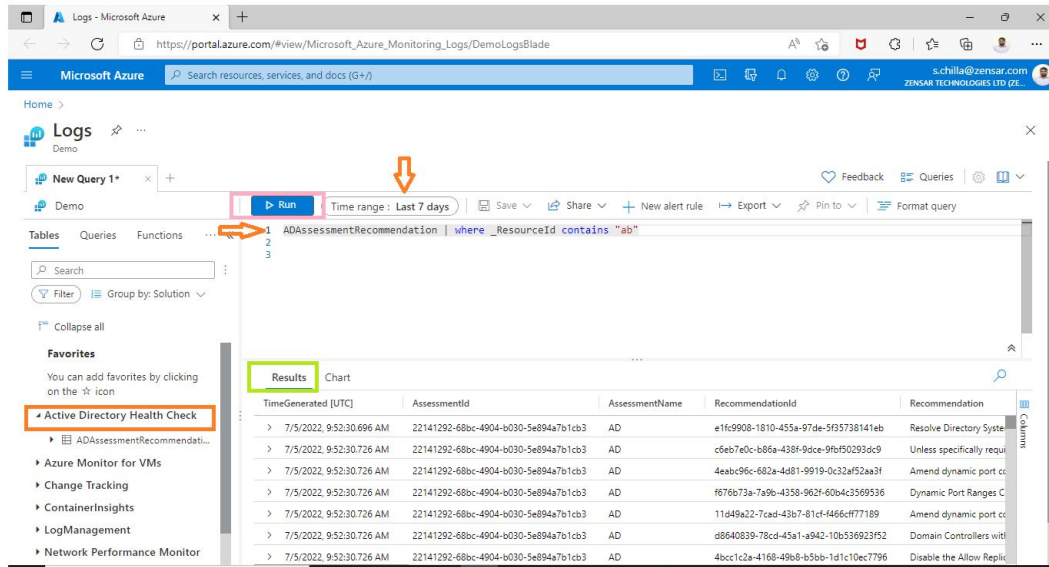
- ✧ Select Logs from the Azure Monitor menu . This step sets the initial scope to a Log Analytics workspace so that your query selects from all data in that workspace .



- ✧ 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 timerange 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 12 hours 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 in 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.



The screenshot shows the Microsoft Azure portal interface. In the 'Logs' section, a query is being run. The 'Run' button is highlighted with an orange arrow. The query is 'ADAssessmentRecommendation | where _ResourceId contains "ab"'. The 'Results' tab is selected, displaying a table of assessment recommendations.

TimeGenerated [UTC]	AssessmentId	AssessmentName	RecommendationId	Recommendation
> 7/5/2022, 9:52:30.696 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	e1fc9908-1810-455a-97de-5f5738141eb	Resolve Directory System
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	c6eb7e0c-b86a-438f-9dce-9bf50293dc9	Unless specifically requi
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	4eabc96c-682a-4d81-9919-0c32af52aa3f	Amend dynamic port co
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	f676b73a-7a9b-4358-962f-60b4c3569536	Dynamic Port Ranges C
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	11d49a22-7cad-43b7-81cf-4466cf77189	Amend dynamic port co
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	d8640839-78cd-45a1-a942-10b536923f52	Domain Controllers wit
> 7/5/2022, 9:52:30.726 AM	22141292-68bc-4904-b030-5e894a7b1cb3	AD	4bcc1c2a-4168-49b8-b5bb-1d1c10ec7796	Disable the Allow Repli