

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Azure Traffic Manager - Part One

By **Ramees** on May 16 2017

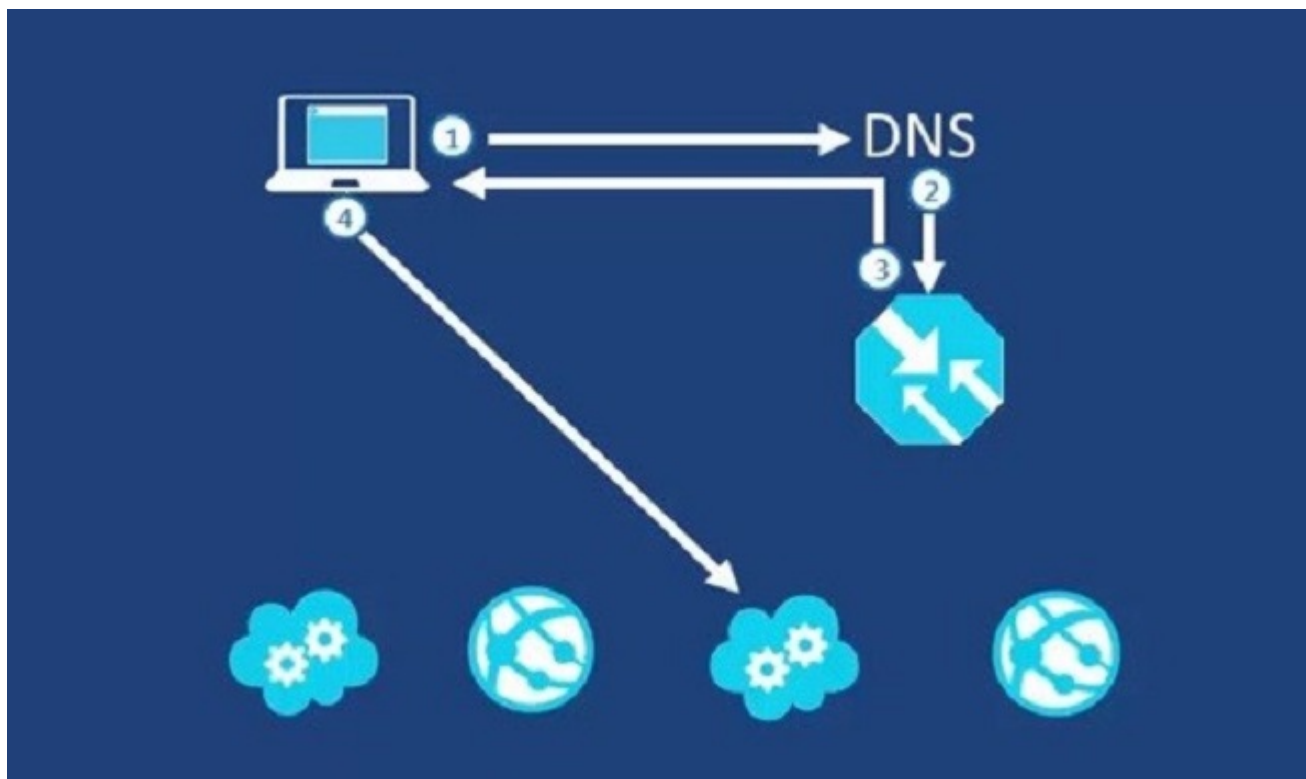
Azure Traffic Manager

Azure Traffic Manager controls the distribution of the user traffic to specified endpoints in different data centers which helps in improving availability, performance and provides load balance of traffic. It supports Azure VMs, Web Apps, and cloud services as endpoints. It uses the Domain Name System (DNS) to direct client requests to the most appropriate endpoint based on a traffic-routing method and the health of the endpoints. You can also use Traffic Manager with external, non-Azure endpoints. When the user tries to access any Web App it will first hit the traffic manager and traffic manager will decide which endpoint it should send to.

The following are the benefits of Azure Traffic Manager,

- Improved availability of critical applications by monitoring the endpoints and providing automatic failover when an endpoint goes down.
- Improve responsiveness for high-performance applications by directing traffic to the endpoint with the lowest network latency for the client.
- Can perform service maintenance without downtime as the Traffic Manager directs traffic to alternative endpoints while the maintenance is in progress.
- Can combine on-premises and Cloud-based applications as traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments

Let's discuss it more detail using the following diagram from Microsoft.



When a user requests any web app through the domain, they are querying the public DNS for the IP address and will connect to that IP address directly to access the service. But here, the public domain may have a CNAME entry pointing to a traffic manager instead of the Web App and Traffic manager have endpoints pointing towards our web apps. The most important point to understand is that Traffic Manager works at the DNS level. Traffic Manager uses DNS to direct clients to specific service endpoints based on the rules of the traffic-routing method. Clients connect to the selected endpoint directly. Traffic Manager is not a proxy or a gateway. Traffic Manager does not see the traffic passing between the client and the service.

Once the reply has been provided, the subsequent traffic will not go through the traffic manager until the next TTL we define. This makes it different from the load balancer. TTL is the time limit until next a process to check through the traffic manager again to route the endpoints. This will help us to not get locked in case of any failure of the nodes which we are connected and in case the traffic manager can switch the endpoints.

You have to take care in case of any databases for your web application, as each Webapp at different endpoints has to share one database and an update from one should reflect on the other one also.

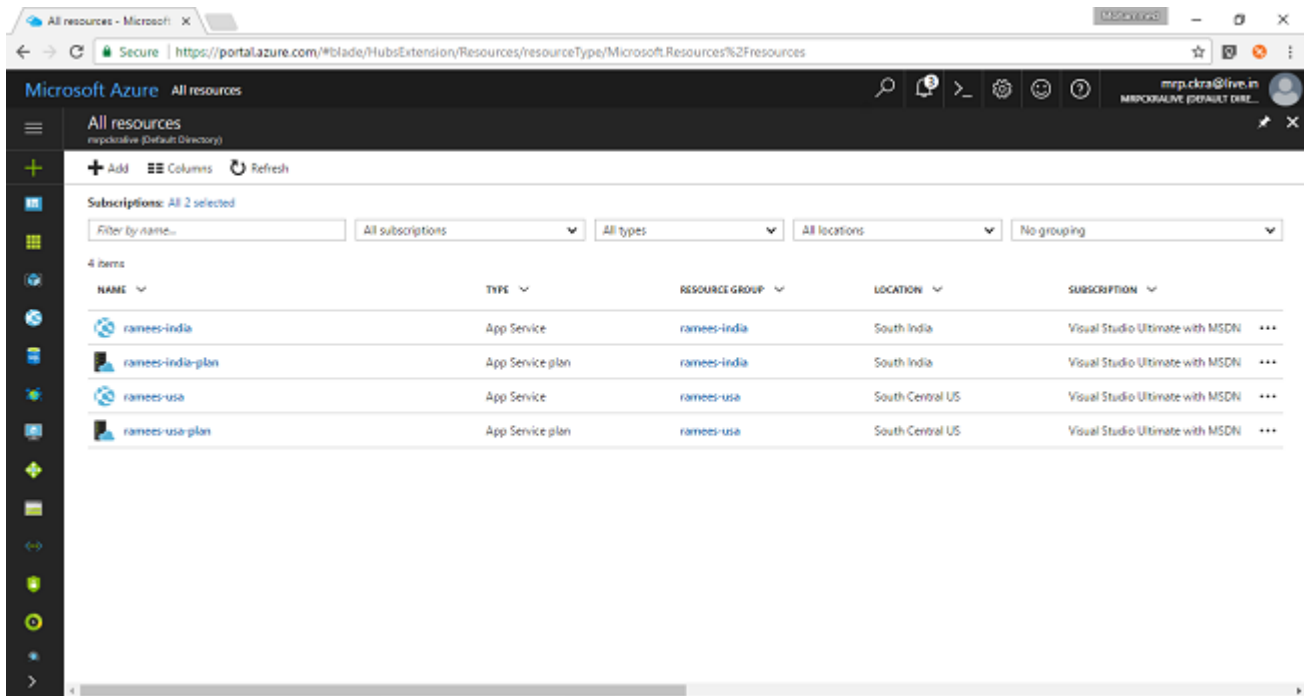
The following are the steps in general to work with Traffic Manager

First, we have to create the endpoints which may be cloud services, websites or it can be outside Azure service also.

What I am going to do here is create two web apps with same contents in two different locations. You check this [article](#) on how to do the same. The only thing you have to take care is when you select the plan. As shown below only plans from standard have traffic manager feature

Tier	Cores	GB RAM	Storage	Estimated Monthly Cost (INR)	Traffic Manager
P1 Premium	1	1.75	250 GB	14,752.68	Yes
P2 Premium	2	3.5	250 GB	29,505.37	Yes
P3 Premium	4	7	250 GB	59,010.73	Yes
S1 Standard	1	1.75	50 GB	2,950.54	Yes
S2 Standard	2	3.5	50 GB	5,901.07	Yes
S3 Standard	4	7	50 GB	11,802.15	Yes
B1 Basic	1	1.75	10 GB	2,163.73	Yes
B2 Basic	2	3.5	10 GB	-	Yes
B3 Basic	4	7	10 GB	-	Yes
F1 Free	-	-	1 GB	-	No
D1 Shared	-	-	1 GB	-	No

Now I have created two Web App with standard plans.



The two end points here are the URLs of the websites which are as follows,

- <http://ramees-india.azurewebsites.net>
- <http://ramees-usa.azurewebsites.net>

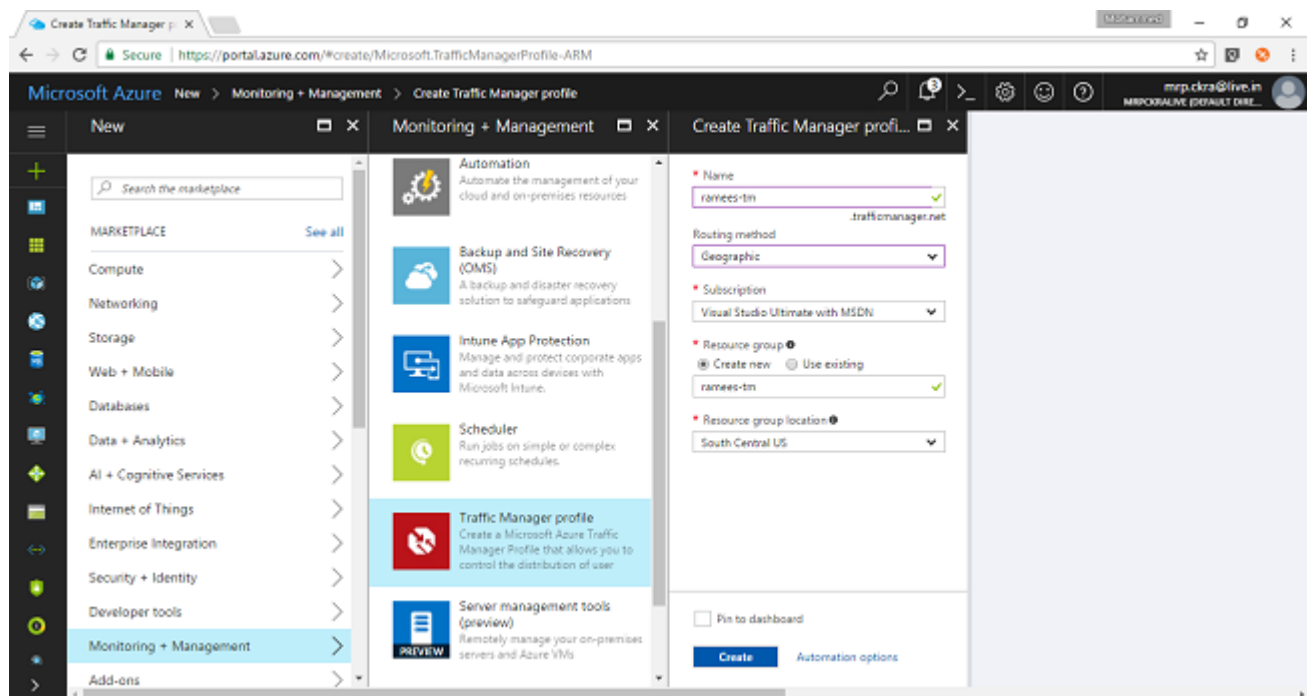
Now I am going to simply publish a webpage to both domains. You can check out this [article](#) on how to do it.

Here I am just giving a text, Hello World from India and Hello World from USA, to both the websites to identify which server we are accessing. In production, we will be using the same content and there the user needs to be aware of it. Here I am using just it for demo purposes.

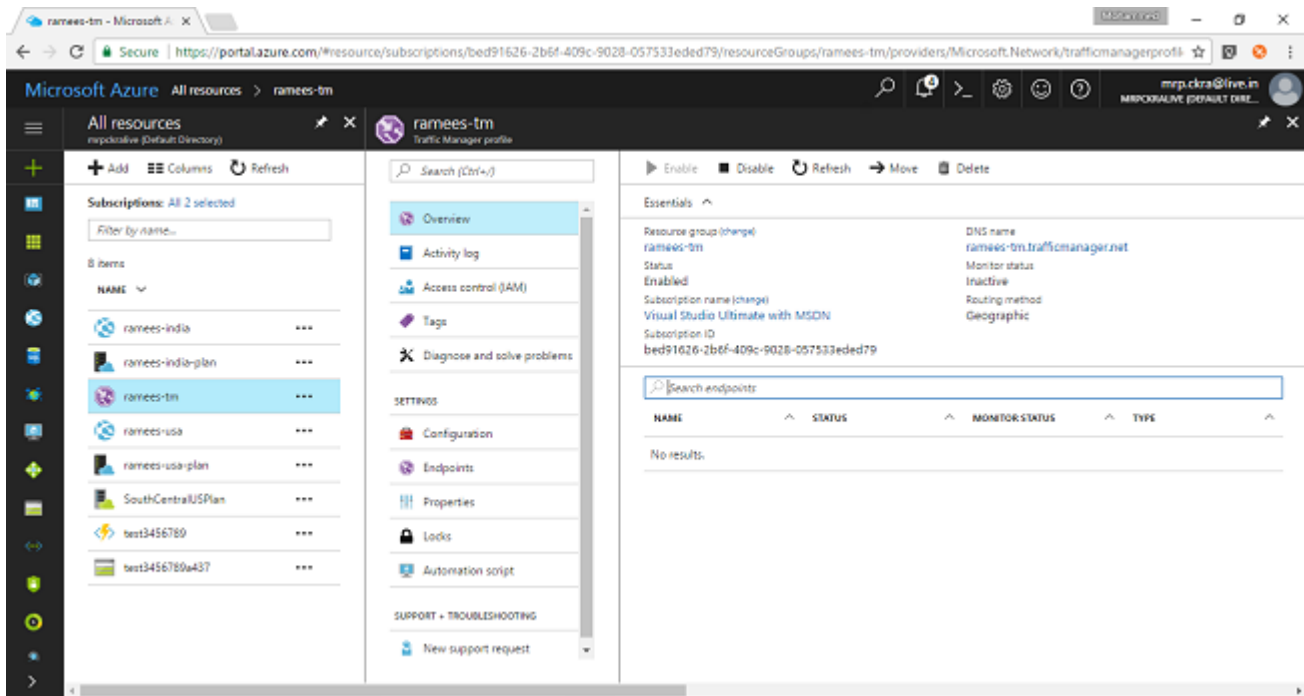




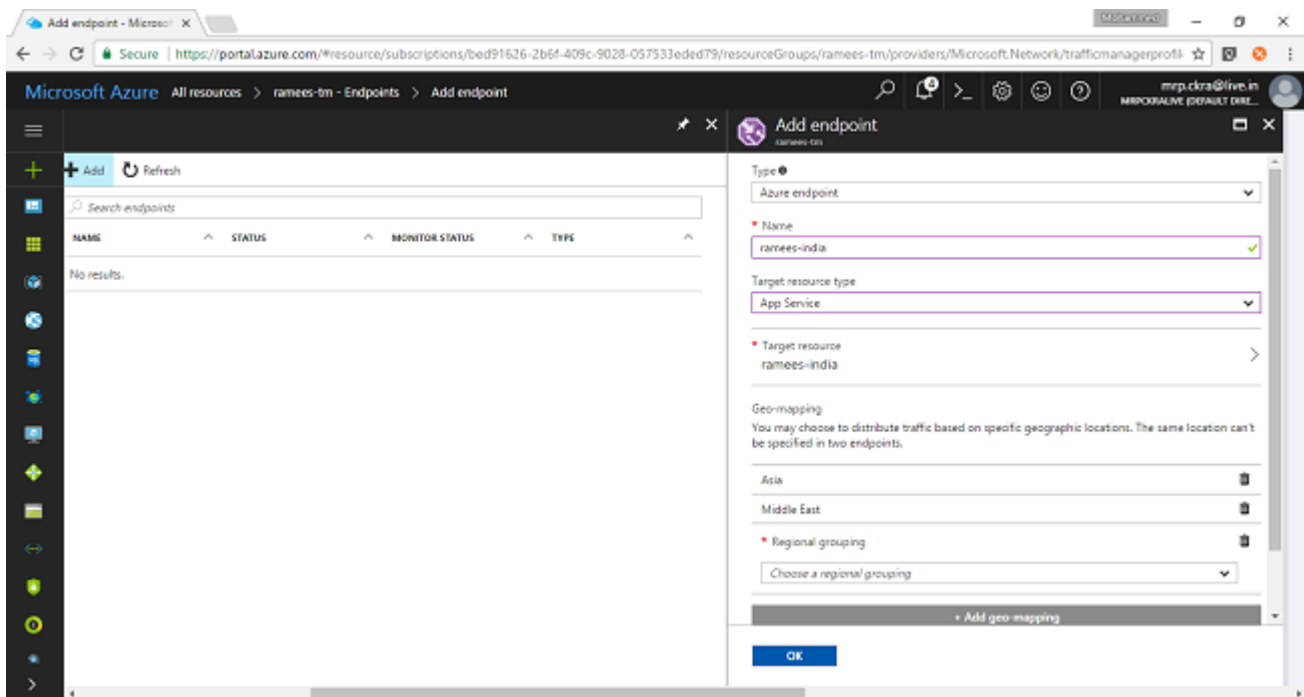
Next create a unique traffic manager profile with a unique DNS and a TTL. As traffic manager will assign a domain name with .trafficmanager.net it has to be unique



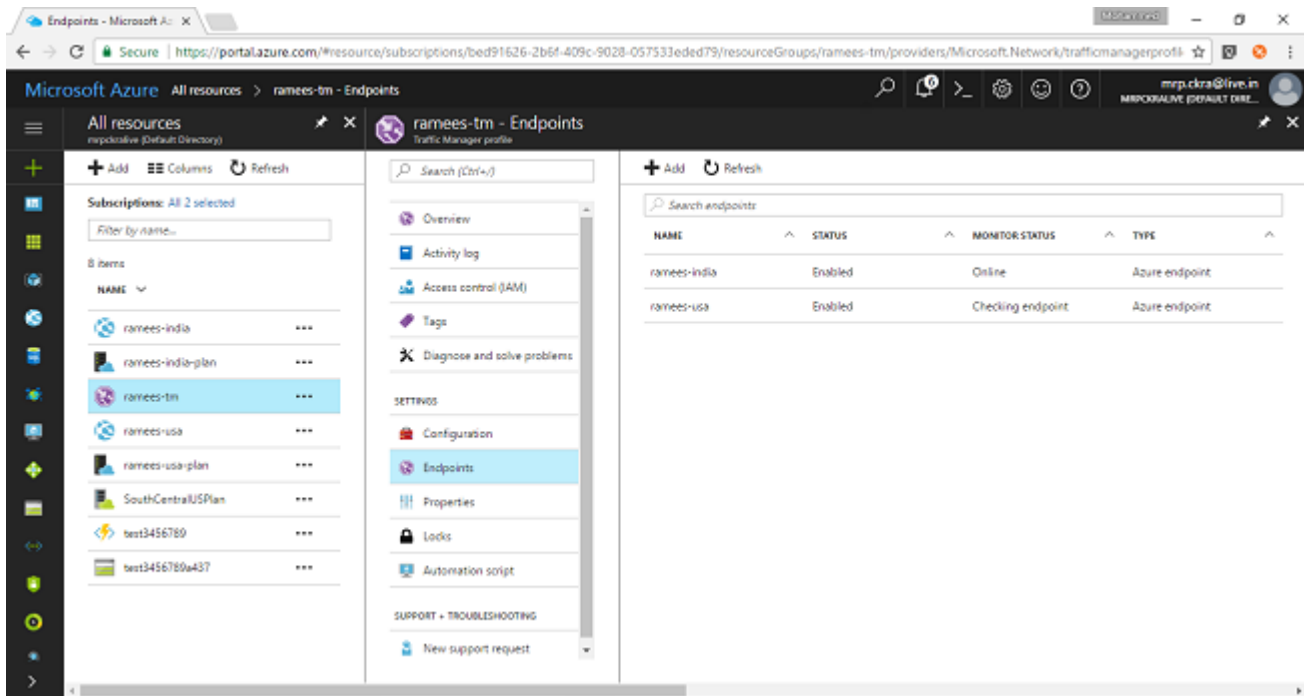
Here I am selecting the geographic routing method which will act according to the geographic regions. On creation by default the status will be inactive as we haven't pointed the endpoints here.



Select the endpoints section to add endpoints. While trying to add it will ask for a name to the endpoint, target resource type and will prompt for selecting from the available resources. As we have selected geographical for routing method here we can define for which regions we can assign the corresponding end point.



On adding it will check the endpoint for some time and on passing it will make it online.

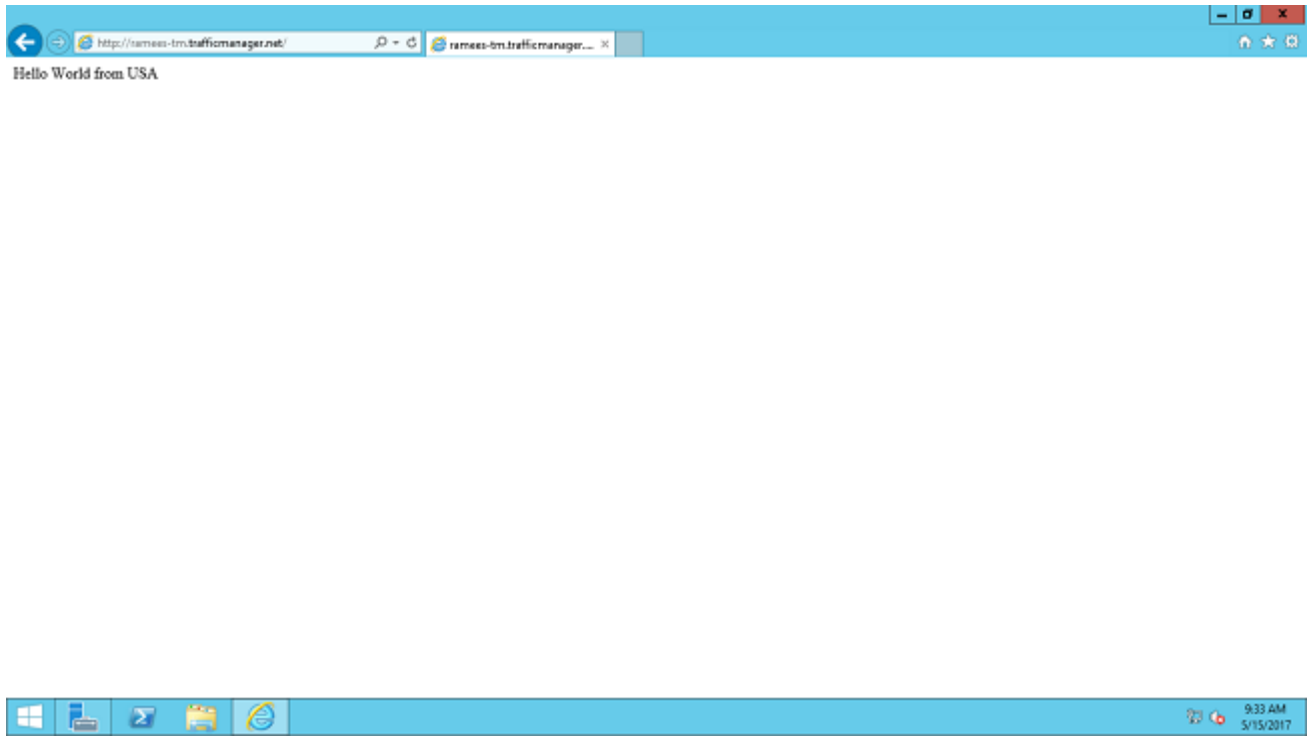


Now we have two endpoints added to our traffic manager and we can have as many as we require. Now let us test our demo. If you navigate directly to each endpoint's direct URL it will be showing the corresponding page from the servers only. To make use of the traffic manager you have to use the traffic manager URL instead so that the traffic manager will decide the endpoint.

On testing by the traffic manager URL, I will be getting the following result which is from the Indian server as I am from India



For demonstration, what I am doing here is creating a Virtual Machine in the South Central US and will try to access the same domain from it. While doing so we are getting the endpoint from the US which is the closest.



Thank you for using C# Corner