😄 Azure Traffic Manager

Azure Traffic Manager is a **DNS-based traffic load balancer** that helps distribute incoming network traffic across multiple services hosted in different Azure regions or external locations. It improves application availability, performance, and resilience by directing users to the most appropriate endpoint based on different routing methods.

Key Features of Azure Traffic Manager:

1. **Global Load Balancing:** Distributes traffic across multiple Azure regions or external endpoints.
2. **Routing Methods:** Supports multiple traffic routing methods, including:
    - **Priority Routing:** Sends traffic to a primary endpoint and fails over to backups if needed.
    - **Weighted Routing:** Distributes traffic based on predefined weight values.
    - **Performance Routing:** Directs users to the closest and fastest available endpoint.
    - **Geographic Routing:** Routes traffic based on the user's geographic location.
    - **Multivalue Routing:** Returns multiple healthy endpoints to DNS queries.
    - **Subnet Routing:** Routes traffic based on the user's IP address range.
3. **Health Monitoring:** Continuously checks endpoint health and automatically redirects traffic if an endpoint is down.
4. **Azure and Non-Azure Support:** Works with both Azure and external (on-premises or other cloud) endpoints.
5. **Failover and High Availability:** Ensures application availability by redirecting traffic when an endpoint fails.
6. **Integration with Azure Services:** Works seamlessly with Azure App Services, Virtual Machines, and other cloud services.

Use Cases:

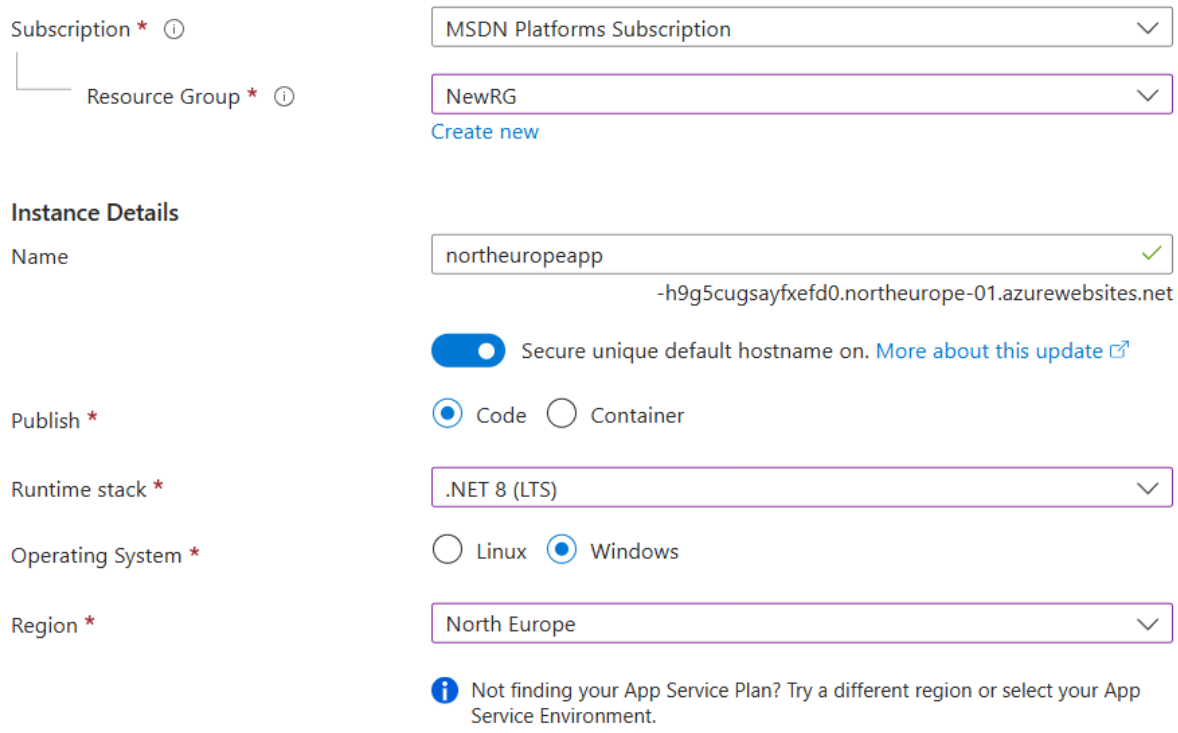
- Disaster recovery by automatically failing over to backup regions.
- Improving performance by directing users to the nearest available region.
- Distributing traffic efficiently among multiple cloud resources.
- Supporting multi-cloud or hybrid cloud architectures.

Since it's **DNS-based**, it does not handle traffic directly but instead resolves the best endpoint at the DNS level. For real-time traffic management, you might need **Azure Front Door** or **Azure Load Balancer**.

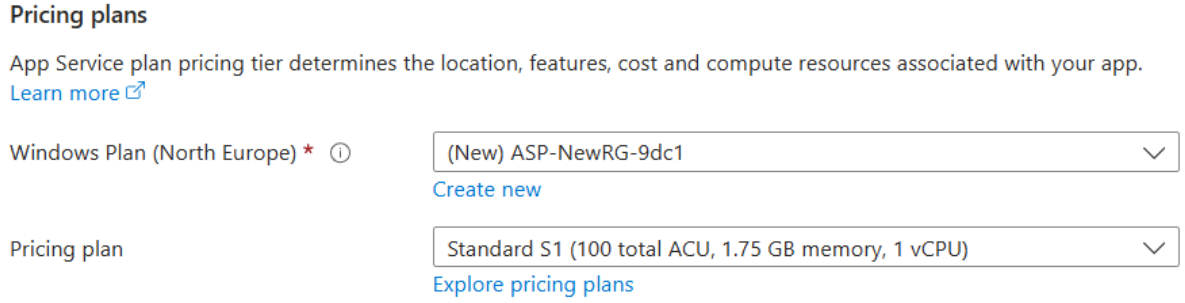
The process involves creating two Azure Web Apps in North Europe and UK South, both running .NET 8 with a Standard S1 pricing plan. After deploying them, we edit their default.html files. Then, we create an Azure Traffic Manager profile using the Priority routing method. We add both web apps as endpoints, configuring HTTPS and custom headers. Traffic initially flows to North Europe, but if it goes offline, Traffic Manager redirects users to UK South. The end goal is to implement high availability and failover using Azure Traffic Manager. Finally, we delete all resources after testing.

😄 To begin with the Lab

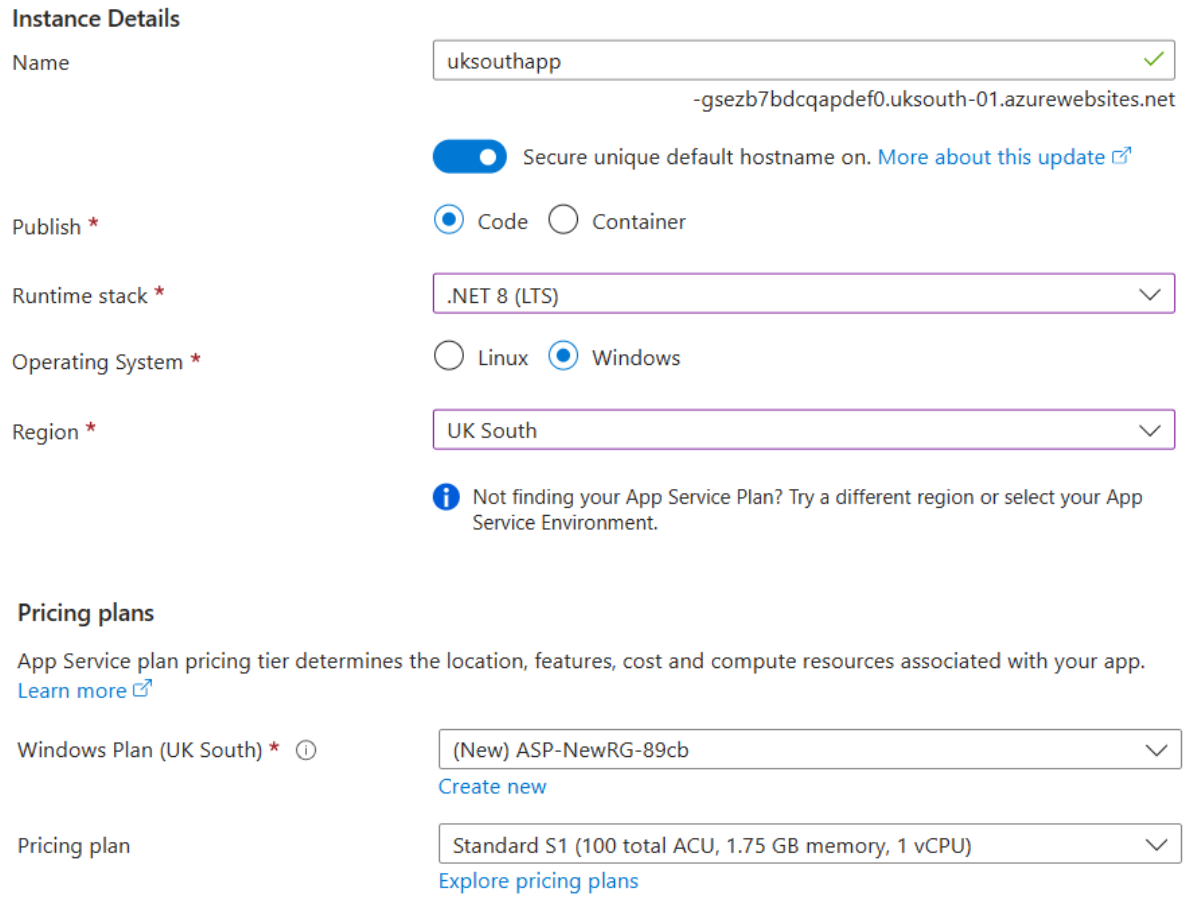
1. In this lab, we will start by creating two web apps in two different regions.
2. From the marketplace place create a web app, choose your resource group, give it a name then choose dot net 8 as your runtimes stack. For region choose North Europe.



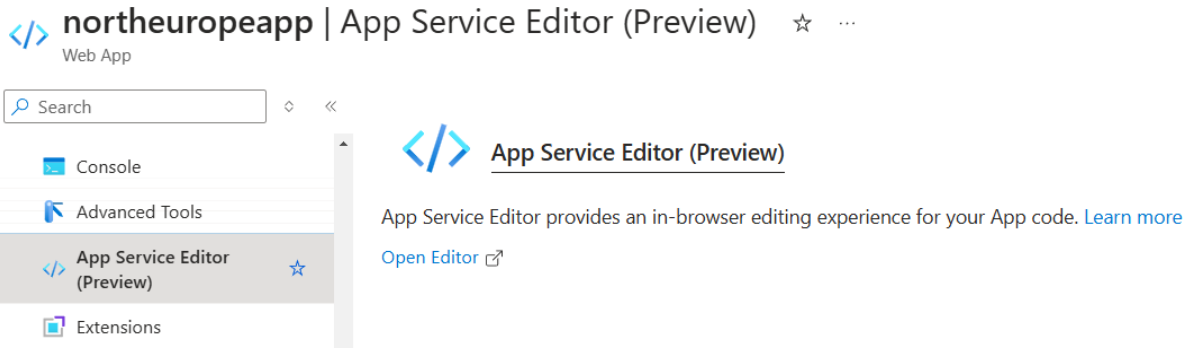
3. Then for the pricing plan you have to choose Standard S1 to use the traffic manager.



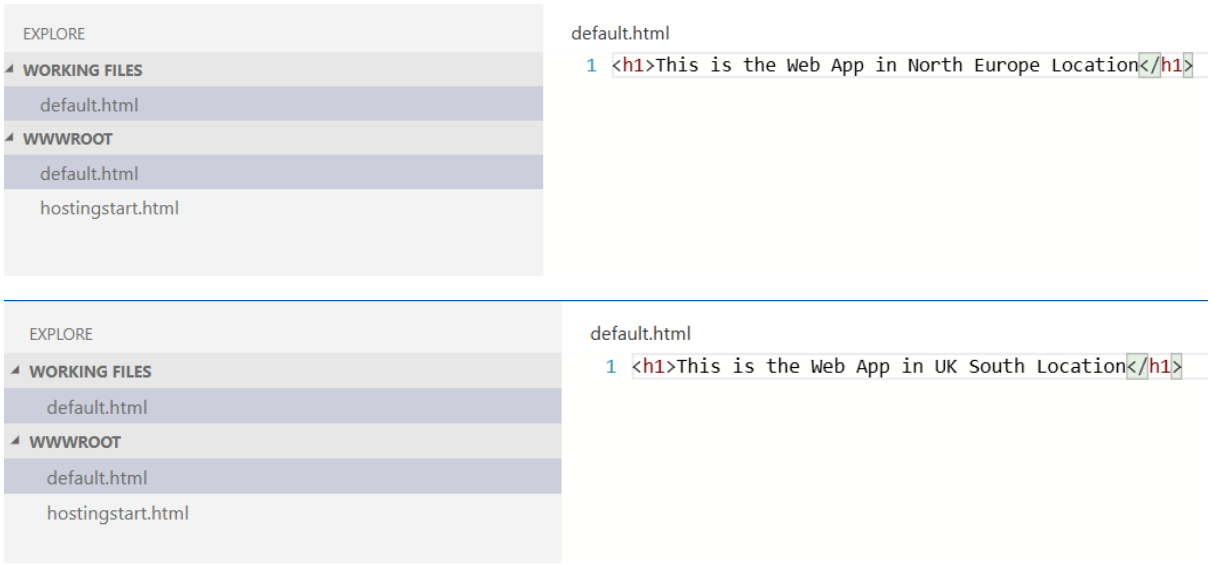
4. Now move to the review page and create your first web app. Since our web app is deploying, we can go ahead and create another web app in the UK South location with the same configurations.



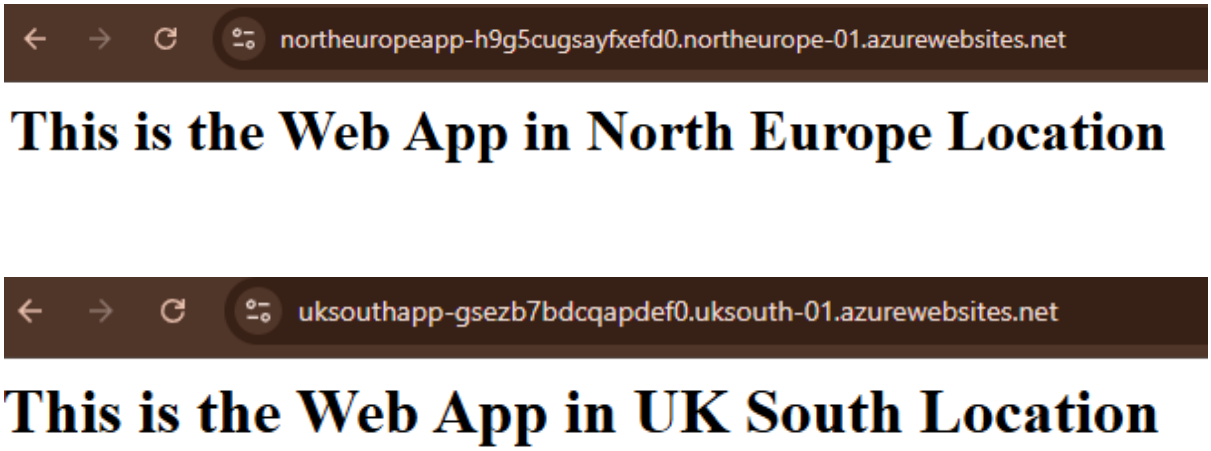
5. Our web app from the North Europe location has been created. Open it and look for App Service Editor. Click on Open Editor.



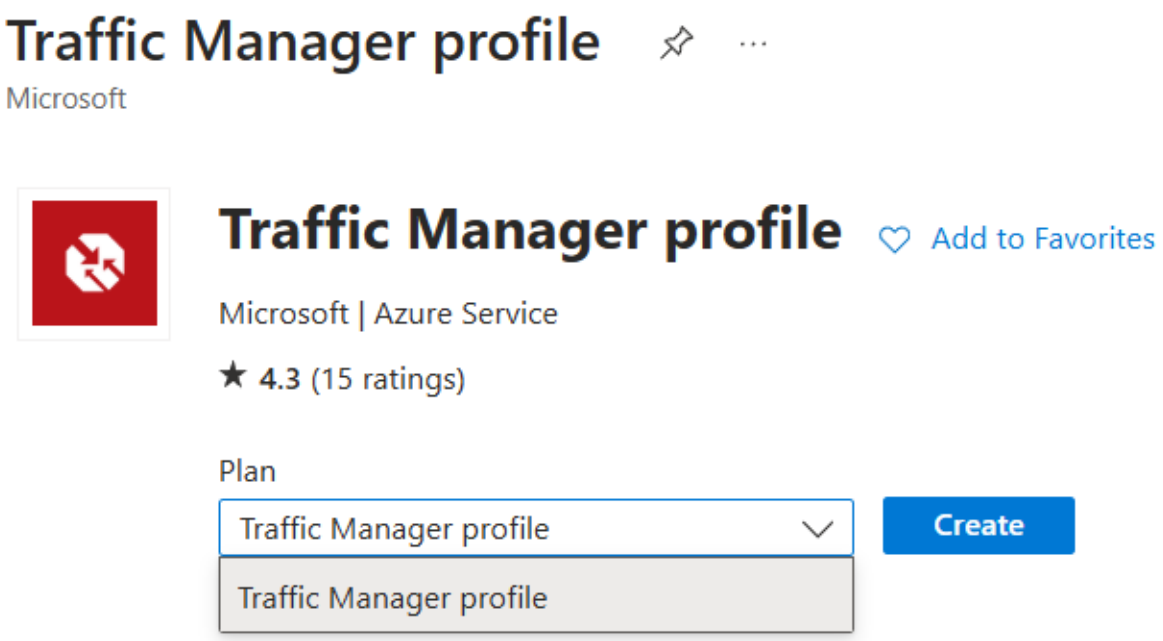
6. Here in the WWWROOT we created a file with the name default.html and then we wrote this text message. We did the same thing for the other web app.



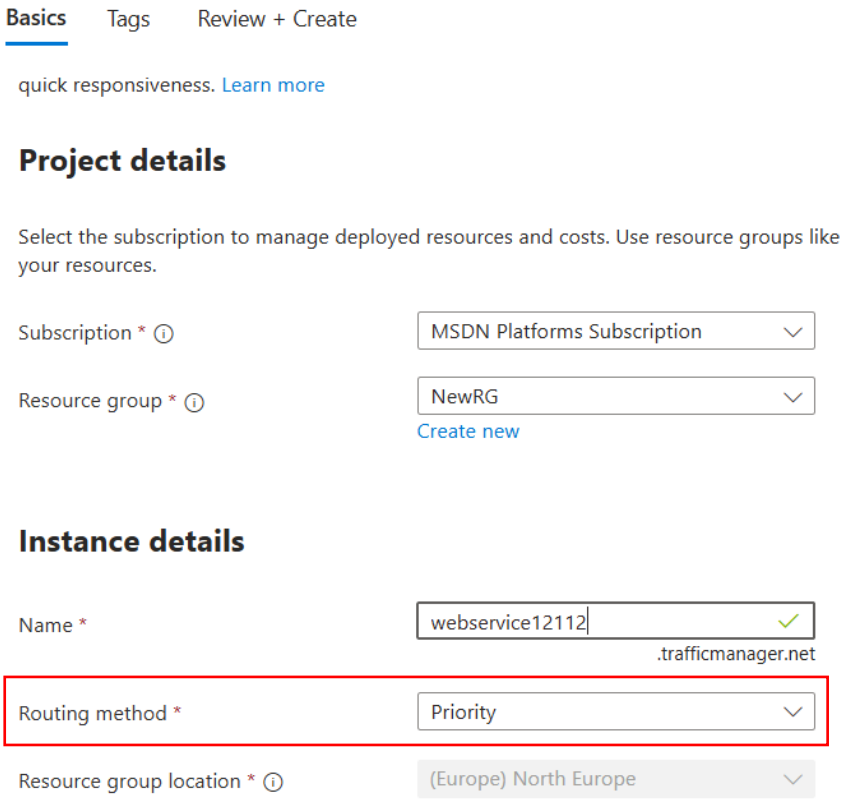
7. Below you can see that both of our web apps are online now.



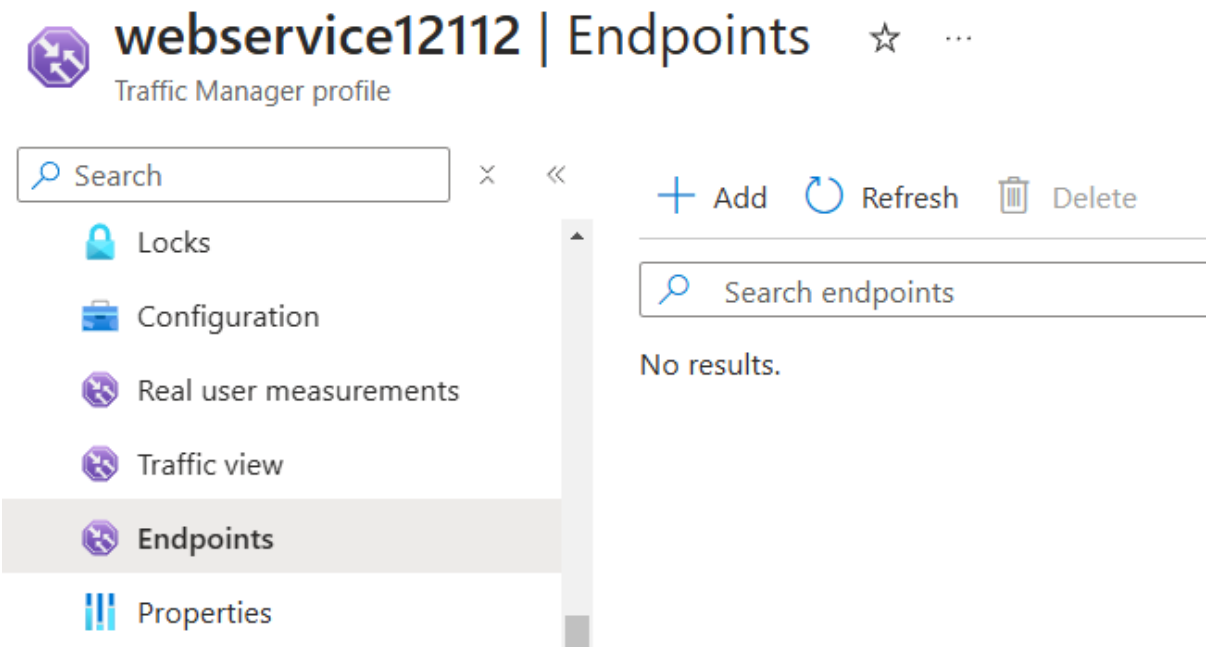
8. Now from the marketplace search for the traffic manager profile and click on create.



9. Here you need to choose your resource group and then give a unique to your traffic manager. Then choose Priority for your Routing Method and create your traffic manager.



10. Now we are going to the endpoints to create them. Click on Add.



11. To add an endpoint, we have to choose type as azure endpoint and then give it a name. In the Target resource type choose App Service and in the target, resource choose your first web app.
12. In the custom header settings, we need to specify the host, so here we are just writing the default domain name of our web app. We will do the same thing for other web app as well.

    host:northeuropeapp-h9g5cugsayfxefd0.northeurope-01.azurewebsites.net

Add endpoint

webservice12112

Type * ⓘ — Azure endpoint

Name * — north-eu-endpoint ✓

Enable Endpoint * — ☑

Target resource type * — App Service

Target resource * — northeuropeapp (North Europe)

Priority * — 1 ✓

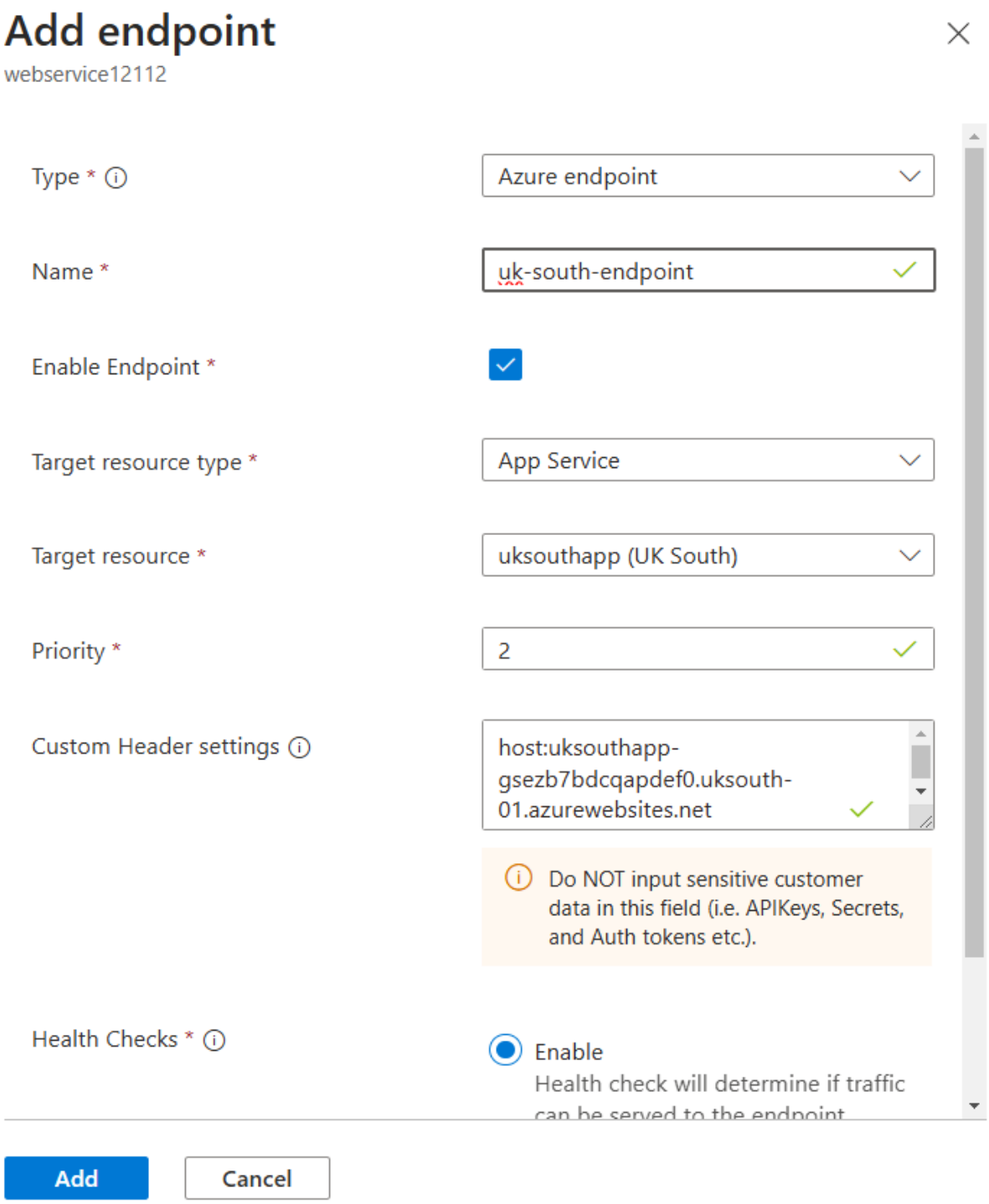
Custom Header settings ⓘ — host:northeuropeapp-h9g5cugsayfxefd0.northeurope-01.azurewebsites.net ✓

ⓘ Do NOT input sensitive customer data in this field (i.e. APIKeys, Secrets, and Auth tokens etc.).

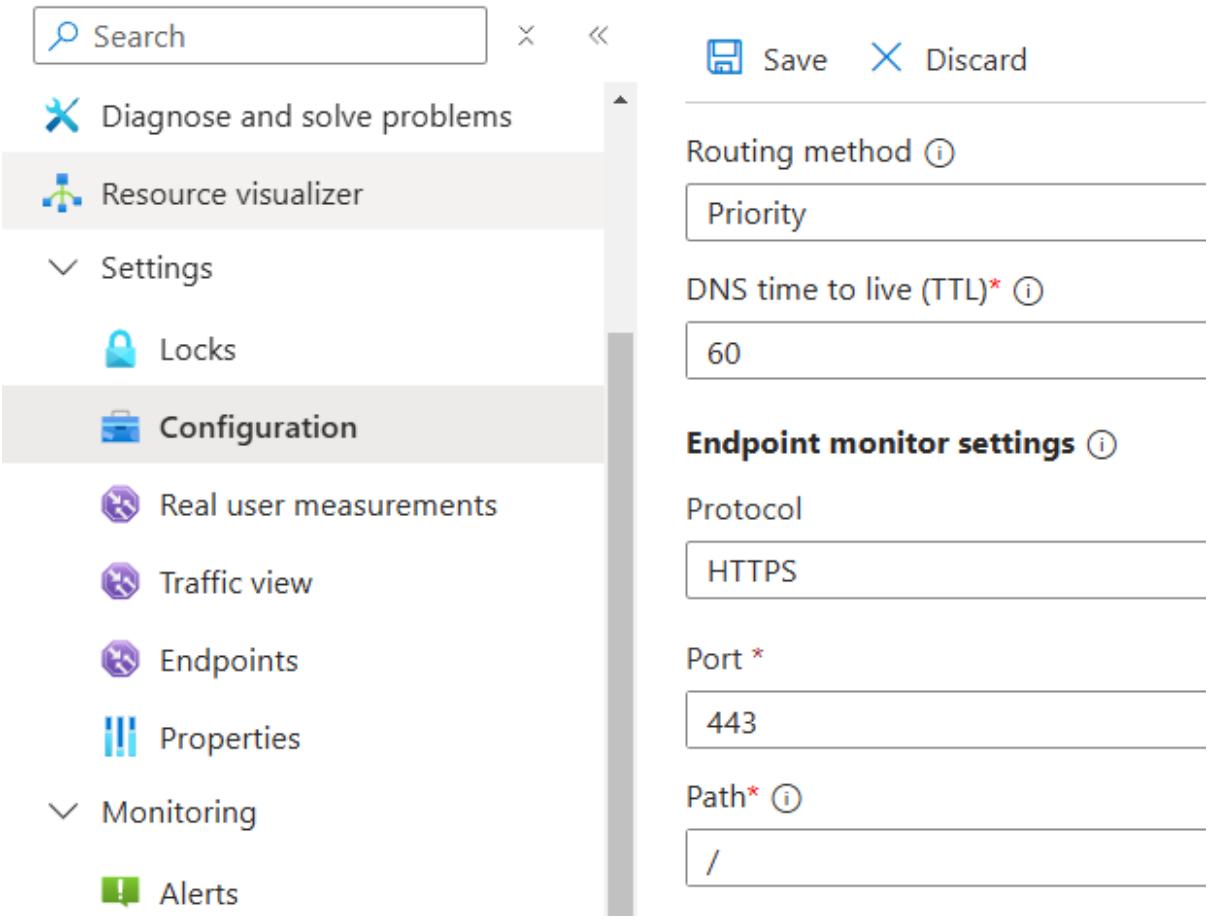
Health Checks * ⓘ — ◉ Enable

Health check will determine if traffic can be served to the endpoint

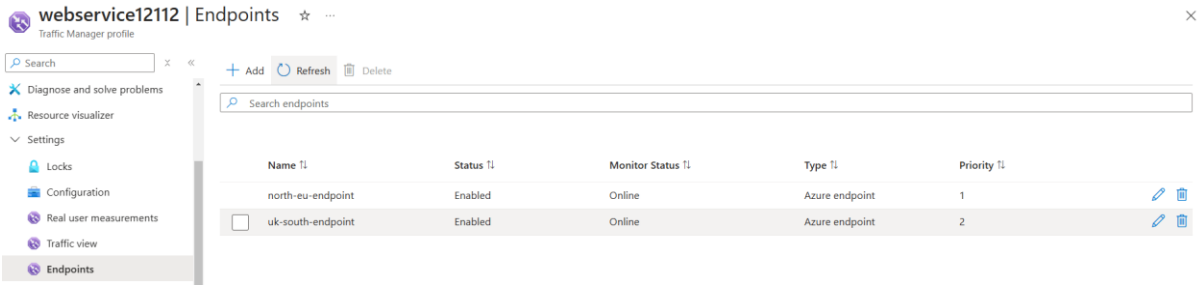
Add | Cancel



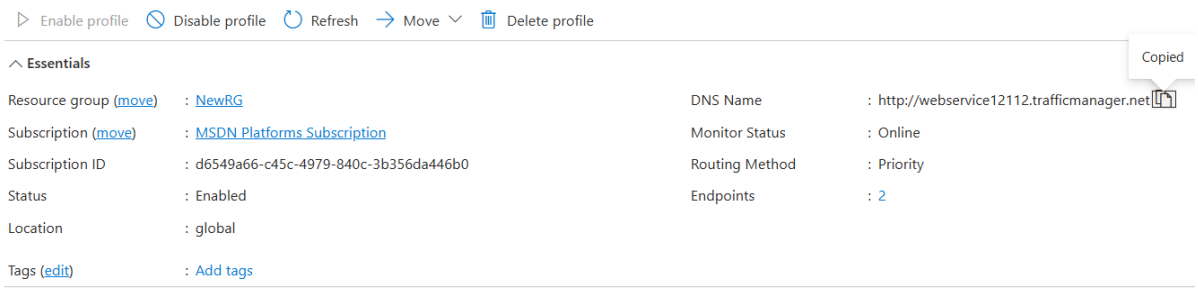
13. After adding them both move to the configuration tab and change the protocol to HTTPS because traffic manager listens on HTTPS save the changes.



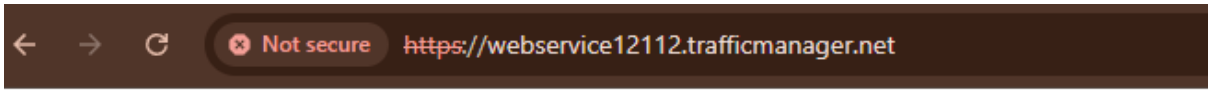
14. Below you can see that both of the endpoints are added successfully and they are showing online.



15. Now from the overview we need to copy the DNS name of traffic manager.

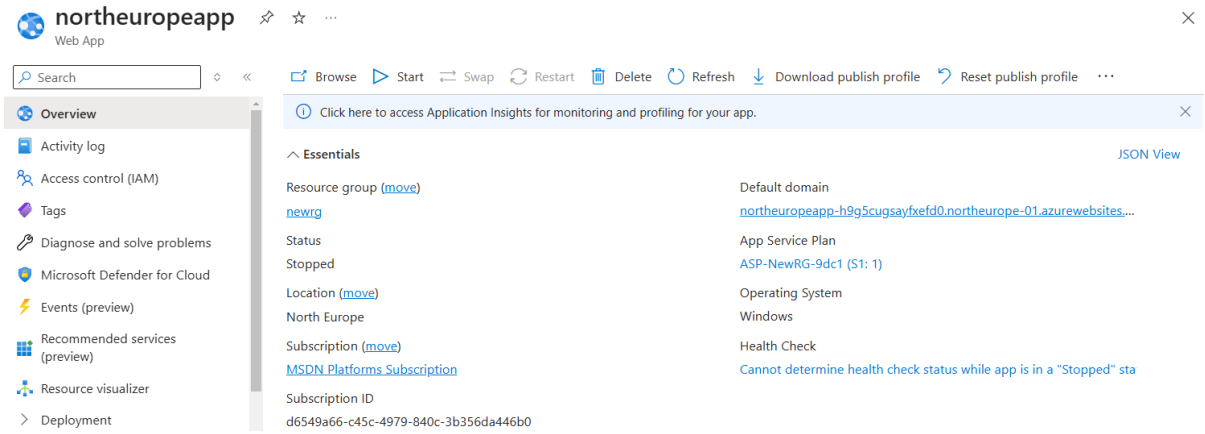


16. Here you can see that our traffic manager is up and running but it will only show the North Europe web app because it is set to first priority.

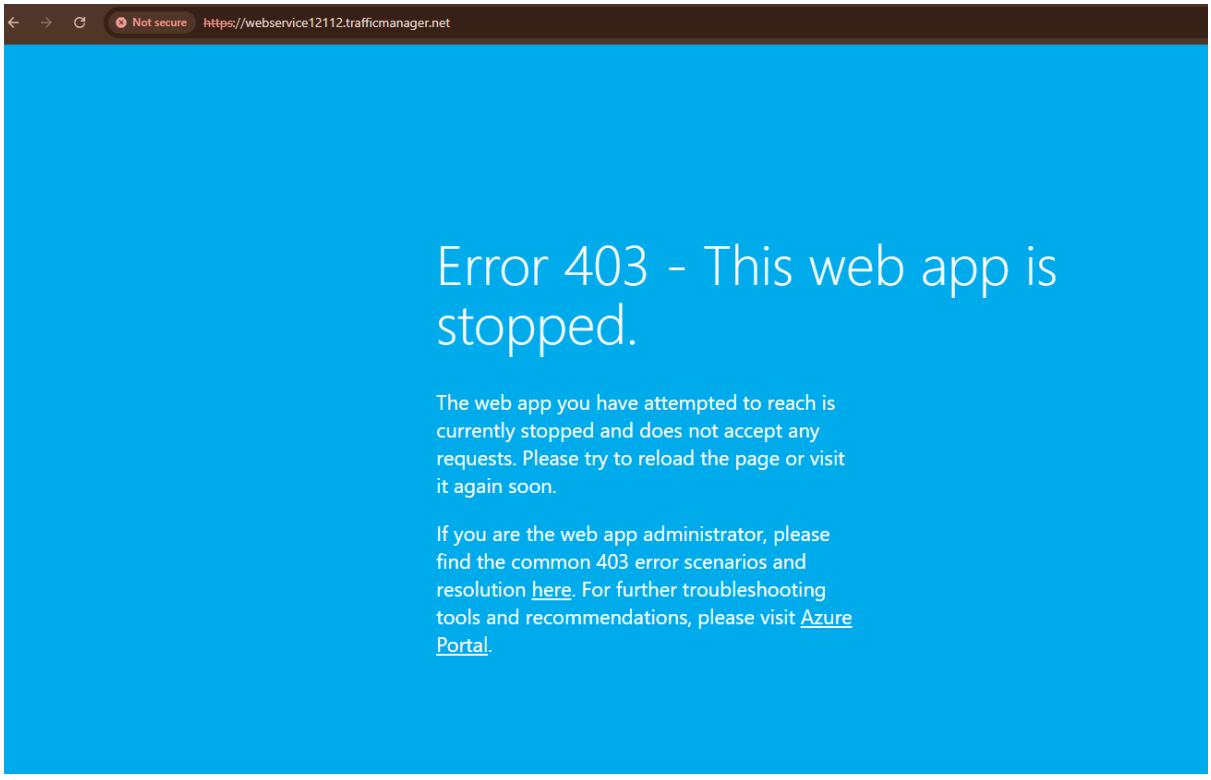


This is the Web App in North Europe Location

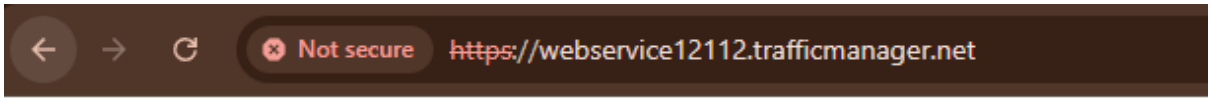
17. Now we will stop the web app in North Europe region and wait for couple of minutes.



18. As you can see, it takes some time to reflect the other web app



19. After a couple of minutes, you can see that we have another web app online in Traffic Manager.



This is the Web App in UK South Location

20. After completing the lab delete all the resources.