Contents

- Load Balancer
- Azure DNS

Resources

- Azure Roadmap
- Blog
- MSDN Forum
- Pricing
- Pricing calculator
- Service Limits
- Service updates
- SLA
- Videos

What is Traffic Manager?

2/4/2020 • 2 minutes to read • Edit Online

Azure Traffic Manager is a DNS-based traffic load balancer that enables you to distribute traffic optimally to services across global Azure regions, while providing high availability and responsiveness.

Traffic Manager uses DNS to direct client requests to the most appropriate service endpoint based on a traffic-routing method and the health of the endpoints. An endpoint is any Internet-facing service hosted inside or outside of Azure. Traffic Manager provides a range of traffic-routing methods and endpoint monitoring options to suit different application needs and automatic failover models. Traffic Manager is resilient to failure, including the failure of an entire Azure region.

> **NOTE**
>
> Azure provides a suite of fully managed load-balancing solutions for your scenarios. If you are looking for Transport Layer Security (TLS) protocol termination ("SSL offload") or per-HTTP/HTTPS request, application-layer processing, review Application Gateway. If you are looking for regional load balancing, review Load Balancer. Your end-to-end scenarios might benefit from combining these solutions as needed.
>
> For an Azure load-balancing options comparison, see Overview of load-balancing options in Azure.

Traffic Manager offers the following features:

Increase application availability

Traffic Manager delivers high availability for your critical applications by monitoring your endpoints and providing automatic failover when an endpoint goes down.

Improve application performance

Azure allows you to run cloud services or websites in datacenters located around the world. Traffic Manager improves application responsiveness by directing traffic to the endpoint with the lowest network latency for the client.

Perform service maintenance without downtime

You can perform planned maintenance operations on your applications without downtime. Traffic Manager can direct traffic to alternative endpoints while the maintenance is in progress.

Combine hybrid applications

Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including the "burst-to-cloud," "migrate-to-cloud," and "failover-to-cloud" scenarios.

Distribute traffic for complex deployments

Using nested Traffic Manager profiles, multiple traffic-routing methods can be combined to create sophisticated and flexible rules to scale to the needs of larger, more complex deployments.

Pricing

For pricing information, see Traffic Manager Pricing.

Next steps

- Learn how to create a Traffic Manager profile.
- Learn how Traffic Manager Works.
- View frequently asked questions about Traffic Manager.

Quickstart: Create a Traffic Manager profile using the Azure portal

2/1/2020 • 4 minutes to read • Edit Online

This quickstart describes how to create a Traffic Manager profile that delivers high availability for your web application.

In this quickstart, you'll read about two instances of a web application. Each of them is running in a different Azure region. You'll create a Traffic Manager profile based on endpoint priority. The profile directs user traffic to the primary site running the web application. Traffic Manager continuously monitors the web application. If the primary site is unavailable, it provides automatic failover to the backup site.

If you don't have an Azure subscription, create a free account now.

Sign in to Azure

Sign in to the Azure portal.

Prerequisites

For this quickstart, you'll need two instances of a web application deployed in two different Azure regions (*East US* and *West Europe*). Each will serve as primary and failover endpoints for Traffic Manager.

1. On the upper-left side of the screen, select **Create a resource** > **Web** > **Web App**.
2. In **Create a Web App**, type or select the following values in the **Basics** tab:
   - **Subscription** > **Resource Group**: Select **Create new** and then type **myResourceGroupTM1**.
   - **Instance Details** > **Name**: Type *myWebAppEastUS*.
   - **Instance Details** > **Publish**: Select **Code**.
   - **Instance Details** > **Runtime stack**: Select **ASP.NET V4.7**
   - **Instance Details** > **Operating System**: Select **Windows**.
   - **Instance Details** > **Region**: Select **East US**.
   - **App Service Plan** > **Windows Plan (East US)**: Select **Create new** and then type **myAppServicePlanEastUS**
   - **App Service Plan** > **Sku and size**: Select **Standard S1**.
3. Select the **Monitoring** tab, or select **Next:Monitoring**. Under **Monitoring**, set **Application Insights** > **Enable Application Insights** to **No**.
4. Select **Review and create**
5. Review the settings, and then click **Create**. When the Web App successfully deploys, it creates a default web site.
6. Follow the steps to create a second Web App named *myWebAppWestEurope*, with a **Resource Group** name of *myResourceGroupTM2*, a **Region** of *West Europe*, a **App Service Plan** name of **myAppServicePlanWestEurope**, and all the other settings the same as *myWebAppEastUS*.

Create a Traffic Manager profile

Create a Traffic Manager profile that directs user traffic based on endpoint priority.

1. On the upper-left side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile**.
2. In the **Create Traffic Manager profile**, enter, or select these settings:

SETTING	VALUE
Name	Enter a unique name for your Traffic Manager profile.
Routing method	Select **Priority**.
Subscription	Select the subscription you want the traffic manager profile applied to.
Resource group	Select *myResourceGroupTM1*.
Location	This setting refers to the location of the resource group. It has no effect on the Traffic Manager profile that will be deployed globally.

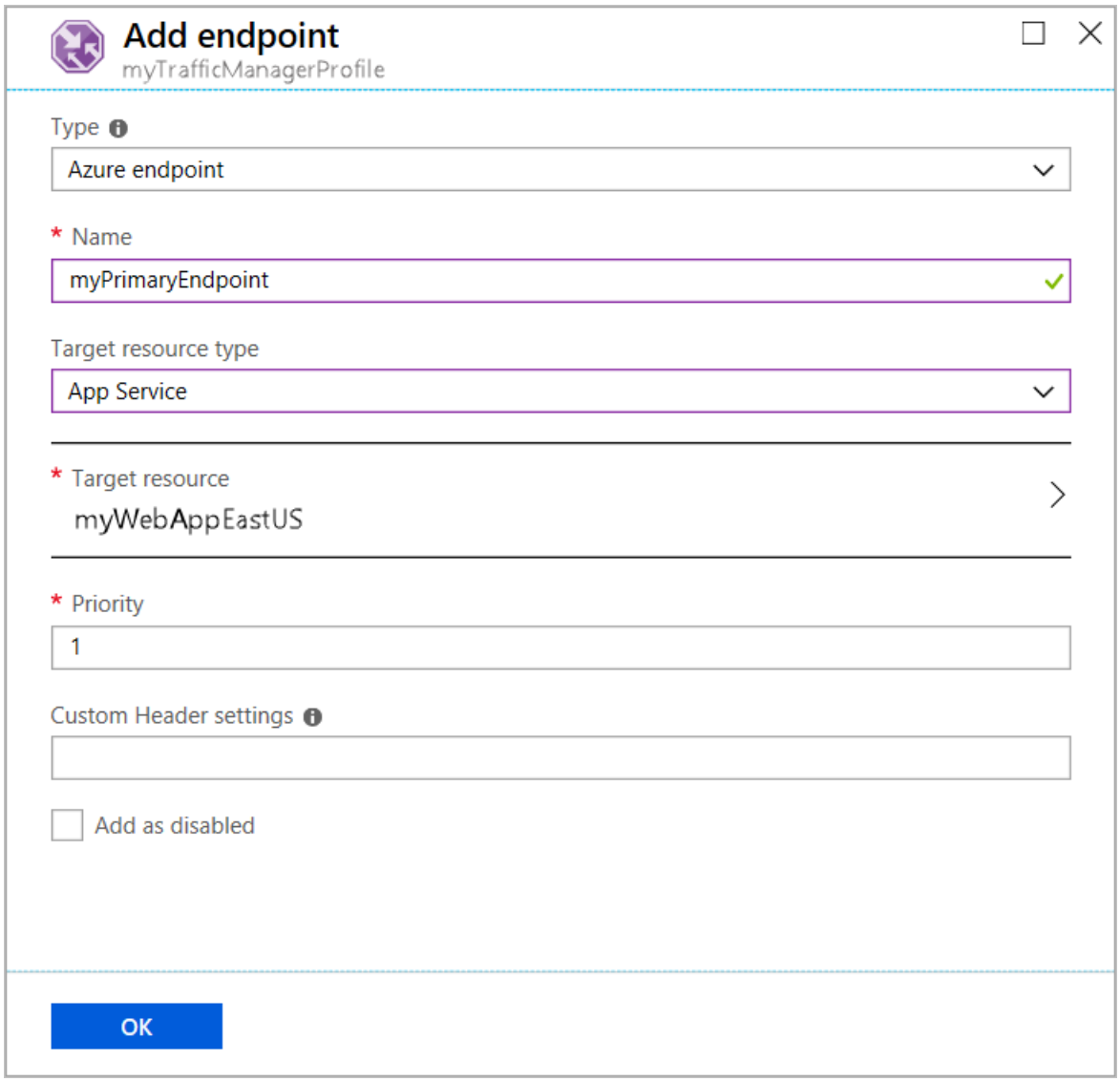
3. Select **Create**.

Add Traffic Manager endpoints

Add the website in the *East US* as primary endpoint to route all the user traffic. Add the website in *West Europe* as a failover endpoint. When the primary endpoint is unavailable, traffic automatically routes to the failover endpoint.

1. In the portal's search bar, enter the Traffic Manager profile name that you created in the preceding section.
2. Select the profile from the search results.
3. In **Traffic Manager profile**, in the **Settings** section, select **Endpoints**, and then select **Add**.
4. Enter, or select, these settings:

SETTING	VALUE
Type	Select **Azure endpoint**.
Name	Enter *myPrimaryEndpoint*.
Target resource type	Select **App Service**.
Target resource	Select **Choose an app service** > **East US**.
Priority	Select **1**. All traffic goes to this endpoint when it's healthy.



5. Select **OK**.

6. To create a failover endpoint for your second Azure region, repeat steps 3 and 4 with these settings:

SETTING	VALUE
Type	Select **Azure endpoint**.
Name	Enter *myFailoverEndpoint*.
Target resource type	Select **App Service**.
Target resource	Select **Choose an app service** > **West Europe**.
Priority	Select **2**. All traffic goes to this failover endpoint if the primary endpoint is unhealthy.

7. Select **OK**.

When you're done adding the two endpoints, they're displayed in **Traffic Manager profile**. Notice that their monitoring status is **Online** now.

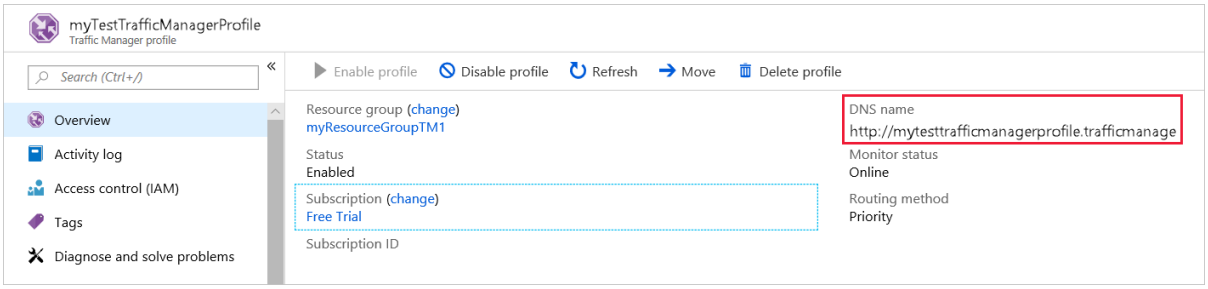
Test Traffic Manager profile

In this section, you'll check the domain name of your Traffic Manager profile. You'll also configure the primary

endpoint to be unavailable. Finally, you get to see that the web app is still available. It's because Traffic Manager sends the traffic to the failover endpoint.

Check the DNS name

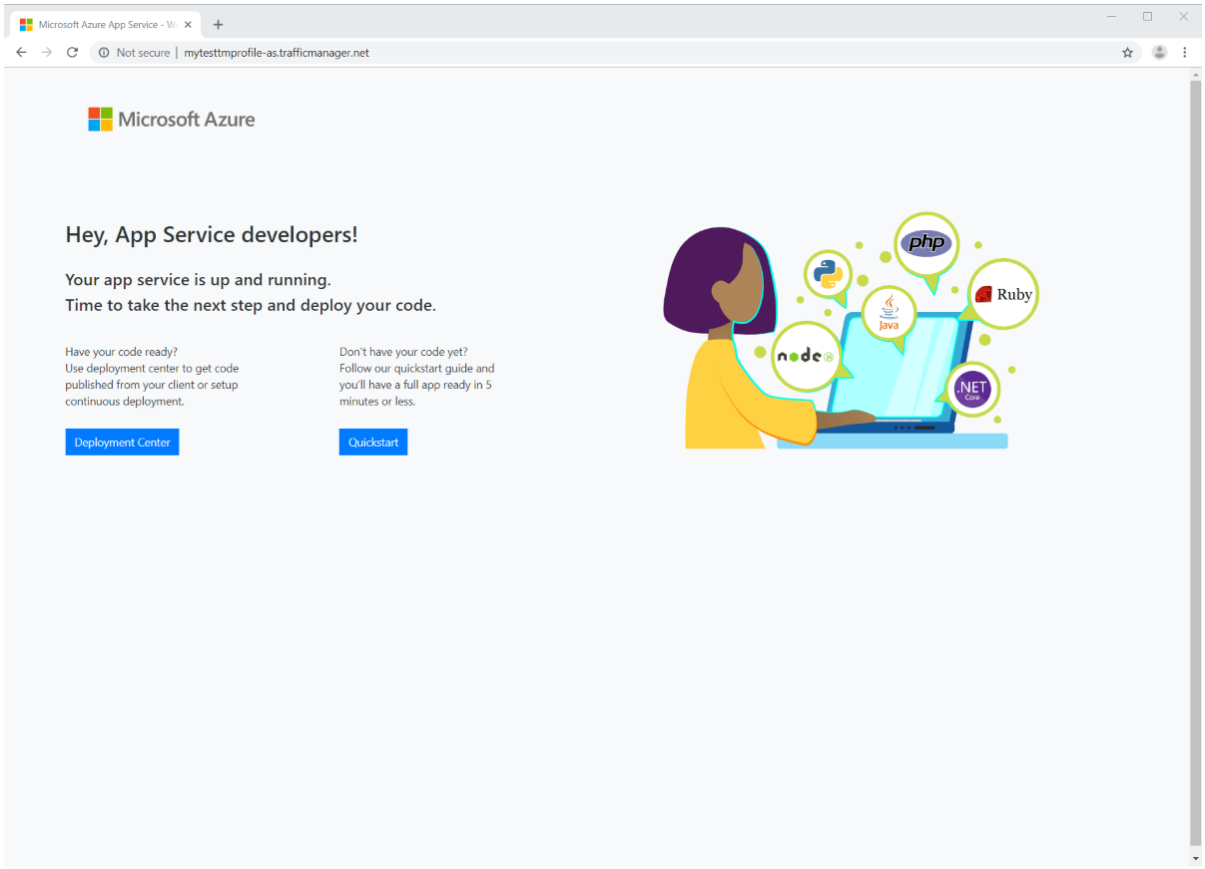
1. In the portal's search bar, search for the **Traffic Manager profile** name that you created in the preceding section.
2. Select the traffic manager profile. The **Overview** appears.
3. The **Traffic Manager profile** displays the DNS name of your newly created Traffic Manager profile.



View Traffic Manager in action

1. In a web browser, enter the DNS name of your Traffic Manager profile to view your Web App's default website.

> **NOTE**
>
> In this quickstart scenario, all requests route to the primary endpoint. It is set to **Priority 1**.



2. To view Traffic Manager failover in action, disable your primary site:
   a. In the Traffic Manager Profile page, from the **Overview** section, select **myPrimaryEndpoint**.
   b. In *myPrimaryEndpoint*, select **Disabled** > **Save**.

c. Close **myPrimaryEndpoint**. Notice that the status is **Disabled** now.

3. Copy the DNS name of your Traffic Manager Profile from the preceding step to view the website in a new web browser session.

4. Verify that the web app is still available.

The primary endpoint isn't available, so you were routed to the failover endpoint.

Clean up resources

When you're done, delete the resource groups, web applications, and all related resources. To do so, select each individual item from your dashboard and select **Delete** at the top of each page.

Next steps

In this quickstart, you created a Traffic Manager profile. It allows you to direct user traffic for high-availability web applications. To learn more about routing traffic, continue to the Traffic Manager tutorials.

Traffic Manager tutorials

Quickstart: Create a Traffic Manager profile for a highly available web application using Azure PowerShell

2/1/2020 • 4 minutes to read • Edit Online

This quickstart describes how to create a Traffic Manager profile that delivers high availability for your web application.

In this quickstart, you'll create two instances of a web application. Each of them is running in a different Azure region. You'll create a Traffic Manager profile based on endpoint priority. The profile directs user traffic to the primary site running the web application. Traffic Manager continuously monitors the web application. If the primary site is unavailable, it provides automatic failover to the backup site.

If you don't have an Azure subscription, create a free account now.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select **Try It** in the upper-right corner of a code block. Selecting **Try It** doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use PowerShell locally, this article requires the Azure PowerShell module version 5.4.1 or later. Run `Get-Module -ListAvailable Az` to find the installed version. If you need to upgrade, see Install Azure PowerShell module. If you are running PowerShell locally, you also need to run `Connect-AzAccount` to create a connection with Azure.

Create a Resource Group

Create a resource group using New-AzResourceGroup.

```
# Variables
$Location1="WestUS"

# Create a Resource Group
New-AzResourceGroup -Name MyResourceGroup -Location $Location1
```

Create a Traffic Manager profile

Create a Traffic Manager profile using New-AzTrafficManagerProfile that directs user traffic based on endpoint priority.

```
# Generates a random value
$Random=(New-Guid).ToString().Substring(0,8)
$mytrafficmanagerprofile="mytrafficmanagerprofile$Random"

New-AzTrafficManagerProfile `
-Name $mytrafficmanagerprofile `
-ResourceGroupName MyResourceGroup `
-TrafficRoutingMethod Priority `
-MonitorPath '/' `
-MonitorProtocol "HTTP" `
-RelativeDnsName $mytrafficmanagerprofile `
-Ttl 30 `
-MonitorPort 80
```

Create Web Apps

For this quickstart, you'll need two instances of a web application deployed in two different Azure regions (*West US* and *East US*). Each will serve as primary and failover endpoints for Traffic Manager.

Create Web App Service plans

Create Web App service plans using New-AzAppServicePlan for the two instances of the web application that you will deploy in two different Azure regions.

```
# Variables
$App1Name="AppServiceTM1$Random"
$App2Name="AppServiceTM2$Random"
$Location1="WestUS"
$Location2="EastUS"

# Create an App service plan
New-AzAppservicePlan -Name "$App1Name-Plan" -ResourceGroupName MyResourceGroup -Location $Location1 -Tier
Standard
New-AzAppservicePlan -Name "$App2Name-Plan" -ResourceGroupName MyResourceGroup -Location $Location2 -Tier
Standard
```

Create a Web App in the App Service Plan

Create two instances the web application using New-AzWebApp in the App Service plans in the *West US* and *East US* Azure regions.

```
$App1ResourceId=(New-AzWebApp -Name $App1Name -ResourceGroupName MyResourceGroup -Location $Location1 -
AppServicePlan "$App1Name-Plan").Id
$App2ResourceId=(New-AzWebApp -Name $App2Name -ResourceGroupName MyResourceGroup -Location $Location2 -
AppServicePlan "$App2Name-Plan").Id
```

Add Traffic Manager endpoints

Add the two Web Apps as Traffic Manager endpoints using New-AzTrafficManagerEndpoint to the Traffic Manager profile as follows:

- Add the Web App located in the *West US* Azure region as the primary endpoint to route all the user traffic.
- Add the Web App located in the *East US* Azure region as the failover endpoint. When the primary endpoint is unavailable, traffic automatically routes to the failover endpoint.

```
New-AzTrafficManagerEndpoint -Name "$App1Name-$Location1" `
-ResourceGroupName MyResourceGroup `
-ProfileName "$mytrafficmanagerprofile" `
-Type AzureEndpoints `
-TargetResourceId $App1ResourceId `
-EndpointStatus "Enabled"

New-AzTrafficManagerEndpoint -Name "$App2Name-$Location2" `
-ResourceGroupName MyResourceGroup `
-ProfileName "$mytrafficmanagerprofile" `
-Type AzureEndpoints `
-TargetResourceId $App2ResourceId `
-EndpointStatus "Enabled"
```

Test Traffic Manager profile

In this section, you'll check the domain name of your Traffic Manager profile. You'll also configure the primary endpoint to be unavailable. Finally, you get to see that the web app is still available. It's because Traffic Manager sends the traffic to the failover endpoint.

Determine the DNS name

Determine the DNS name of the Traffic Manager profile using Get-AzTrafficManagerProfile.

```
Get-AzTrafficManagerProfile -Name $mytrafficmanagerprofile `
-ResourceGroupName MyResourceGroup
```

Copy the **RelativeDnsName** value. The DNS name of your Traffic Manager profile is *http://<relativednsname>.trafficmanager.net*.

View Traffic Manager in action

1. In a web browser, enter the DNS name of your Traffic Manager profile (*http://<relativednsname>.trafficmanager.net*) to view your Web App's default website.

   > **NOTE**
   > In this quickstart scenario, all requests route to the primary endpoint. It is set to **Priority 1**.

2. To view Traffic Manager failover in action, disable your primary site using Disable-AzTrafficManagerEndpoint.

```
Disable-AzTrafficManagerEndpoint -Name $App1Name-$Location1 `
-Type AzureEndpoints `
-ProfileName $mytrafficmanagerprofile `
-ResourceGroupName MyResourceGroup `
-Force
```

3. Copy the DNS name of your Traffic Manager profile (*http://<relativednsname>.trafficmanager.net*) to view the website in a new web browser session.

4. Verify that the web app is still available.

Clean up resources

When you're done, delete the resource groups, web applications, and all related resources using Remove-AzResourceGroup.

```
Remove-AzResourceGroup -Name MyResourceGroup
```

Next steps

In this quickstart, you created a Traffic Manager profile that provides high availability for your web application. To learn more about routing traffic, continue to the Traffic Manager tutorials.

Traffic Manager tutorials

Quickstart: Create a Traffic Manager profile for a highly available web application using Azure CLI

2/1/2020 • 6 minutes to read • Edit Online

This quickstart describes how to create a Traffic Manager profile that delivers high availability for your web application.

In this quickstart, you'll create two instances of a web application. Each of them is running in a different Azure region. You'll create a Traffic Manager profile based on endpoint priority. The profile directs user traffic to the primary site running the web application. Traffic Manager continuously monitors the web application. If the primary site is unavailable, it provides automatic failover to the backup site.

If you don't have an Azure subscription, create a free account now.

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select **Try It** in the upper-right corner of a code block. Selecting **Try It** doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use the CLI locally, this tutorial requires that you are running a version of the Azure CLI version 2.0.28 or later. To find the version, run `az --version`. If you need to install or upgrade, see Install Azure CLI.

Create a resource group

Create a resource group with az group create. An Azure resource group is a logical container into which Azure resources are deployed and managed.

The following example creates a resource group named *myResourceGroup* in the *eastus* location:

```
az group create \
  --name myResourceGroup \
  --location eastus
```

Create a Traffic Manager profile

Create a Traffic Manager profile using az network traffic-manager profile create that directs user traffic based on endpoint priority.

In the following example, replace **<profile_name>** with a unique Traffic Manager profile name.

```
az network traffic-manager profile create \
 --name <profile_name> \
 --resource-group myResourceGroup \
 --routing-method Priority \
 --path "/" \
 --protocol HTTP \
 --unique-dns-name <profile_name> \
 --ttl 30 \
 --port 80
```

Create web apps

For this quickstart, you'll need two instances of a web application deployed in two different Azure regions (*East US* and *West Europe*). Each will serve as primary and failover endpoints for Traffic Manager.

Create web app service plans

Create web app service plans using az appservice plan create for the two instances of the web application that you will deploy in two different Azure regions.

In the following example, replace **<appspname_eastus>** and **<appspname_westeurope>** with a unique App Service Plan Name

```
az appservice plan create \
    --name <appspname_eastus> \
    --resource-group myResourceGroup \
    --location eastus \
    --sku S1

az appservice plan create \
    --name <appspname_westeurope> \
    --resource-group myResourceGroup \
    --location westeurope \
    --sku S1
```

Create a web app in the app service plan

Create two instances the web application using az webapp create in the App Service plans in the *East US* and *West Europe* Azure regions.

In the following example, replace **<app1name_eastus>** and **<app2name_westeurope>** with a unique App

Name, and replace **<appspname_eastus>** and **<appspname_westeurope>** with the name used to create the App Service plans in the previous section.

```
az webapp create \
    --name <app1name_eastus> \
    --plan <appspname_eastus> \
    --resource-group myResourceGroup

az webapp create \
    --name <app2name_westeurope> \
    --plan <appspname_westeurope> \
    --resource-group myResourceGroup
```

Add Traffic Manager endpoints

Add the two Web Apps as Traffic Manager endpoints using az network traffic-manager endpoint create to the Traffic Manager profile as follows:

- Determine the Web App id and add the Web App located in the *East US* Azure region as the primary endpoint to route all the user traffic.
- Determinet the Web App id and add the Web App located in the *West Europe* Azure region as the failover endpoint.

When the primary endpoint is unavailable, traffic automatically routes to the failover endpoint.

In the following example, replace **<app1name_eastus>** and **<app2name_westeurope>** with the App Names created for each region in the previous section, replace **<appspname_eastus>** and **<appspname_westeurope>** with the name used to create the App Service plans in the previous section, and replace **<profile_name>** with the profile name used in the previous section.

East US endpoint

```
az webapp show \
    --name <app1name_eastus> \
    --resource-group myResourceGroup \
    --query id
```

Make note of id displayed in output and use in the following command to add the endpoint:

```
az network traffic-manager endpoint create \
    --name <app1name_eastus> \
    --resource-group myResourceGroup \
    --profile-name <profile_name> \
    --type azureEndpoints \
    --target-resource-id <ID from az webapp show> \
    --priority 1 \
    --endpoint-status Enabled
```

West Europe endpoint

```
az webapp show \
    --name <app2name_westeurope> \
    --resource-group myResourceGroup \
    --query id
```

Make note of id displayed in output and use in the following command to add the endpoint:

```
az network traffic-manager endpoint create \
    --name <app1name_westeurope> \
    --resource-group myResourceGroup \
    --profile-name <profile_name> \
    --type azureEndpoints \
    --target-resource-id <ID from az webapp show> \
    --priority 2 \
    --endpoint-status Enabled
```

Test your Traffic Manager profile

In this section, you'll check the domain name of your Traffic Manager profile. You'll also configure the primary endpoint to be unavailable. Finally, you get to see that the web app is still available. It's because Traffic Manager sends the traffic to the failover endpoint.

In the following example, replace **<app1name_eastus>** and **<app2name_westeurope>** with the App Names created for each region in the previous section, replace **<appspname_eastus>** and **<appspname_westeurope>** with the name used to create the App Service plans in the previous section, and replace **<profile_name>** with the profile name used in the previous section.

Determine the DNS name

Determine the DNS name of the Traffic Manager profile using az network traffic-manager profile show.

```
az network traffic-manager profile show \
    --name <profile_name> \
    --resource-group myResourceGroup \
    --query dnsConfig.fqdn
```

Copy the **RelativeDnsName** value. The DNS name of your Traffic Manager profile is *http://<relativednsname>.trafficmanager.net*.

View Traffic Manager in action

1. In a web browser, enter the DNS name of your Traffic Manager profile (*http://<relativednsname>.trafficmanager.net*) to view your Web App's default website.

   > **NOTE**
   >
   > In this quickstart scenario, all requests route to the primary endpoint. It is set to **Priority 1**.

2. To view Traffic Manager failover in action, disable your primary site using az network traffic-manager endpoint update.

```
az network traffic-manager endpoint update \
    --name <app1name_eastus> \
    --resource-group myResourceGroup \
    --profile-name <profile_name> \
    --type azureEndpoints \
    --endpoint-status Disabled
```

3. Copy the DNS name of your Traffic Manager profile (*http://<relativednsname>.trafficmanager.net*) to view the website in a new web browser session.

4. Verify that the web app is still available.

Clean up resources

When you're done, delete the resource groups, web applications, and all related resources using az group delete.

```
az group delete \
    --resource-group myResourceGroup
```

Next steps

In this quickstart, you created a Traffic Manager profile that provides high availability for your web application. To learn more about routing traffic, continue to the Traffic Manager tutorials.

Traffic Manager tutorials

Tutorial: Improve website response using Traffic Manager

2/11/2020 • 9 minutes to read • Edit Online

This tutorial describes how to use Traffic Manager to create a highly responsive website by directing user traffic to the website with the lowest latency. Typically, the datacenter with the lowest latency is the one that is closest in geographic distance.

In this tutorial, you learn how to:

- Create two VMs running a basic website on IIS
- Create two test VMs to view Traffic Manager in action
- Configure DNS name for the VMs running IIS
- Create a Traffic Manager profile for improved website performance
- Add VM endpoints to the Traffic Manager profile
- View Traffic Manager in action

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

In order to see the Traffic Manager in action, this tutorial requires that you deploy the following:

- Two instances of basic websites running in different Azure regions - **East US** and **West Europe**.
- Two test VMs for testing the Traffic Manager - one VM in **East US** and the second VM in **West Europe**. The test VMs are used to illustrate how Traffic Manager routes user traffic to the website that is running in the same region as it provides the lowest latency.

Sign in to Azure

Sign in to the Azure portal at https://portal.azure.com.

Create websites

In this section, you create two website instances that provide the two service endpoints for the Traffic Manager profile in two Azure regions. Creating the two websites includes the following steps:

1. Create two VMs for running a basic website - one in **East US**, and the other in **West Europe**.
2. Install IIS server on each VM and update the default website page that describes the VM name that a user is connected to when visiting the website.

Create VMs for running websites

In this section, you create two VMs *myIISVMEastUS* and *myIISVMWestEurope* in the **East US** and **West Europe** Azure regions.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

   - **Subscription** > **Resource Group**: Select **Create new** and then type **myResourceGroupTM1**.
   - **Instance Details** > **Virtual machine name**: Type *myIISVMEastUS*.
   - **Instance Details** > **Region**: Select **East US**.

- **Administrator Account** > **Username**: Enter a user name of your choosing.
- **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
- **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
- **Inbound Port Rules** > **Select inbound ports**: Select **RDP** and **HTTP** in the pull down box.

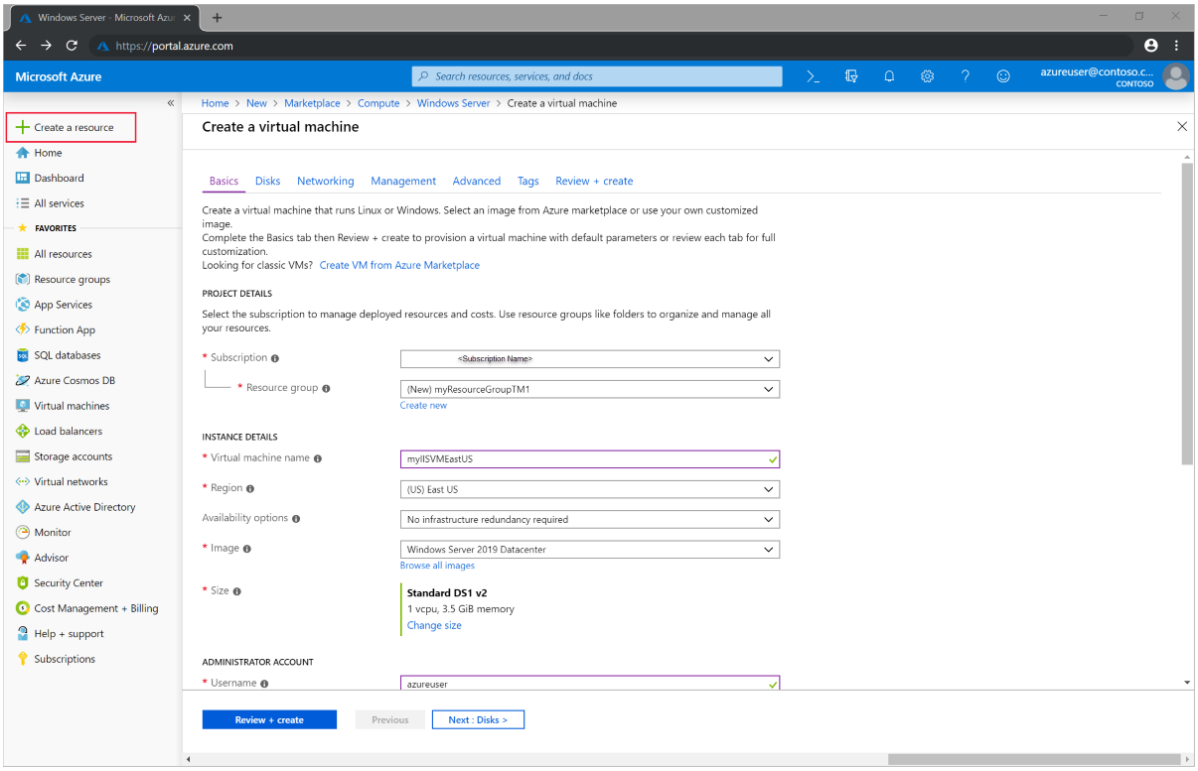
3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.

4. Select **Review + create**.

5. Review the settings, and then click **Create**.

6. Follow the steps to create a second VM named *myIISVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myIISVMEastUS*.

7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.



Install IIS and customize the default web page

In this section, you install the IIS server on the two VMs *myIISVMEastUS* and *myIISVMWestEurope*, and then update the default website page. The customized website page shows the name of the VM that you are connecting to when you visit the website from a web browser.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.

2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.

3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.

4. Select **OK**.

5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or

Continue, to proceed with the connection.

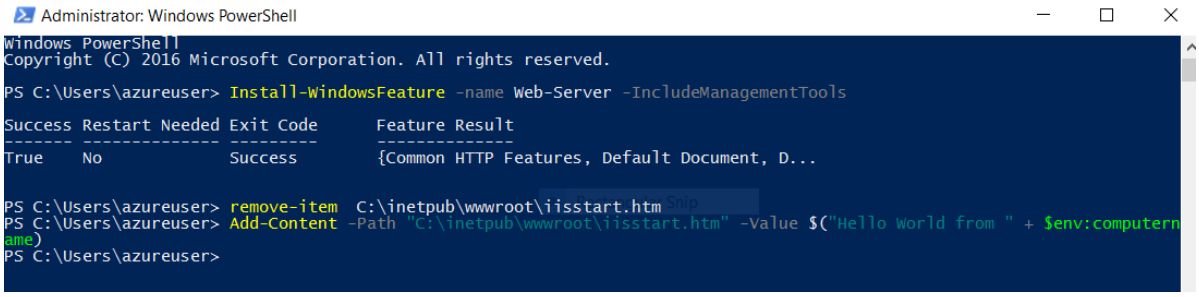
6. On the server desktop, navigate to **Windows Administrative Tools**>**Server Manager**.

7. Launch Windows PowerShell on VM1 and using the following commands to install IIS server and update the default htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)
```



8. Close the RDP connection with *myIISVMEastUS*.

9. Repeat steps 1-8 with by creating an RDP connection with the VM *myIISVMWestEurope* within the *myResourceGroupTM2* resource group to install IIS and customize its default web page.

Configure DNS names for the VMs running IIS

Traffic Manager routes user traffic based on DNS name of the service endpoints. In this section, you configure the DNS names for the IIS servers - *myIISVMEastUS* and *myIISVMWestEurope*.

1. Click **All resources** in the left-hand menu, and then from the resources list, select *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, under **DNS name**, select **Configure**.
3. On the **Configuration** page, under DNS name label, add a unique name, and then select **Save**.
4. Repeat steps 1-3, for the VM named *myIISVMWestEurope* that is located in the *myResourceGroupTM2* resource group.

Create test VMs

In this section, you create a VM (*myVMEastUS* and *myVMWestEurope*) in each Azure region (**East US** and **West Europe**). You will use these VMs to test how Traffic Manager routes traffic to the nearest IIS server when you browse to the website.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

   - **Subscription** > **Resource Group**: Select **myResourceGroupTM1**.
   - **Instance Details** > **Virtual machine name**: Type *myVMEastUS*.
   - **Instance Details** > **Region**: Select **East US**.
   - **Administrator Account** > **Username**: Enter a user name of your choosing.
   - **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.

- **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
- **Inbound Port Rules** > **Select inbound ports**: Select **RDP** in the pull down box.

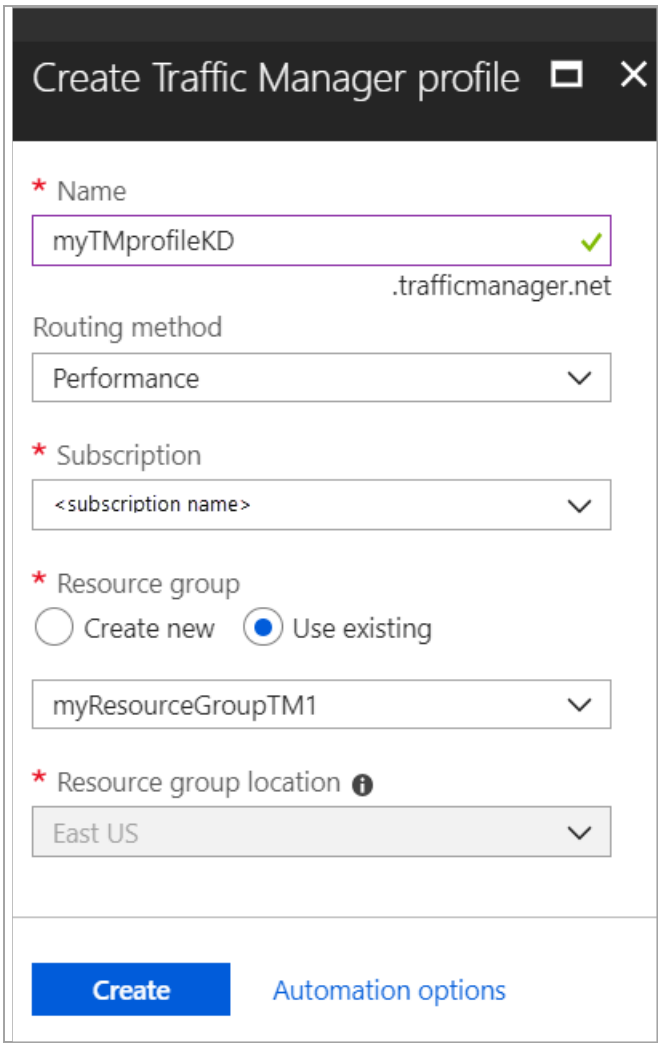
3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.
4. Select **Review + create**.
5. Review the settings, and then click **Create**.
6. Follow the steps to create a second VM named *myVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myVMEastUS*.
7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

Create a Traffic Manager profile

Create a Traffic Manager profile that directs user traffic by sending them to the endpoint with lowest latency.

1. On the top left-hand side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
2. In the **Create Traffic Manager profile**, enter or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	This name needs to be unique within the trafficmanager.net zone and results in the DNS name, trafficmanager.net that is used to access your Traffic Manager profile.
Routing method	Select the **Performance** routing method.
Subscription	Select your subscription.
Resource group	Select the Resource group *myResourceGroupTM1*.
Location	Select **East US**. This setting refers to the location of the resource group, and has no impact on the Traffic Manager profile that will be deployed globally.



Add Traffic Manager endpoints

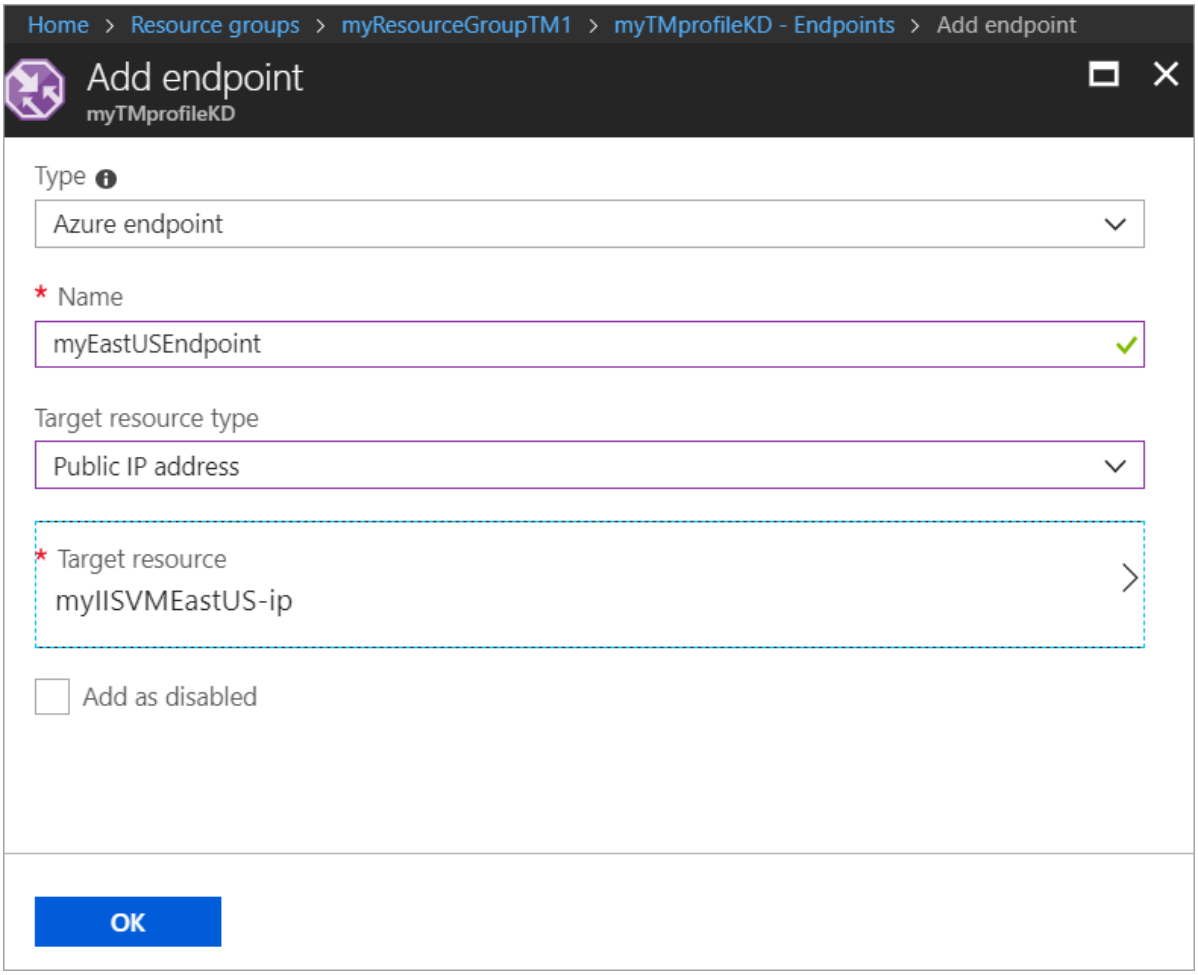
Add the two VMs running the IIS servers - *myIISVMEastUS* & *myIISVMWestEurope* to route user traffic to the closest endpoint to the user.

1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section and select the profile in the results that the displayed.
2. In **Traffic Manager profile**, in the **Settings** section, click **Endpoints**, and then click **Add**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

SETTING	VALUE
Type	Azure endpoint
Name	myEastUSEndpoint
Target resource type	Public IP Address
Target resource	**Choose a Public IP address** to show the listing of resources with Public IP addresses under the same subscription. In **Resource**, select the public IP address named *myIISVMEastUS-ip*. This is the public IP address of the IIS server VM in East US.

4. Repeat steps 2 and 3 to add another endpoint named *myWestEuropeEndpoint* for the public IP address *myIISVMWestEurope-ip* that is associated with the IIS server VM named *myIISVMWestEurope*.

5. When the addition of both endpoints is complete, they are displayed in **Traffic Manager profile** along with their monitoring status as **Online**.



Test Traffic Manager profile

In this section, you test how the Traffic Manager routes user traffic to the nearest VMs running the website to provide minimum latency. To view the Traffic Manager in action, complete the following steps:

1. Determine the DNS name of your Traffic Manager profile.
2. View Traffic Manager in action as follows:
   - From the test VM (*myVMEastUS*) that is located in the **East US** region, in a web browser, browse to the DNS name of your Traffic Manager profile.
   - From the test VM (*myVMWestEurope*) that is located in the **West Europe** region, in a web browser, browse to the DNS name of your Traffic Manager profile.

Determine DNS name of Traffic Manager profile

In this tutorial, for simplicity, you use the DNS name of the Traffic Manager profile to visit the websites.

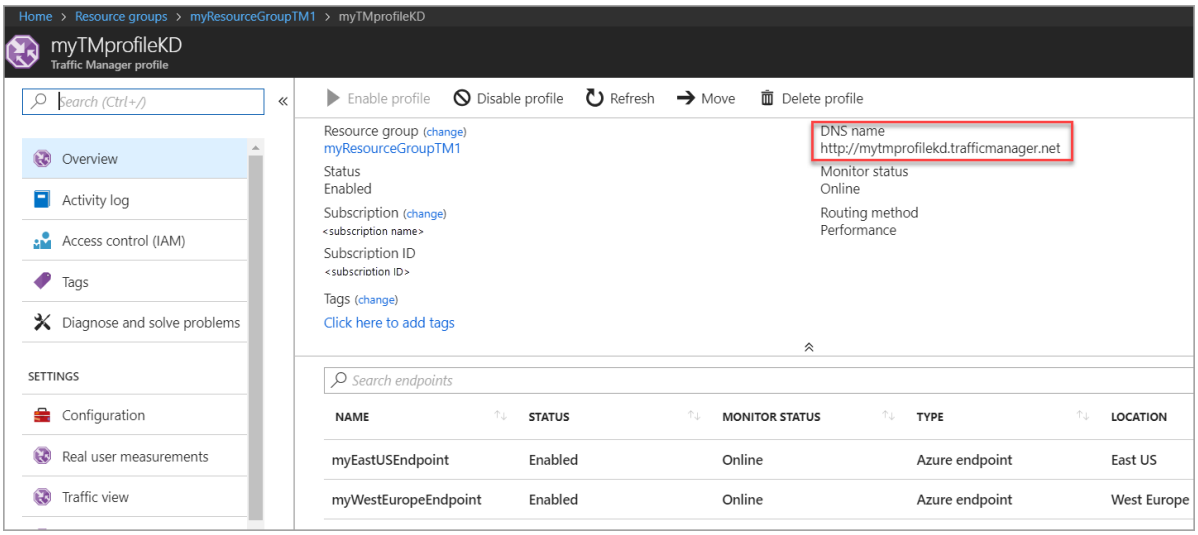
You can determine the DNS name of the Traffic Manager profile as follows:

1. In the portal's search bar, search for the **Traffic Manager profile** name that you created in the preceding section. In the results that are displayed, click the traffic manager profile.

2. Click **Overview**.

3. The **Traffic Manager profile** displays the DNS name of your newly created Traffic Manager profile. In production deployments, you configure a vanity domain name to point to the Traffic Manager domain

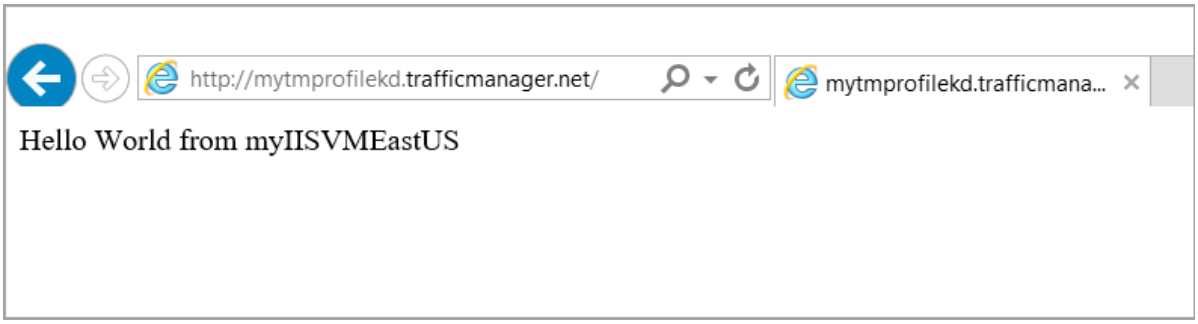
name, using a DNS CNAME record.



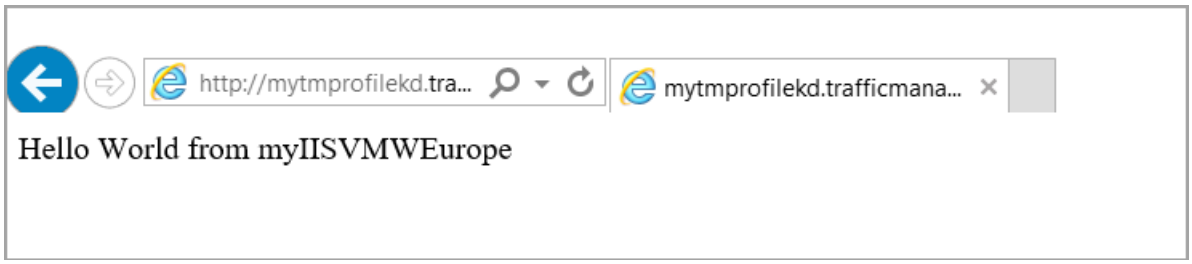
View Traffic Manager in action

In this section, you can see the Traffic Manager is action.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.
3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.
5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.
6. In a web browser on the VM *myVMEastUS*, type the DNS name of your Traffic Manager profile to view your website. Since the VM located in **East US**, you are routed to the nearest website hosted on the nearest IIS server *myIISVMEastUS* that is located in **East US**.



7. Next, connect to the VM *myVMWestEurope* located in **West Europe** using steps 1-5 and browse to the Traffic Manager profile domain name from this VM. Since the VM located in **West Europe**, you are now routed to the website hosted on nearest the IIS server *myIISVMWestEurope* that is located in **West Europe**.



Delete the Traffic Manager profile

When no longer needed, delete the resource groups (**ResourceGroupTM1** and **ResourceGroupTM2**). To do so, select the resource group (**ResourceGroupTM1** or **ResourceGroupTM2**), and then select **Delete**.

Next steps

Distribute traffic to a set of endpoints

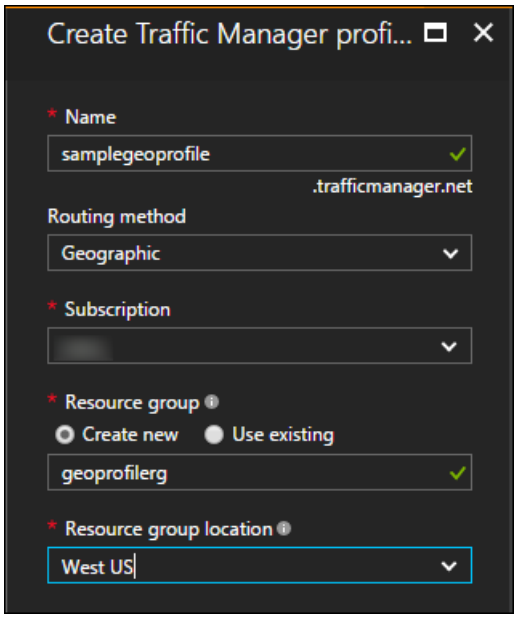
Tutorial: Configure the geographic traffic routing method using Traffic Manager

2/1/2020 • 3 minutes to read • Edit Online

The Geographic traffic routing method allows you to direct traffic to specific endpoints based on the geographic location where the requests originate. This tutorial shows you how to create a Traffic Manager profile with this routing method and configure the endpoints to receive traffic from specific geographies.

Create a Traffic Manager Profile

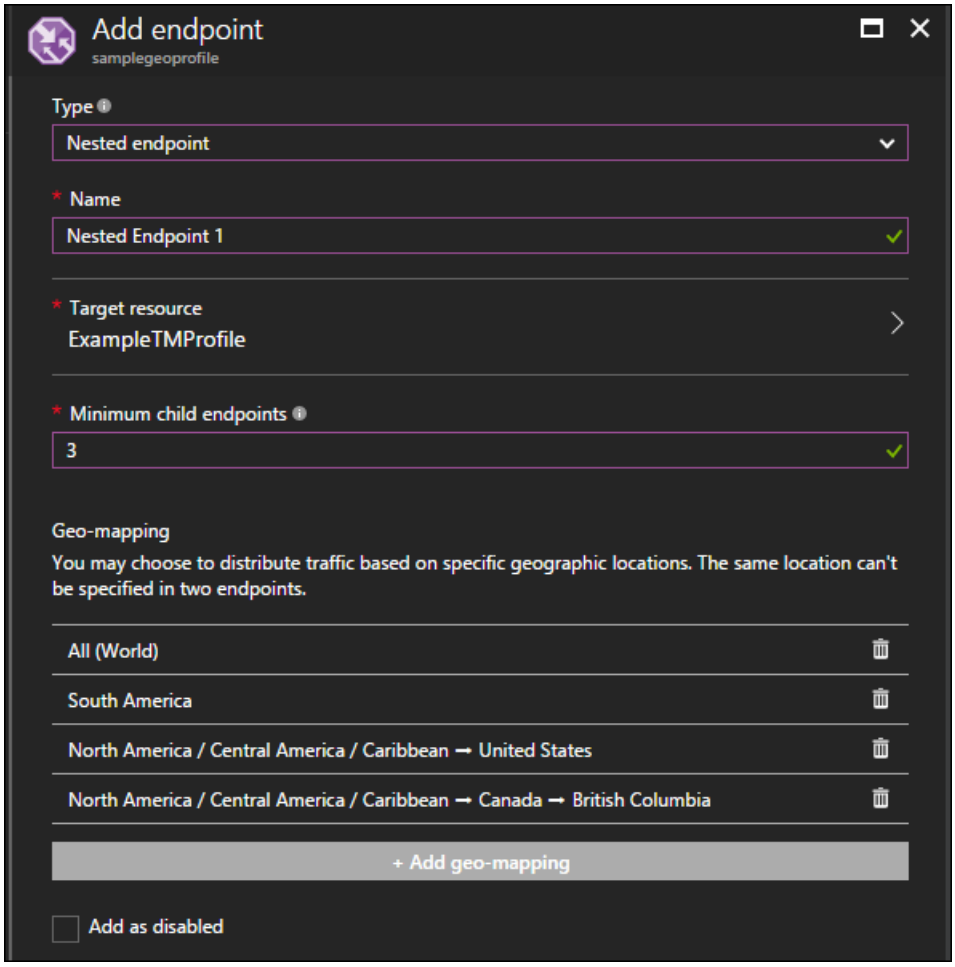
1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. Click **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
3. In the **Create Traffic Manager profile**:
   a. Provide a name for your profile. This name needs to be unique within the trafficmanager.net zone. To access your Traffic Manager profile, you use the DNS name `<profilename>.trafficmanager.net`.
   b. Select the **Geographic** routing method.
   c. Select the subscription you want to create this profile under.
   d. Use an existing resource group or create a new resource group to place this profile under. If you choose to create a new resource group, use the **Resource Group location** dropdown to specify the location of the resource group. This setting refers to the location of the resource group, and has no impact on the Traffic Manager profile that's deployed globally.
   e. After you click **Create**, your Traffic Manager profile is created and deployed globally.



Add endpoints

1. Search for the Traffic Manager profile name you created in the portal's search bar and click on the result when it is shown.
2. Navigate to **Settings** -> **Endpoints** in Traffic Manager.
3. Click **Add** to show the **Add Endpoint**.
4. Click **Add** and in the **Add endpoint** that is displayed, complete as follows:

5. Select **Type** depending upon the type of endpoint you are adding. For geographic routing profiles used in production, we strongly recommend using nested endpoint types containing a child profile with more than one endpoint. For more details, see FAQs about geographic traffic routing methods.
6. Provide a **Name** by which you want to recognize this endpoint.
7. Certain fields on this page depend on the type of endpoint you are adding:
   a. If you are adding an Azure endpoint, select the **Target resource type** and the **Target** based on the resource you want to direct traffic to
   b. If you are adding an **External** endpoint, provide the **Fully-qualified domain name (FQDN)** for your endpoint.
   c. If you are adding a **Nested endpoint**, select the **Target resource** that corresponds to the child profile you want to use and specify the **Minimum child endpoints count**.
8. In the Geo-mapping section, use the drop down to add the regions from where you want traffic to be sent to this endpoint. You must add at least one region, and you can have multiple regions mapped.
9. Repeat this for all endpoints you want to add under this profile



Use the Traffic Manager profile

1. In the portal's search bar, search for the **Traffic Manager profile** name that you created in the preceding section and click on the traffic manager profile in the results that the displayed.
2. Click **Overview**.
3. The **Traffic Manager profile** displays the DNS name of your newly created Traffic Manager profile. This can be used by any clients (for example, by navigating to it using a web browser) to get routed to the right endpoint as determined by the routing type. In the case of geographic routing, Traffic Manager looks at the source IP of the incoming request and determines the region from which it is originating. If that region is mapped to an endpoint, traffic is routed to there. If this region is not mapped to an endpoint, then Traffic Manager returns a NODATA query response.

Next steps

- Learn more about Geographic traffic routing method.
- Learn how to test Traffic Manager settings.

Tutorial: Configure the weighted traffic routing method in Traffic Manager

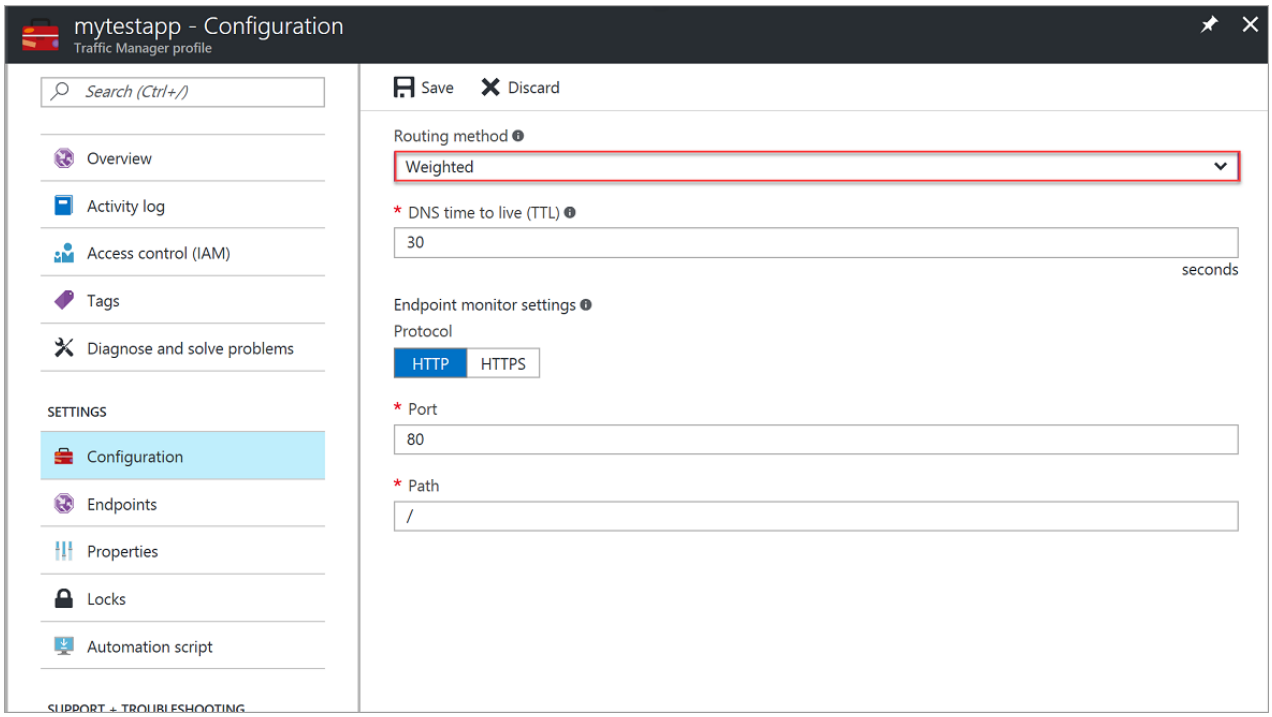
2/1/2020 • 2 minutes to read • Edit Online

A common traffic routing method pattern is to provide a set of identical endpoints, which include cloud services and websites, and send traffic to each equally. The following steps outline how to configure this type of traffic routing method.

> **NOTE**
>
> Azure Web App already provides round-robin load balancing functionality for websites within an Azure Region (which may comprise multiple datacenters). Traffic Manager allows you to distribute traffic across websites in different datacenters.

To configure the weighted traffic routing method

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. In the portal's search bar, search for the **Traffic Manager profiles** and then click the profile name that you want to configure the routing method for.
3. In the **Traffic Manager profile** blade, verify that both the cloud services and websites that you want to include in your configuration are present.
4. In the **Settings** section, click **Configuration**, and in the **Configuration** blade, complete as follows:
   a. For **traffic routing method settings**, verify that the traffic routing method is **Weighted**. If it is not, click **Weighted** from the dropdown list.
   b. Set the **Endpoint monitor settings** identical for all every endpoint within this profile as follows:
      a. Select the appropriate **Protocol**, and specify the **Port** number.
      b. For **Path** type a forward slash /. To monitor endpoints, you must specify a path and filename. A forward slash "/" is a valid entry for the relative path and implies that the file is in the root directory (default).
      c. At the top of the page, click **Save**.
5. Test the changes in your configuration as follows:
   a. In the portal's search bar, search for the Traffic Manager profile name and click the Traffic Manager profile in the results that the displayed.
   b. In the **Traffic Manager** profile blade, click **Overview**.
   c. The **Traffic Manager profile** blade displays the DNS name of your newly created Traffic Manager profile. This can be used by any clients (for example,by navigating to it using a web browser) to get routed to the right endpoint as determined by the routing type. In this case all requests are routed each endpoint in a round-robin fashion.
6. Once your Traffic Manager profile is working, edit the DNS record on your authoritative DNS server to point your company domain name to the Traffic Manager domain name.



Next steps

- Learn about priority traffic routing method.
- Learn about performance traffic routing method.
- Learn about geographic routing method.
- Learn how to test Traffic Manager settings.

Tutorial: Configure priority traffic routing method in Traffic Manager

2/1/2020 • 2 minutes to read • Edit Online

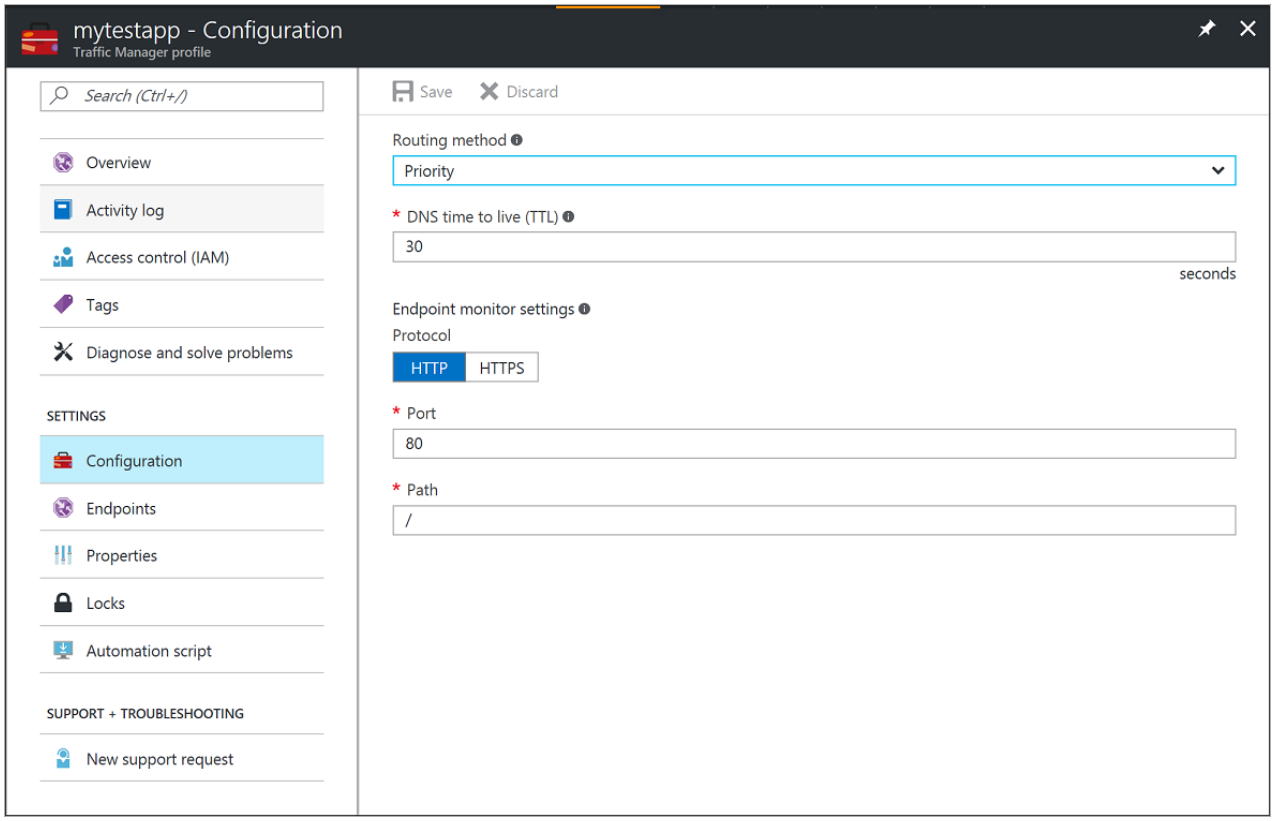
Regardless of the website mode, Azure Websites already provide failover functionality for websites within a datacenter (also known as a region). Traffic Manager provides failover for websites in different datacenters.

A common pattern for service failover is to send traffic to a primary service and provide a set of identical backup services for failover. The following steps explain how to configure this prioritized failover with Azure cloud services and websites:

To configure the priority traffic routing method

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. In the portal's search bar, search for the **Traffic Manager profiles** and then click the profile name that you want to configure the routing method for.
3. In the **Traffic Manager profile** blade, verify that both the cloud services and websites that you want to include in your configuration are present.
4. In the **Settings** section, click **Configuration**, and in the **Configuration** blade, complete as follows:
   a. For **traffic routing method settings**, verify that the traffic routing method is **Priority**. If it is not, click **Priority** from the dropdown list.
   b. Set the **Endpoint monitor settings** identical for all every endpoint within this profile as follows:
      a. Select the appropriate **Protocol**, and specify the **Port** number.
      b. For **Path** type a forward slash /. To monitor endpoints, you must specify a path and filename. A forward slash "/" is a valid entry for the relative path and implies that the file is in the root directory (default).
      c. At the top of the page, click **Save**.
5. In the **Settings** section, click **Endpoints**.
6. In the **Endpoints** blade, review the priority order for your endpoints. When you select the **Priority** traffic routing method, the order of the selected endpoints matters. Verify the priority order of endpoints. The primary endpoint is on top. Double-check on the order it is displayed. all requests will be routed to the first endpoint and if Traffic Manager detects it be unhealthy, the traffic automatically fails over to the next endpoint.
7. To change the endpoint priority order, click the endpoint, and in the **Endpoint** blade that is displayed, click **Edit** and change the **Priority** value as needed.
8. Click **Save** to save change the endpoint settings.
9. After you complete your configuration changes, click **Save** at the bottom of the page.
10. Test the changes in your configuration as follows:
    a. In the portal's search bar, search for the Traffic Manager profile name and click the Traffic Manager profile in the results that the displayed.
    b. In the **Traffic Manager** profile blade, click **Overview**.
    c. The **Traffic Manager profile** blade displays the DNS name of your newly created Traffic Manager profile. This can be used by any clients (for example, by navigating to it using a web browser) to get routed to the right endpoint as determined by the routing type. In this case all requests are routed to the first endpoint and if Traffic Manager detects it be unhealthy, the traffic automatically fails over to the next endpoint.

11. Once your Traffic Manager profile is working, edit the DNS record on your authoritative DNS server to point your company domain name to the Traffic Manager domain name.



Next steps

- Learn about weighted traffic routing method.
- Learn about performance routing method.
- Learn about geographic routing method.
- Learn how to test Traffic Manager settings.

Tutorial: Control traffic routing with weighted endpoints by using Traffic Manager

2/1/2020 • 8 minutes to read • Edit Online

This tutorial describes how to use Azure Traffic Manager to control routing of user traffic between endpoints by using the weighted routing method. In this routing method, you assign weights to each endpoint in the Traffic Manager profile configuration. User traffic is then routed based on the weight assigned to each endpoint. The weight is an integer from 1 to 1,000. The higher the weight value assigned to an endpoint, the higher its priority.

In this tutorial, you learn how to:

- Create two VMs running a basic website on IIS.
- Create two test VMs to view Traffic Manager in action.
- Configure a DNS name for the VMs running IIS.
- Create a Traffic Manager profile.
- Add VM endpoints to the Traffic Manager profile.
- View Traffic Manager in action.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

To see Traffic Manager in action, deploy the following for this tutorial:

- Two instances of basic websites running in different Azure regions: East US and West Europe.
- Two test VMs for testing Traffic Manager: one in East US and the other in West Europe. The test VMs are used to illustrate how Traffic Manager routes user traffic to a website that has higher weight assigned to its endpoint.

Sign in to Azure

Sign in to the Azure portal.

Create websites

In this section, you create two website instances that provide the two service endpoints for the Traffic Manager profile in two Azure regions. To create the two websites, complete the following steps:

1. Create two VMs for running a basic website: one in East US, and the other in West Europe.
2. Install an IIS server on each VM. Update the default webpage that describes the VM name that a user is connected to when visiting the website.

Create VMs for running websites

In this section, you create two VMs (*myIISVMEastUS* and *myIISVMWestEurope*) in the East US and West Europe Azure regions.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

   - **Subscription** > **Resource Group**: Select **Create new** and then type **myResourceGroupTM1**.
   - **Instance Details** > **Virtual machine name**: Type *myIISVMEastUS*.
   - **Instance Details** > **Region**: Select **East US**.

- **Administrator Account** > **Username**: Enter a user name of your choosing.
- **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
- **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
- **Inbound Port Rules** > **Select inbound ports**: Select **RDP** and **HTTP** in the pull down box.

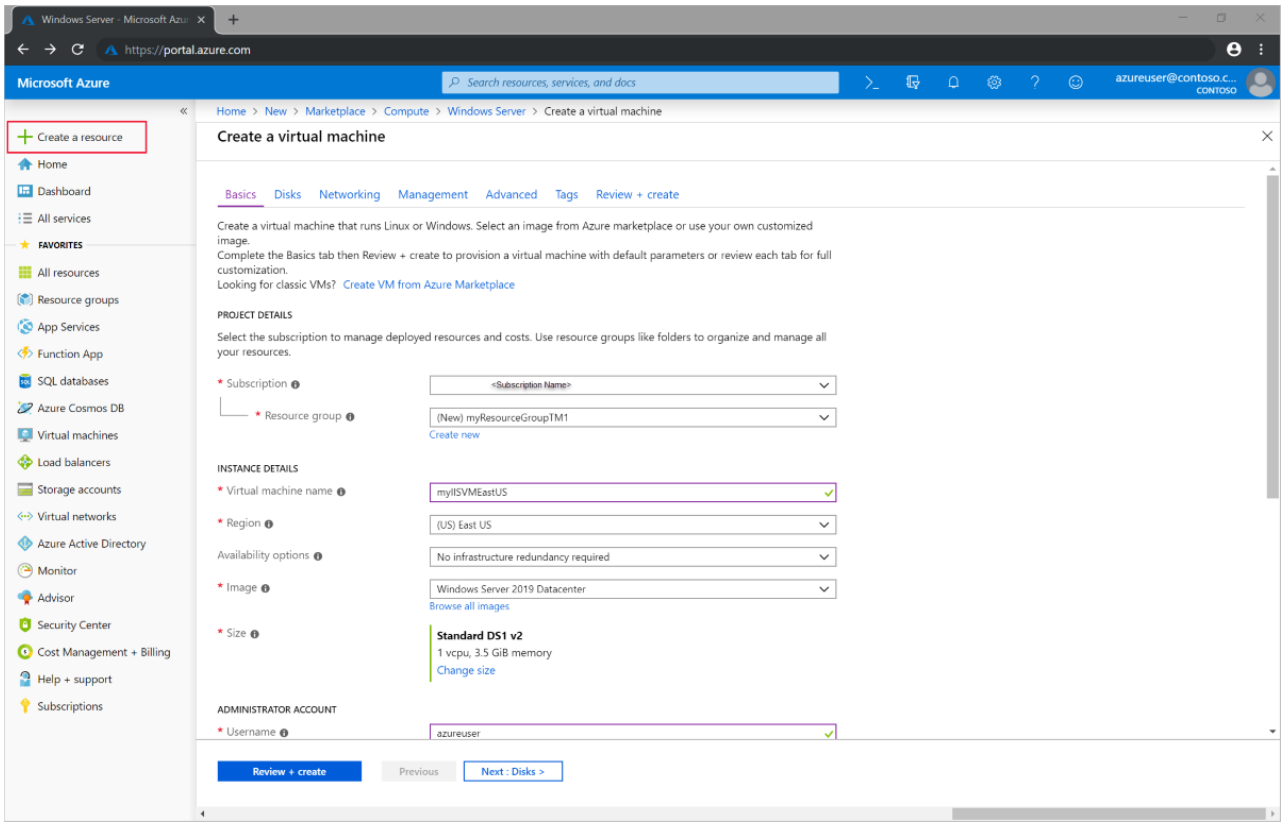
3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.

4. Select **Review + create**.

5. Review the settings, and then click **Create**.

6. Follow the steps to create a second VM named *myIISVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myIISVMEastUS*.

7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.



Install IIS and customize the default webpage

In this section, you install the IIS server on the two VMs myIISVMEastUS and myIISVMWestEurope, and then update the default webpage. The customized webpage shows the name of the VM that you're connecting to when you visit the website from a web browser.

1. Select **All resources** on the left menu. From the resource list, select **myIISVMEastUS** in the **myResourceGroupTM1** resource group.

2. On the **Overview** page, select **Connect**. In **Connect to virtual machine**, select **Download RDP file**.

3. Open the downloaded .rdp file. If you're prompted, select **Connect**. Enter the user name and password that you specified when you created the VM. You might need to select **More choices** > **Use a different account**, to specify the credentials that you entered when you created the VM.

4. Select **OK**.

5. You might receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue** to proceed with the connection.

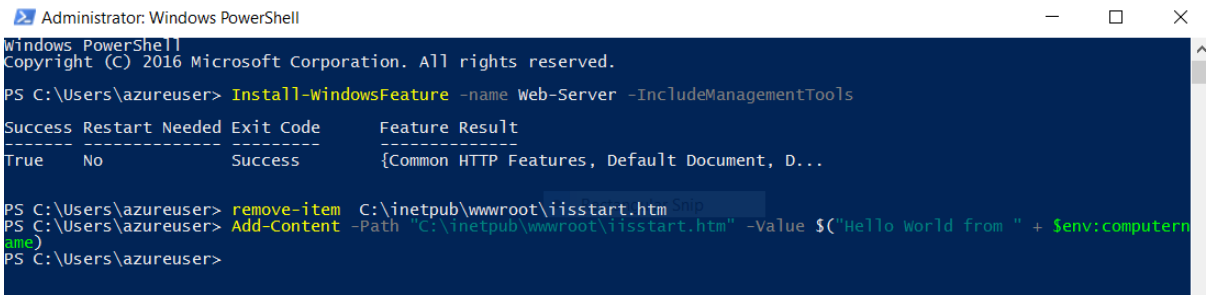
6. On the server desktop, browse to **Windows Administrative Tools** > **Server Manager**.

7. Open Windows PowerShell on VM1. Use the following commands to install the IIS server and update the default .htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default .htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom .htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)
```



8. Close the RDP connection with **myIISVMEastUS**.

9. Repeat steps 1-8. Create an RDP connection with the VM **myIISVMWestEurope** within the **myResourceGroupTM2** resource group, to install IIS and customize its default webpage.

Configure DNS names for the VMs running IIS

Traffic Manager routes user traffic based on the DNS name of the service endpoints. In this section, you configure the DNS names for the IIS servers myIISVMEastUS and myIISVMWestEurope.

1. Select **All resources** on the left menu. From the resource list, select **myIISVMEastUS** in the **myResourceGroupTM1** resource group.
2. On the **Overview** page, under **DNS name**, select **Configure**.
3. On the **Configuration** page, under the DNS name label, add a unique name. Then select **Save**.
4. Repeat steps 1-3 for the VM named **myIISVMWestEurope** in the **myResourceGroupTM2** resource group.

Create a test VM

In this section, you create a VM (*myVMEastUS* and *myVMWestEurope*) in each Azure region (**East US** and **West Europe**). You will use these VMs to test how Traffic Manager routes traffic to the website endpoint that has the higher weight value.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

   - **Subscription** > **Resource Group**: Select **myResourceGroupTM1**.
   - **Instance Details** > **Virtual machine name**: Type *myVMEastUS*.
   - **Instance Details** > **Region**: Select **East US**.
   - **Administrator Account** > **Username**: Enter a user name of your choosing.
   - **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
   - **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
   - **Inbound Port Rules** > **Select inbound ports**: Select **RDP** in the pull down box.

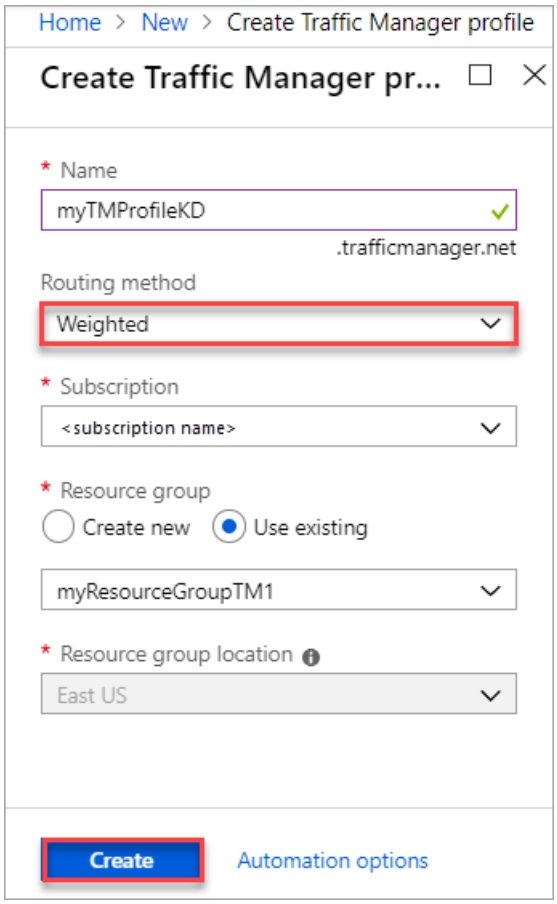
3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.
4. Select **Review + create**.
5. Review the settings, and then click **Create**.
6. Follow the steps to create a second VM named *myVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myVMEastUS*.
7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

Create a Traffic Manager profile

Create a Traffic Manager profile based on the **Weighted** routing method.

1. On the upper-left side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
2. In **Create Traffic Manager profile**, enter or select the following information. Accept the defaults for the other settings, and then select **Create**.

SETTING	VALUE
Name	Enter a unique name within the trafficmanager.net zone. It results in the DNS name trafficmanager.net, which is used to access your Traffic Manager profile.
Routing method	Select the **Weighted** routing method.
Subscription	Select your subscription.
Resource group	Select **Use existing** and then select **myResourceGroupTM1**.



Add Traffic Manager endpoints

Add the two VMs running the IIS servers myIISVMEastUS and myIISVMWestEurope, to route user traffic to them.

1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section. Select the profile in the results that are displayed.
2. In **Traffic Manager profile**, in the **Settings** section, select **Endpoints** > **Add**.
3. Enter or select the following information. Accept the defaults for the other settings, and then select **OK**.

SETTING	VALUE
Type	Enter the Azure endpoint.
Name	Enter **myEastUSEndpoint**.
Target resource type	Select **Public IP address**.
Target resource	Choose a public IP address to show the listing of resources with public IP addresses under the same subscription. In **Resource**, select the public IP address named **myIISVMEastUS-ip**. This is the public IP address of the IIS server VM in East US.
Weight	Enter **100**.

4. Repeat steps 2 and 3 to add another endpoint named **myWestEuropeEndpoint** for the public IP address **myIISVMWestEurope-ip**. This address is associated with the IIS server VM named myIISVMWestEurope. For **Weight**, enter **25**.

5. When the addition of both endpoints is complete, they're displayed in the Traffic Manager profile along with their monitoring status as **Online**.

Test the Traffic Manager profile

To view Traffic Manager in action, complete the following steps:

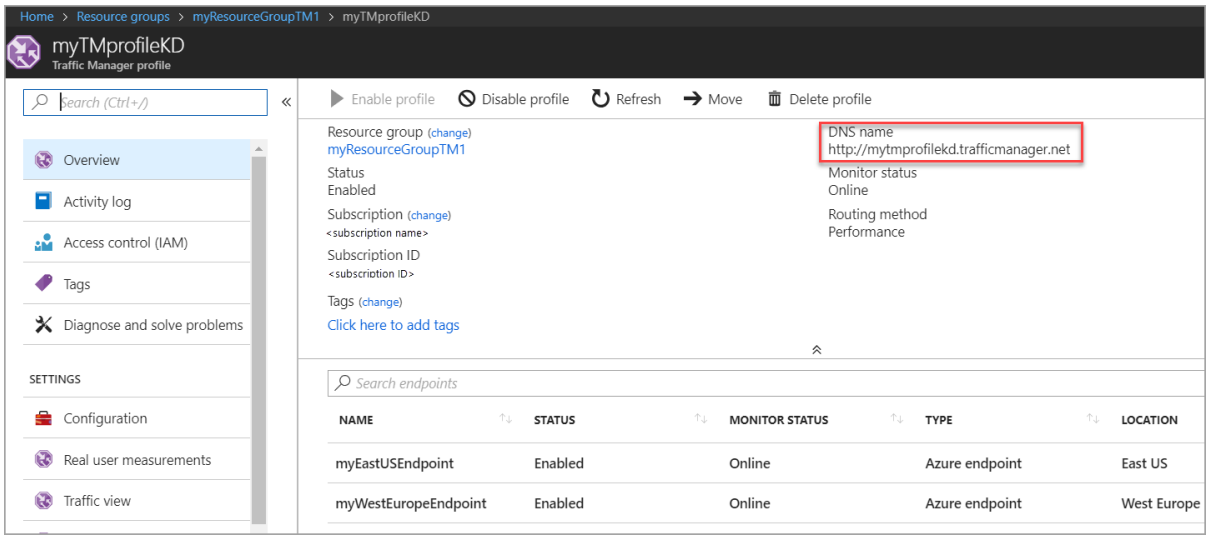
1. Determine the DNS name of your Traffic Manager profile.
2. View Traffic Manager in action.

Determine DNS name of Traffic Manager profile

In this tutorial, for simplicity, you use the DNS name of the Traffic Manager profile to visit the websites.

You can determine the DNS name of the Traffic Manager profile as follows:

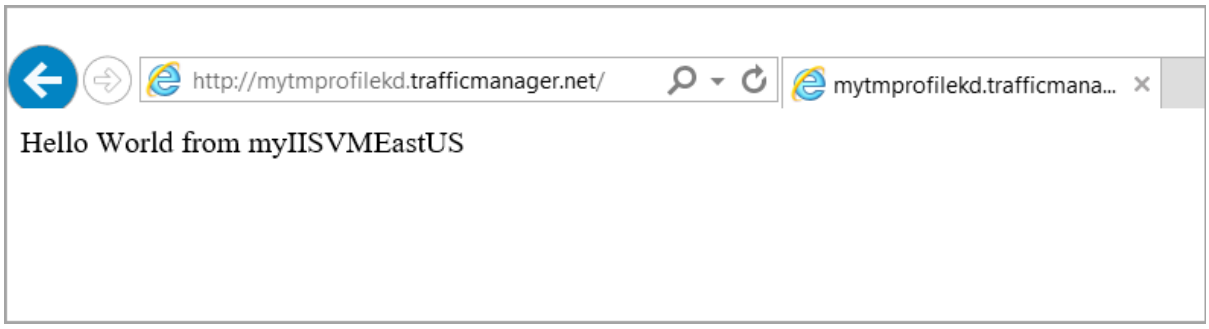
1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section. In the results that are displayed, select the Traffic Manager profile.
2. Select **Overview**.
3. The Traffic Manager profile displays its DNS name. In production deployments, you configure a vanity domain name to point to the Traffic Manager domain name, by using a DNS CNAME record.



View Traffic Manager in action

In this section, you can see Traffic Manager in action.

1. Select **All resources** on the left menu. From the resource list, select **myVMEastUS** in the **myResourceGroupTM1** resource group.
2. On the **Overview** page, select **Connect**. In **Connect to virtual machine**, select **Download RDP file**.
3. Open the downloaded .rdp file. If you're prompted, select **Connect**. Enter the user name and password that you specified when creating the VM. You might need to select **More choices** > **Use a different account**, to specify the credentials that you entered when you created the VM.
4. Select **OK**.
5. You might receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue** to proceed with the connection.
6. In a web browser on the VM myVMEastUS, enter the DNS name of your Traffic Manager profile to view your website. You're routed to website hosted on the IIS server myIISVMEastUS because it's assigned a higher weight of **100**. The IIS server myIISVMWestEurope is assigned a lower endpoint weight value of **25**.



7. Repeat steps 1-6 on the VM myVMWestEurope to see the weighted website response.

Delete the Traffic Manager profile

When you no longer need the resource groups that you created in this tutorial, you can delete them. To do so, select the resource group (**ResourceGroupTM1** or **ResourceGroupTM2**), and then select **Delete**.

Next steps

Route traffic to specific endpoints based on the user's geographic location

Tutorial: Direct traffic to specific endpoints based on user subnet using Traffic Manager

2/11/2020 • 10 minutes to read • Edit Online

This article describes how to configure the subnet traffic-routing method. The **Subnet** traffic-routing method allows you to map a set of IP address ranges to specific endpoints and when a request is received by Traffic Manager, it inspects the source IP of the request and returns the endpoint associated with it.

In this tutorial, using subnet routing, depending on the IP address of the user's query, traffic is either routed to an internal website or a production website.

In this tutorial, you learn how to:

- Create two VMs running a basic website on IIS
- Create two test VMs to view Traffic Manager in action
- Configure DNS name for the VMs running IIS
- Create a Traffic Manager profile for routing traffic based on user's subnet
- Add VM endpoints to the Traffic Manager profile
- View Traffic Manager in action

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

In order to see the Traffic Manager in action, this tutorial requires that you deploy the following:

- two basic websites running in different Azure regions - **East US** (serves as internal website) and **West Europe** (serves as production website).
- two test VMs for testing the Traffic Manager - one VM in **East US** and the second VM in **West Europe**.

The test VMs are used to illustrate how Traffic Manager routes user traffic to the internal website or the production website based on subnet from where the user query originates.

Sign in to Azure

Sign in to the Azure portal at https://portal.azure.com.

Create websites

In this section, you create two website instances that provide the two service endpoints for the Traffic Manager profile in two Azure regions. Creating the two websites includes the following steps:

1. Create two VMs for running a basic website - one in **East US**, and the other in **West Europe**.
2. Install IIS server on each VM and update the default website page that describes the VM name that a user is connected to when visiting the website.

Create VMs for running websites

In this section, you create two VMs *myIISVMEastUS* and *myIISVMWestEurope* in the **East US** and **West Europe** Azure regions.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

- **Subscription** > **Resource Group**: Select **Create new** and then type **myResourceGroupTM1**.
- **Instance Details** > **Virtual machine name**: Type *myIISVMEastUS*.
- **Instance Details** > **Region**: Select **East US**.
- **Administrator Account** > **Username**: Enter a user name of your choosing.
- **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
- **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
- **Inbound Port Rules** > **Select inbound ports**: Select **RDP** and **HTTP** in the pull down box.

3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.
4. Select **Review + create**.
5. Review the settings, and then click **Create**.
6. Follow the steps to create a second VM named *myIISVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myIISVMEastUS*.
7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

Install IIS and customize the default web page

In this section, you install the IIS server on the two VMs - *myIISVMEastUS* & *myIISVMWestEurope*, and then update the default website page. The customized website page shows the name of the VM that you are connecting to when you visit the website from a web browser.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.
3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.
5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.
6. On the server desktop, navigate to **Windows Administrative Tools**>**Server Manager**.
7. Launch Windows PowerShell on VM *myIISVMEastUS*, and using the following commands to install IIS server and update the default htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from my " + $env:computername)
```

8. Close the RDP connection with *myIISVMEastUS* VM.
9. Repeat steps 1-6 with by creating an RDP connection with the VM *myIISVMWestEurope* within the *myResourceGroupTM2* resource group to install IIS and customize its default web page.

10. Launch Windows PowerShell on *myIISVMWestEurope* VM, and using the following commands to install IIS server and update the default htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from my " + $env:computername)
```

Configure DNS names for the VMs running IIS

Traffic Manager routes user traffic based on DNS name of the service endpoints. In this section, you configure the DNS names for the IIS servers - *myIISVMEastUS* and *myIISVMWestEurope*.

1. Click **All resources** in the left-hand menu, and then from the resources list, select *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, under **DNS name**, select **Configure**.
3. On the **Configuration** page, under DNS name label, add a unique name, and then select **Save**.
4. Repeat steps 1-3, for the VM named *myIISVMWestEurope* that is located in the *myResourceGroupTM2* resource group.

Create test VMs

In this section, you create a VM (*myVMEastUS* and *myVMWestEurope*) in each Azure region (**East US** and **West Europe**). You will use these VMs to test how Traffic Manager routes user traffic based on the subnet of the user's query.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2019 Datacenter**.

2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:

   - **Subscription** > **Resource Group**: Select **myResourceGroupTM1**.
   - **Instance Details** > **Virtual machine name**: Type *myVMEastUS*.
   - **Instance Details** > **Region**: Select **East US**.
   - **Administrator Account** > **Username**: Enter a user name of your choosing.
   - **Administrator Account** > **Password**: Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
   - **Inbound Port Rules** > **Public inbound ports**: Select **Allow selected ports**.
   - **Inbound Port Rules** > **Select inbound ports**: Select **RDP** in the pull down box.

3. Select the **Management** tab, or select **Next: Disks**, then **Next: Networking**, then **Next: Management**. Under **Monitoring**, set **Boot diagnostics** to **Off**.

4. Select **Review + create**.

5. Review the settings, and then click **Create**.

6. Follow the steps to create a second VM named *myVMWestEurope*, with a **Resource group** name of *myResourceGroupTM2*, a **location** of *West Europe*, and all the other settings the same as *myVMEastUS*.

7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

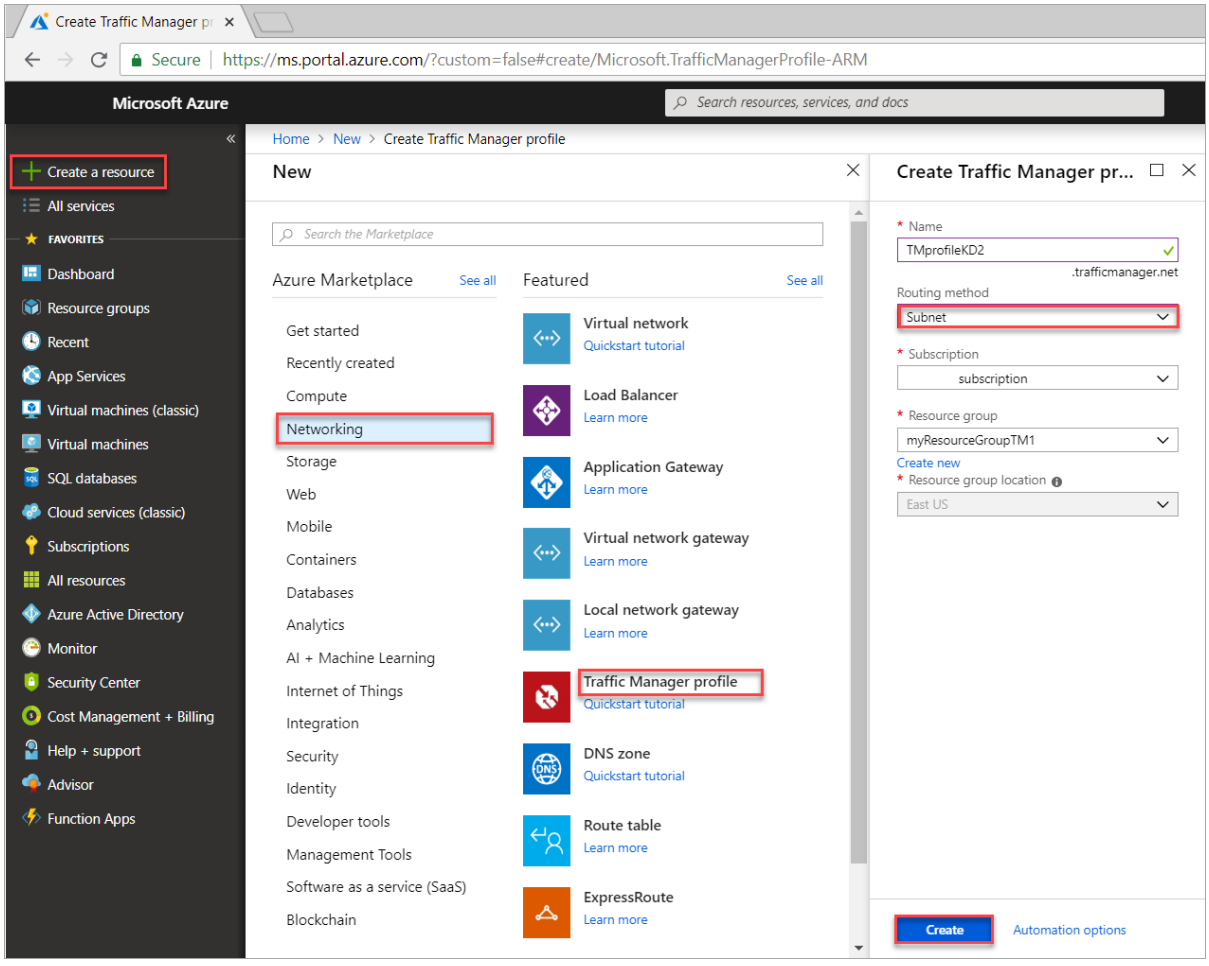
Create a Traffic Manager profile

Create a Traffic Manager profile that allows you to return specific endpoints based on the source IP of the request.

1. On the top left-hand side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.

2. In the **Create Traffic Manager profile**, enter or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	This name needs to be unique within the trafficmanager.net zone and results in the DNS name, trafficmanager.net that is used to access your Traffic Manager profile.
Routing method	Select the **Subnet** routing method.
Subscription	Select your subscription.
Resource group	Select **Existing** and enter *myResourceGroupTM1*.



Add Traffic Manager endpoints

Add the two VMs running the IIS servers - *myIISVMEastUS* & *myIISVMWestEurope* to route user traffic based on the subnet of the user's query.

1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section and select the profile in the results that the displayed.

2. In **Traffic Manager profile**, in the **Settings** section, click **Endpoints**, and then click **Add**.

3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

SETTING	VALUE
Type	Azure endpoint
Name	myInternalWebSiteEndpoint
Target resource type	Public IP Address
Target resource	**Choose a Public IP address** to show the listing of resources with Public IP addresses under the same subscription. In **Resource**, select the public IP address named *myIISVMEastUS-ip*. This is the public IP address of the IIS server VM in East US.
Subnet routing settings	Add the IP address of *myVMEastUS* test VM. Any user query originating from this VM will be directed to the *myInternalWebSiteEndpoint*.

4. Repeat steps 2 and 3 to add another endpoint named *myProdWebsiteEndpoint* for the public IP address *myIISVMWestEurope-ip* that is associated with the IIS server VM named *myIISVMWestEurope*. For **Subnet routing settings**, add the IP address of the test VM - *myVMWestEurope*. Any user query from this test VM will be routed to the endpoint - *myProdWebsiteEndpoint*.

5. When the addition of both endpoints is complete, they are displayed in **Traffic Manager profile** along with their monitoring status as **Online**.

Test Traffic Manager profile

In this section, you test how the Traffic Manager routes user traffic from a given subnet to a specific endpoint. To view the Traffic Manager in action, complete the following steps:

1. Determine the DNS name of your Traffic Manager profile.
2. View Traffic Manager in action as follows:
   - From the test VM (*myVMEastUS*) that is located in the **East US** region, in a web browser, browse to the DNS name of your Traffic Manager profile.
   - From the test VM (*myVMWestEurope*) that is located in the **West Europe** region, in a web browser, browse to the DNS name of your Traffic Manager profile.

Determine DNS name of Traffic Manager profile

In this tutorial, for simplicity, you use the DNS name of the Traffic Manager profile to visit the websites.

You can determine the DNS name of the Traffic Manager profile as follows:

1. In the portal's search bar, search for the **Traffic Manager profile** name that you created in the preceding section. In the results that are displayed, click the traffic manager profile.
2. Click **Overview**.
3. The **Traffic Manager profile** displays the DNS name of your newly created Traffic Manager profile. In production deployments, you configure a vanity domain name to point to the Traffic Manager domain name, using a DNS CNAME record.

View Traffic Manager in action

In this section, you can see the Traffic Manager is action.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.
3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.
5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.
6. In a web browser on the VM *myVMEastUS*, type the DNS name of your Traffic Manager profile to view your website. Since the VM *myVMEastUS* IP address is associated with the endpoint *myInternalWebsiteEndpoint*, the web browser launches the Test website server - *myIISVMEastUS*.
7. Next, connect to the VM *myVMWestEurope* located in **West Europe** using steps 1-5 and browse to the Traffic Manager profile domain name from this VM. Since the VM *myVMWestEurope* IP address is associated with the endpoint *myProductionWebsiteEndpoint*, the web browser launches the Test website server - *myIISVMWestEurope*.

Delete the Traffic Manager profile

When no longer needed, delete the resource groups (**ResourceGroupTM1** and **ResourceGroupTM2**). To do so, select the resource group (**ResourceGroupTM1** or **ResourceGroupTM2**), and then select **Delete**.

Next steps

- Learn about weighted traffic routing method.
- Learn about priority routing method.
- Learn about geographic routing method.

Tutorial: Configure an alias record to support apex domain names with Traffic Manager

2/11/2020 • 4 minutes to read • Edit Online

You can create an alias record for your domain name apex to reference an Azure Traffic Manager profile. An example is contoso.com. Instead of using a redirecting service, you configure Azure DNS to reference a Traffic Manager profile directly from your zone.

In this tutorial, you learn how to:

- Create a host VM and network infrastructure.
- Create a Traffic Manager profile.
- Create an alias record.
- Test the alias record.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

You must have a domain name available that you can host in Azure DNS to test with. You must have full control of this domain. Full control includes the ability to set the name server (NS) records for the domain.

For instructions on how to host your domain in Azure DNS, see Tutorial: Host your domain in Azure DNS.

The example domain used for this tutorial is contoso.com, but use your own domain name.

Create the network infrastructure

First, create a virtual network and a subnet to place your web servers in.

1. Sign in to the Azure portal at https://portal.azure.com.
2. In the upper left in the portal, select **Create a resource**. Enter *resource group* in the search box, and create a resource group named **RG-DNS-Alias-TM**.
3. Select **Create a resource** > **Networking** > **Virtual network**.
4. Create a virtual network named **VNet-Servers**. Place it in the **RG-DNS-Alias-TM** resource group, and name the subnet **SN-Web**.

Create two web server virtual machines

1. Select **Create a resource** > **Windows Server 2016 VM**.
2. Enter **Web-01** for the name, and place the VM in the **RG-DNS-Alias-TM** resource group. Enter a username and a password, and select **OK**.
3. For **Size**, select an SKU with 8-GB RAM.
4. For **Settings**, select the **VNet-Servers** virtual network and the **SN-Web** subnet.
5. Select **Public IP address**. Under **Assignment**, select **Static**, and then select **OK**.
6. For public inbound ports, select **HTTP** > **HTTPS** > **RDP (3389)**, and then select **OK**.
7. On the **Summary** page, select **Create**. This procedure takes a few minutes to finish.

Repeat this procedure to create another virtual machine named **Web-02**.

Add a DNS label

The public IP addresses need a DNS label to work with Traffic Manager.

1. In the **RG-DNS-Alias-TM** resource group, select the **Web-01-ip** public IP address.
2. Under **Settings**, select **Configuration**.
3. In the DNS name label text box, enter **web01pip**.
4. Select **Save**.

Repeat this procedure for the **Web-02-ip** public IP address by using **web02pip** for the DNS name label.

Install IIS

Install IIS on both **Web-01** and **Web-02**.

1. Connect to **Web-01**, and sign in.
2. On the **Server Manager** dashboard, select **Add roles and features**.
3. Select **Next** three times. On the **Server Roles** page, select **Web Server (IIS)**.
4. Select **Add Features**, and select **Next**.
5. Select **Next** four times. Then select **Install**. This procedure takes a few minutes to finish.
6. When the installation finishes, select **Close**.
7. Open a web browser. Browse to **localhost** to verify that the default IIS web page appears.

Repeat this procedure to install IIS on **Web-02**.

Create a Traffic Manager profile

1. Open the **RG-DNS-Alias-TM** resource group, and select the **Web-01-ip** Public IP address. Note the IP address for later use. Repeat this step for the **Web-02-ip** public IP address.
2. Select **Create a resource** > **Networking** > **Traffic Manager profile**.
3. For the name, enter **TM-alias-test**. Place it in the **RG-DNS-Alias-TM** resource group.
4. Select **Create**.
5. After deployment finishes, select **Go to resource**.
6. On the Traffic Manager profile page, under **Settings**, select **Endpoints**.
7. Select **Add**.
8. For **Type**, select **External endpoint**, and for **Name**, enter **EP-Web01**.
9. In the **Fully qualified domain name (FQDN) or IP** text box, enter the IP address for **Web-01-ip** that you noted previously.
10. Select the same **Location** as your other resources, and then select **OK**.

Repeat this procedure to add the **Web-02** endpoint by using the IP address you noted previously for **Web-02-ip**.

Create an alias record

Create an alias record that points to the Traffic Manager profile.

1. Select your Azure DNS zone to open the zone.
2. Select **Record set**.
3. Leave the **Name** text box empty to represent the domain name apex. An example is contoso.com.
4. Leave the **Type** as an **A** record.
5. Select the **Alias Record Set** check box.
6. Select **Choose Azure service**, and select the **TM-alias-test** Traffic Manager profile.

Test the alias record

1. From a web browser, browse to your domain name apex. An example is contoso.com. You see the IIS default web page. Close the web browser.
2. Shut down the **Web-01** virtual machine. Wait a few minutes for it to completely shut down.
3. Open a new web browser, and browse to your domain name apex again.
4. You see the IIS default web page again, because Traffic Manager handled the situation and directed traffic to **Web-02**.

Clean up resources

When you no longer need the resources created for this tutorial, delete the **RG-DNS-Alias-TM** resource group.

Next steps

In this tutorial, you created an alias record to use your apex domain name to reference a Traffic Manager profile. To learn about Azure DNS and web apps, continue with the tutorial for web apps.

Host load-balanced web apps at the zone apex

Azure CLI samples for Traffic Manager

2/1/2020 • 2 minutes to read • Edit Online

The following table includes links to bash scripts for Traffic Manager built using the Azure CLI.

TITLE	DESCRIPTION
Direct traffic across multiple regions for high application availability	Creates two app service plans, two web apps, a traffic manager profile, and two traffic manager endpoints.

Azure PowerShell samples for Traffic Manager

2/1/2020 • 2 minutes to read • Edit Online

The following table includes links to Traffic Manager scripts built using Azure PowerShell.

TITLE	DESCRIPTION
Direct traffic across multiple regions for high application availability	Creates two app service plans, two web apps, a traffic manager profile, and two traffic manager endpoints.

Traffic Manager routing methods

2/1/2020 • 14 minutes to read • Edit Online

Azure Traffic Manager supports six traffic-routing methods to determine how to route network traffic to the various service endpoints. For any profile, Traffic Manager applies the traffic-routing method associated to it to each DNS query it receives. The traffic-routing method determines which endpoint is returned in the DNS response.

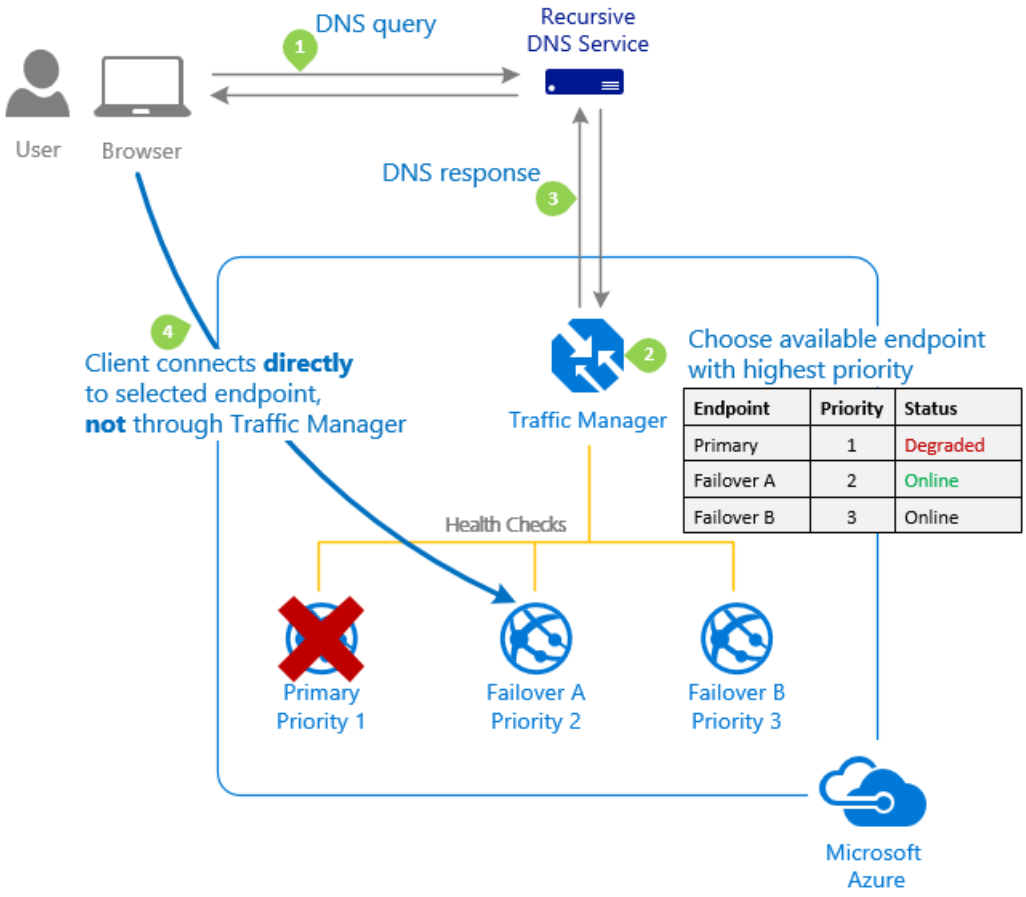
The following traffic routing methods are available in Traffic Manager:

- **Priority:** Select **Priority** when you want to use a primary service endpoint for all traffic, and provide backups in case the primary or the backup endpoints are unavailable.
- **Weighted:** Select **Weighted** when you want to distribute traffic across a set of endpoints, either evenly or according to weights, which you define.
- **Performance:** Select **Performance** when you have endpoints in different geographic locations and you want end users to use the "closest" endpoint in terms of the lowest network latency.
- **Geographic:** Select **Geographic** so that users are directed to specific endpoints (Azure, External, or Nested) based on which geographic location their DNS query originates from. This empowers Traffic Manager customers to enable scenarios where knowing a user's geographic region and routing them based on that is important. Examples include complying with data sovereignty mandates, localization of content & user experience and measuring traffic from different regions.
- **Multivalue:** Select **MultiValue** for Traffic Manager profiles that can only have IPv4/IPv6 addresses as endpoints. When a query is received for this profile, all healthy endpoints are returned.
- **Subnet:** Select **Subnet** traffic-routing method to map sets of end-user IP address ranges to a specific endpoint within a Traffic Manager profile. When a request is received, the endpoint returned will be the one mapped for that request's source IP address.

All Traffic Manager profiles include monitoring of endpoint health and automatic endpoint failover. For more information, see Traffic Manager Endpoint Monitoring. A single Traffic Manager profile can use only one traffic routing method. You can select a different traffic routing method for your profile at any time. Changes are applied within one minute, and no downtime is incurred. Traffic-routing methods can be combined by using nested Traffic Manager profiles. Nesting enables sophisticated and flexible traffic-routing configurations that meet the needs of larger, complex applications. For more information, see nested Traffic Manager profiles.

Priority traffic-routing method

Often an organization wants to provide reliability for its services by deploying one or more backup services in case their primary service goes down. The 'Priority' traffic-routing method allows Azure customers to easily implement this failover pattern.



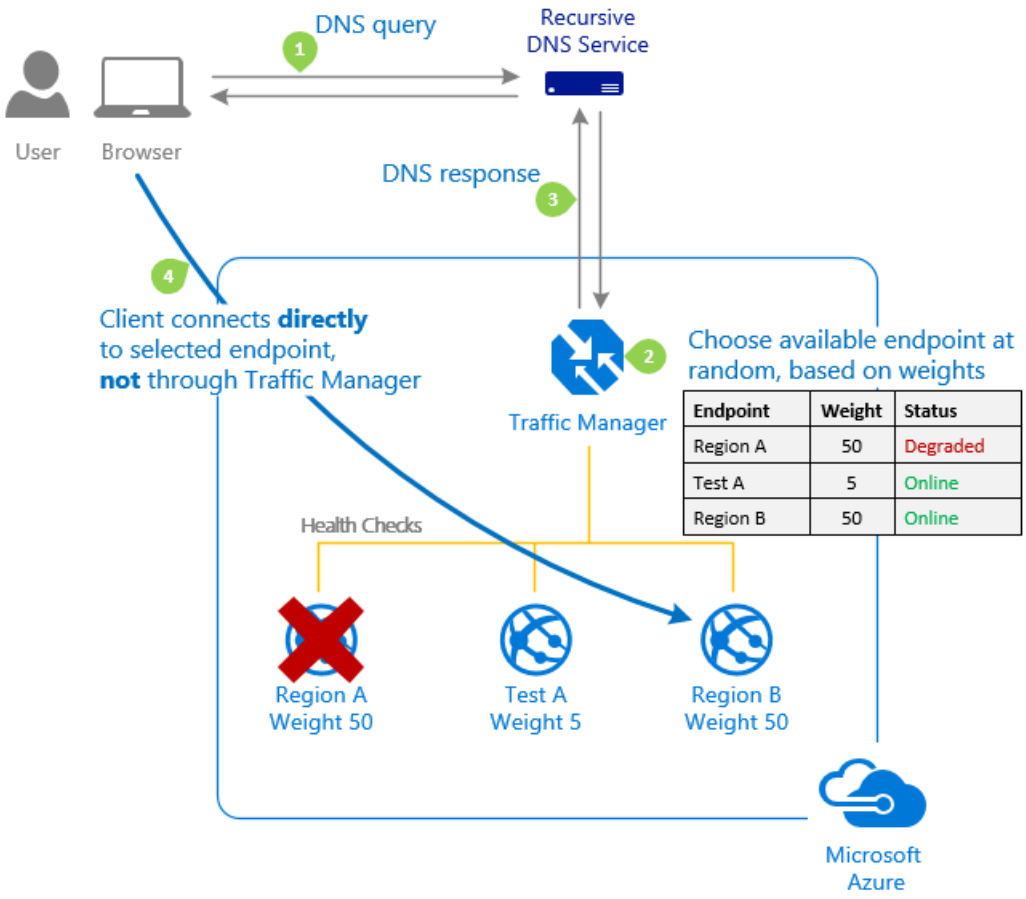
The Traffic Manager profile contains a prioritized list of service endpoints. By default, Traffic Manager sends all traffic to the primary (highest-priority) endpoint. If the primary endpoint is not available, Traffic Manager routes the traffic to the second endpoint. If both the primary and secondary endpoints are not available, the traffic goes to the third, and so on. Availability of the endpoint is based on the configured status (enabled or disabled) and the ongoing endpoint monitoring.

Configuring endpoints

With Azure Resource Manager, you configure the endpoint priority explicitly using the 'priority' property for each endpoint. This property is a value between 1 and 1000. Lower values represent a higher priority. Endpoints cannot share priority values. Setting the property is optional. When omitted, a default priority based on the endpoint order is used.

Weighted traffic-routing method

The 'Weighted' traffic-routing method allows you to distribute traffic evenly or to use a pre-defined weighting.



In the Weighted traffic-routing method, you assign a weight to each endpoint in the Traffic Manager profile configuration. The weight is an integer from 1 to 1000. This parameter is optional. If omitted, Traffic Managers uses a default weight of '1'. The higher weight, the higher the priority.

For each DNS query received, Traffic Manager randomly chooses an available endpoint. The probability of choosing an endpoint is based on the weights assigned to all available endpoints. Using the same weight across all endpoints results in an even traffic distribution. Using higher or lower weights on specific endpoints causes those endpoints to be returned more or less frequently in the DNS responses.

The weighted method enables some useful scenarios:

- Gradual application upgrade: Allocate a percentage of traffic to route to a new endpoint, and gradually increase the traffic over time to 100%.
- Application migration to Azure: Create a profile with both Azure and external endpoints. Adjust the weight of the endpoints to prefer the new endpoints.
- Cloud-bursting for additional capacity: Quickly expand an on-premises deployment into the cloud by putting it behind a Traffic Manager profile. When you need extra capacity in the cloud, you can add or enable more endpoints and specify what portion of traffic goes to each endpoint.

In addition to using the Azure portal, you can configure weights using Azure PowerShell, CLI, and the REST APIs.

It is important to understand that DNS responses are cached by clients and by the recursive DNS servers that the clients use to resolve DNS names. This caching can have an impact on weighted traffic distributions. When the number of clients and recursive DNS servers is large, traffic distribution works as expected. However, when the number of clients or recursive DNS servers is small, caching can significantly skew the traffic distribution.

Common use cases include:

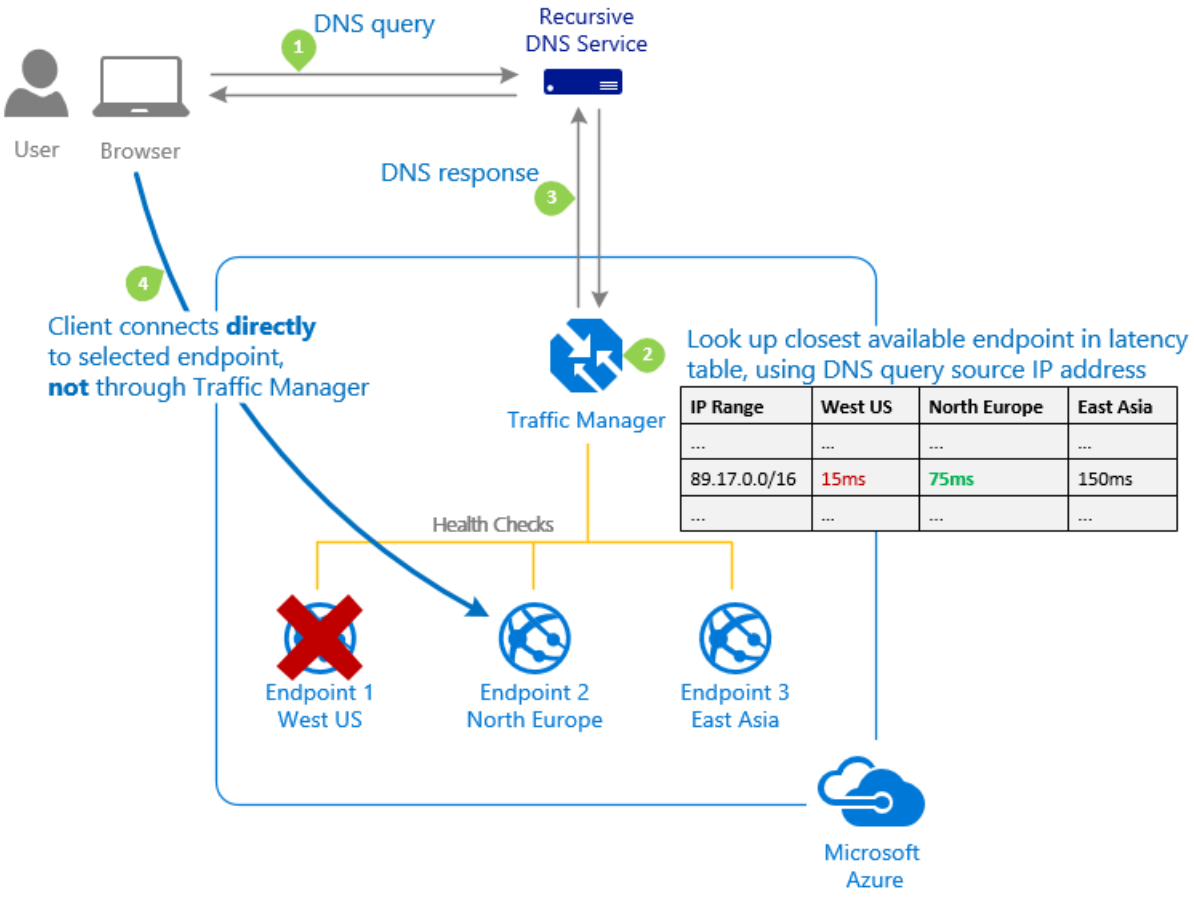
- Development and testing environments

- Application-to-application communications
- Applications aimed at a narrow user-base that share a common recursive DNS infrastructure (for example, employees of company connecting through a proxy)

These DNS caching effects are common to all DNS-based traffic routing systems, not just Azure Traffic Manager. In some cases, explicitly clearing the DNS cache may provide a workaround. In other cases, an alternative traffic-routing method may be more appropriate.

Performance traffic-routing method

Deploying endpoints in two or more locations across the globe can improve the responsiveness of many applications by routing traffic to the location that is 'closest' to you. The 'Performance' traffic-routing method provides this capability.



The 'closest' endpoint is not necessarily closest as measured by geographic distance. Instead, the 'Performance' traffic-routing method determines the closest endpoint by measuring network latency. Traffic Manager maintains an Internet Latency Table to track the round-trip time between IP address ranges and each Azure datacenter.

Traffic Manager looks up the source IP address of the incoming DNS request in the Internet Latency Table. Traffic Manager then chooses an available endpoint in the Azure datacenter that has the lowest latency for that IP address range, and returns that endpoint in the DNS response.

As explained in How Traffic Manager Works, Traffic Manager does not receive DNS queries directly from clients. Rather, DNS queries come from the recursive DNS service that the clients are configured to use. Therefore, the IP address used to determine the 'closest' endpoint is not the client's IP address, but it is the IP address of the recursive DNS service. In practice, this IP address is a good proxy for the client.

Traffic Manager regularly updates the Internet Latency Table to account for changes in the global Internet and new Azure regions. However, application performance varies based on real-time variations in load across the

Internet. Performance traffic-routing does not monitor load on a given service endpoint. However, if an endpoint becomes unavailable, Traffic Manager does not include it in DNS query responses.

Points to note:

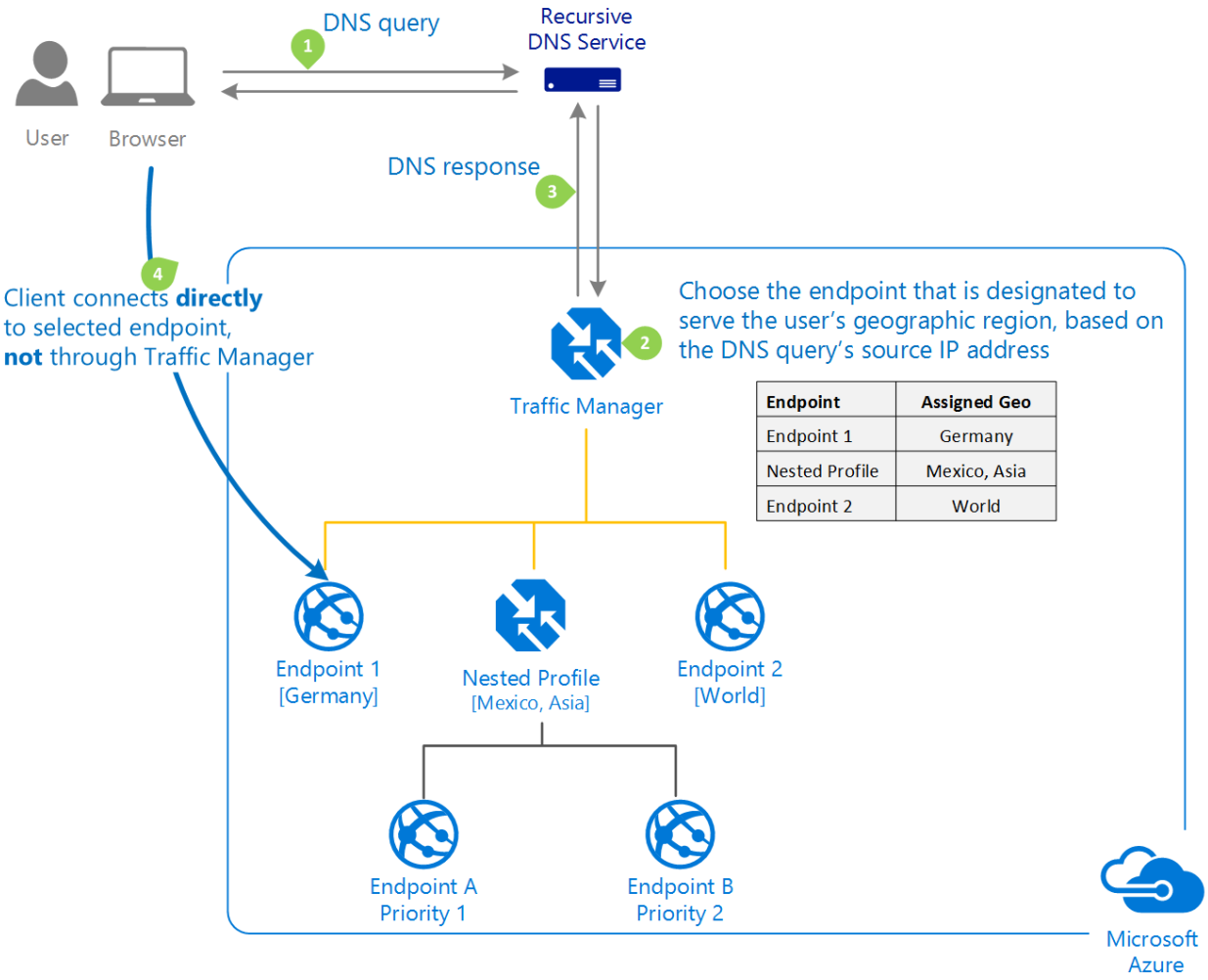
- If your profile contains multiple endpoints in the same Azure region, then Traffic Manager distributes traffic evenly across the available endpoints in that region. If you prefer a different traffic distribution within a region, you can use nested Traffic Manager profiles.
- If all enabled endpoints in the closest Azure region are degraded, Traffic Manager moves traffic to the endpoints in the next closest Azure region. If you want to define a preferred failover sequence, use nested Traffic Manager profiles.
- When using the Performance traffic routing method with external endpoints or nested endpoints, you need to specify the location of those endpoints. Choose the Azure region closest to your deployment. Those locations are the values supported by the Internet Latency Table.
- The algorithm that chooses the endpoint is deterministic. Repeated DNS queries from the same client are directed to the same endpoint. Typically, clients use different recursive DNS servers when traveling. The client may be routed to a different endpoint. Routing can also be affected by updates to the Internet Latency Table. Therefore, the Performance traffic-routing method does not guarantee that a client is always routed to the same endpoint.
- When the Internet Latency Table changes, you may notice that some clients are directed to a different endpoint. This routing change is more accurate based on current latency data. These updates are essential to maintain the accuracy of Performance traffic-routing as the Internet continually evolves.

Geographic traffic-routing method

Traffic Manager profiles can be configured to use the Geographic routing method so that users are directed to specific endpoints (Azure, External or Nested) based on which geographic location their DNS query originates from. This empowers Traffic Manager customers to enable scenarios where knowing a user's geographic region and routing them based on that is important. Examples include complying with data sovereignty mandates, localization of content & user experience and measuring traffic from different regions. When a profile is configured for geographic routing, each endpoint associated with that profile needs to have a set of geographic regions assigned to it. A geographic region can be at following levels of granularity

- World– any region
- Regional Grouping – for example, Africa, Middle East, Australia/Pacific etc.
- Country/Region – for example, Ireland, Peru, Hong Kong SAR etc.
- State/Province – for example, USA-California, Australia-Queensland, Canada-Alberta etc. (note: this granularity level is supported only for states / provinces in Australia, Canada, and USA).

When a region or a set of regions is assigned to an endpoint, any requests from those regions gets routed only to that endpoint. Traffic Manager uses the source IP address of the DNS query to determine the region from which a user is querying from – usually this is the IP address of the local DNS resolver doing the query on behalf of the user.



Traffic Manager reads the source IP address of the DNS query and decides which geographic region it is originating from. It then looks to see if there is an endpoint that has this geographic region mapped to it. This lookup starts at the lowest granularity level (State/Province where it is supported, else at the Country/Region level) and goes all the way up to the highest level, which is **World**. The first match found using this traversal is designated as the endpoint to return in the query response. When matching with a Nested type endpoint, an endpoint within that child profile is returned, based on its routing method. The following points are applicable to this behavior:

- A geographic region can be mapped only to one endpoint in a Traffic Manager profile when the routing type is Geographic Routing. This ensures that routing of users is deterministic, and customers can enable scenarios that require unambiguous geographic boundaries.
- If a user's region comes under two different endpoints' geographic mapping, Traffic Manager selects the endpoint with the lowest granularity and does not consider routing requests from that region to the other endpoint. For example, consider a Geographic Routing type profile with two endpoints - Endpoint1 and Endpoint2. Endpoint1 is configured to receive traffic from Ireland and Endpoint2 is configured to receive traffic from Europe. If a request originates from Ireland, it is always routed to Endpoint1.
- Since a region can be mapped only to one endpoint, Traffic Manager returns it regardless of whether the endpoint is healthy or not.

  > **IMPORTANT**
  > It is strongly recommended that customers using the geographic routing method associate it with the Nested type endpoints that has child profiles containing at least two endpoints within each.

- If an endpoint match is found and that endpoint is in the **Stopped** state, Traffic Manager returns a

NODATA response. In this case, no further lookups are made higher up in the geographic region hierarchy. This behavior is also applicable for nested endpoint types when the child profile is in the **Stopped** or **Disabled** state.

- If an endpoint displays a **Disabled** status, it won't be included in the region matching process. This behavior is also applicable for nested endpoint types when the endpoint is in the **Disabled** state.
- If a query is coming from a geographic region that has no mapping in that profile, Traffic Manager returns a NODATA response. Therefore, it is strongly recommended that customers use geographic routing with one endpoint, ideally of type Nested with at least two endpoints within the child profile, with the region **World** assigned to it. This also ensures that any IP addresses that do not map to a region are handled.

As explained in How Traffic Manager Works, Traffic Manager does not receive DNS queries directly from clients. Rather, DNS queries come from the recursive DNS service that the clients are configured to use. Therefore, the IP address used to determine the region is not the client's IP address, but it is the IP address of the recursive DNS service. In practice, this IP address is a good proxy for the client.

FAQs

- What are some use cases where geographic routing is useful?
- How do I decide if I should use Performance routing method or Geographic routing method?
- What are the regions that are supported by Traffic Manager for geographic routing?
- How does traffic manager determine where a user is querying from?
- Is it guaranteed that Traffic Manager can correctly determine the exact geographic location of the user in every case?
- Does an endpoint need to be physically located in the same region as the one it is configured with for geographic routing?
- Can I assign geographic regions to endpoints in a profile that is not configured to do geographic routing?
- Why am I getting an error when I try to change the routing method of an existing profile to Geographic?
- Why is it strongly recommended that customers create nested profiles instead of endpoints under a profile with geographic routing enabled?
- Are there any restrictions on the API version that supports this routing type?

Multivalue traffic-routing method

The **Multivalue** traffic-routing method allows you to get multiple healthy endpoints in a single DNS query response. This enables the caller to do client-side retries with other endpoints in the event of a returned endpoint being unresponsive. This pattern can increase the availability of a service and reduce the latency associated with a new DNS query to obtain a healthy endpoint. MultiValue routing method works only if all the endpoints of type 'External' and are specified as IPv4 or IPv6 addresses. When a query is received for this profile, all healthy endpoints are returned and are subject to a configurable maximum return count.

FAQs

- What are some use cases where MultiValue routing is useful?
- How many endpoints are returned when MultiValue routing is used?
- Will I get the same set of endpoints when MultiValue routing is used?

Subnet traffic-routing method

The **Subnet** traffic-routing method allows you to map a set of end user IP address ranges to specific endpoints in a profile. After that, if Traffic Manager receives a DNS query for that profile, it will inspect the source IP address of that request (in most cases this will be the outgoing IP address of the DNS resolver used by the caller), determine which endpoint it is mapped to and will return that endpoint in the query response.

The IP address to be mapped to an endpoint can be specified as CIDR ranges (e.g. 1.2.3.0/24) or as an address range (e.g. 1.2.3.4-5.6.7.8). The IP ranges associated with an endpoint need to be unique within that profile and cannot have an overlap with the IP address set of a different endpoint in the same profile. If you define an endpoint with no address range, that functions as a fallback and take traffic from any remaining subnets. If no fallback endpoint is included, Traffic Manager sends a NODATA response for any undefined ranges. It is therefore highly recommended that you either define a fallback endpoint, or else ensure that all possible IP ranges are specified across your endpoints.

Subnet routing can be used to deliver a different experience for users connecting from a specific IP space. For example, using subnet routing, a customer can make all requests from their corporate office be routed to a different endpoint where they might be testing an internal only version of their app. Another scenario is if you want to provide a different experience to users connecting from a specific ISP (For example, block users from a given ISP).

FAQs

- What are some use cases where subnet routing is useful?
- How does Traffic Manager know the IP address of the end user?
- How can I specify IP addresses when using Subnet routing?
- How can I specify a fallback endpoint when using Subnet routing?
- What happens if an endpoint is disabled in a Subnet routing type profile?

Next steps

Learn how to develop high-availability applications using Traffic Manager endpoint monitoring

Country/Region hierarchy used by Azure Traffic Manager for geographic traffic routing method

2/1/2020 • 2 minutes to read • Edit Online

This article lists the countries and regions used by the **Geographic** traffic routing method in Azure Traffic Manager. You can also obtain this information programmatically by calling the Azure Traffic Manager's REST API.

- WORLD(World)
  - GEO-EU(Europe)
    - AD(Andorra)
    - AL(Albania)
    - AT(Austria)
    - AX(Åland Islands)
    - BA(Bosnia and Herzegovina)
    - BE(Belgium)
    - BG(Bulgaria)
    - BY(Belarus)
    - CH(Switzerland)
    - CY(Cyprus)
    - CZ(Czech Republic)
    - DE(Germany)
    - DK(Denmark)
    - EE(Estonia)
    - ES(Spain)
    - FI(Finland)
    - FO(Faroe Islands)
    - FR(France)
    - GB(United Kingdom)
    - GG(Guernsey)
    - GI(Gibraltar)
    - GR(Greece)
    - HR(Croatia)
    - HU(Hungary)

- IE(Ireland)
- IM(Isle of Man)
- IS(Iceland)
- IT(Italy)
- JE(Jersey)
- LI(Liechtenstein)
- LT(Lithuania)
- LU(Luxembourg)
- LV(Latvia)
- MC(Monaco)
- MD(Moldova)
- ME(Montenegro)
- MK(North Macedonia)
- MT(Malta)
- NL(Netherlands)
- NO(Norway)
- PL(Poland)
- PT(Portugal)
- RO(Romania)
- RS(Serbia)
- RU(Russia)
- SE(Sweden)
- SI(Slovenia)
- SJ(Svalbard)
- SK(Slovakia)
- SM(San Marino)
- UA(Ukraine)
  - Region of Crimea
- VA(Vatican City)
- XJ(Jan Mayen)
- XK(Kosovo)

- GEO-ME(Middle East)
  - AE(United Arab Emirates)
  - BH(Bahrain)

- IL(Israel)
- IQ(Iraq)
- IR(Iran)
- JO(Jordan)
- KW(Kuwait)
- LB(Lebanon)
- OM(Oman)
- PS(Palestinian Authority)
- QA(Qatar)
- SY(Syria)
- SA(Saudi Arabia)
- TR(Turkey)
- YE(Yemen)

- GEO-NA(North America / Central America / Caribbean)
  - AG(Antigua and Barbuda)
  - AI(Anguilla)
  - AW(Aruba)
  - BB(Barbados)
  - BL(Saint Barthélemy)
  - BM(Bermuda)
  - BQ(Bonaire)
  - BS(Bahamas)
  - BZ(Belize)
  - CA(Canada)
    - CA-AB(Alberta)
    - CA-BC(British Columbia)
    - CA-MB(Manitoba)
    - CA-NB(New Brunswick)
    - CA-NL(Newfoundland and Labrador)
    - CA-NS(Nova Scotia)
    - CA-NT(Northwest Territories)
    - CA-NU(Nunavut)
    - CA-ON(Ontario)

- CA-PE(Prince Edward Island)
- CA-QC(Québec)
- CA-SK(Saskatchewan)
- CA-YT(Yukon Territory)
- CR(Costa Rica)
- CU(Cuba)
- CW(CuraÃ§ao)
- DM(Dominica)
- DO(Dominican Republic)
- GD(Grenada)
- GL(Greenland)
- GP(Guadeloupe)
- GT(Guatemala)
- HN(Honduras)
- HT(Haiti)
- JM(Jamaica)
- KN(Saint Kitts and Nevis)
- KY(Cayman Islands)
- LC(Saint Lucia)
- MF(Saint Martin)
- MQ(Martinique)
- MS(Montserrat)
- MX(Mexico)
- NI(Nicaragua)
- PA(Panama)
- PM(Saint Pierre and Miquelon)
- PR(Puerto Rico)
- SV(El Salvador)
- SX(Sint Maarten)
- TC(Turks and Caicos Islands)
- TT(Trinidad and Tobago)
- UM(U.S. Outlying Islands)
- US(United States)

- US-AK(Alaska)
- US-AL(Alabama)
- US-AR(Arkansas)
- US-AZ(Arizona)
- US-CA(California)
- US-CO(Colorado)
- US-CT(Connecticut)
- US-DC(District of Columbia)
- US-DE(Delaware)
- US-FL(Florida)
- US-GA(Georgia)
- US-HI(Hawaii)
- US-IA(Iowa)
- US-ID(Idaho)
- US-IL(Illinois)
- US-IN(Indiana)
- US-KS(Kansas)
- US-KY(Kentucky)
- US-LA(Louisiana)
- US-MA(Massachusetts)
- US-MD(Maryland)
- US-ME(Maine)
- US-MI(Michigan)
- US-MN(Minnesota)
- US-MO(Missouri)
- US-MS(Mississippi)
- US-MT(Montana)
- US-NC(North Carolina)
- US-ND(North Dakota)
- US-NE(Nebraska)
- US-NH(New Hampshire)
- US-NJ(New Jersey)
- US-NM(New Mexico)

        - US-NV(Nevada)
        - US-NY(New York)
        - US-OH(Ohio)
        - US-OK(Oklahoma)
        - US-OR(Oregon)
        - US-PA(Pennsylvania)
        - US-RI(Rhode Island)
        - US-SC(South Carolina)
        - US-SD(South Dakota)
        - US-TN(Tennessee)
        - US-TX(Texas)
        - US-UT(Utah)
        - US-VA(Virginia)
        - US-VT(Vermont)
        - US-WA(Washington)
        - US-WI(Wisconsin)
        - US-WV(West Virginia)
        - US-WY(Wyoming)
    - VC(Saint Vincent and the Grenadines)
    - VG(British Virgin Islands)
    - VI(U.S. Virgin Islands)
    - XE(Sint Eustatius)
    - XS(Saba)
- GEO-AS(Asia)
    - AF(Afghanistan)
    - AM(Armenia)
    - AZ(Azerbaijan)
    - BD(Bangladesh)
    - BN(Brunei)
    - BT(Bhutan)
    - CC(Cocos (Keeling) Islands)
    - CN(China)
    - CX(Christmas Island)

- GE(Georgia)
- HK(Hong Kong SAR)
- ID(Indonesia)
- IN(India)
- IO(British Indian Ocean Territory)
- JP(Japan)
- KG(Kyrgyzstan)
- KH(Cambodia)
- KP(North Korea)
- KR(Korea)
- KZ(Kazakhstan)
- LA(Laos)
- LK(Sri Lanka)
- MM(Myanmar)
- MN(Mongolia)
- MO(Macao SAR)
- MV(Maldives)
- MY(Malaysia)
- NP(Nepal)
- PH(Philippines)
- PK(Pakistan)
- SG(Singapore)
- TH(Thailand)
- TJ(Tajikistan)
- TL(Timor_Leste)
- TM(Turkmenistan)
- TW(Taiwan)
- UZ(Uzbekistan)
- VN(Vietnam)

- GEO-AF(Africa)
  - AO(Angola)
  - BF(Burkina Faso)
  - BI(Burundi)

- BJ(Benin)
- BV(Bouvet Island)
- BW(Botswana)
- CD(Congo (DRC))
- CF(Central African Republic)
- CI(Côte d'Ivoire)
- CM(Cameroon)
- CV(Cabo Verde)
- DJ(Djibouti)
- DZ(Algeria)
- EG(Egypt)
- ER(Eritrea)
- ET(Ethiopia)
- GA(Gabon)
- GH(Ghana)
- GM(Gambia)
- GN(Guinea)
- GQ(Equatorial Guinea)
- GW(Guinea_Bissau)
- KE(Kenya)
- KM(Comoros)
- LR(Liberia)
- LS(Lesotho)
- LY(Libya)
- MA(Morocco)
- MG(Madagascar)
- ML(Mali)
- MR(Mauritania)
- MU(Mauritius)
- MW(Malawi)
- MZ(Mozambique)
- NA(Namibia)
- NE(Niger)

- NG(Nigeria)
- RE(Réunion)
- RW(Rwanda)
- SC(Seychelles)
- SD(Sudan)
- SH(St Helena, Ascension, Tristan da Cunha)
- SL(Sierra Leone)
- SN(Senegal)
- SO(Somalia)
- SS(South Sudan)
- ST(São Tomé and Príncipe)
- SZ(Swaziland)
- TD(Chad)
- TF(French Southern Territories)
- TG(Togo)
- TN(Tunisia)
- TZ(Tanzania)
- UG(Uganda)
- YT(Mayotte)
- ZA(South Africa)
- ZM(Zambia)
- ZW(Zimbabwe)

- GEO-AN(Antarctica)
  - AQ(Antarctica)
- GEO-SA(South America)
  - AR(Argentina)
  - BO(Bolivia)
  - BR(Brazil)
  - CL(Chile)
  - CO(Colombia)
  - EC(Ecuador)
  - FK(Falkland Islands)
  - GF(French Guiana)
  - GS(South Georgia and South Sandwich Islands)

- GY(Guyana)
- PE(Peru)
- PY(Paraguay)
- SR(Suriname)
- UY(Uruguay)
- VE(Venezuela)

- GEO-AP(Australia / Pacific)
  - AS(American Samoa)
  - AU(Australia)
    - AU-ACT(Australian Capital Territory)
    - AU-NSW(New South Wales)
    - AU-NT(Northern Territory)
    - AU-QLD(Queensland)
    - AU-SA(South Australia)
    - AU-TAS(Tasmania)
    - AU-VIC(Victoria)
    - AU-WA(Western Australia)
  - CK(Cook Islands)
  - FJ(Fiji)
  - FM(Micronesia)
  - GU(Guam)
  - HM(Heard Island and McDonald Islands)
  - KI(Kiribati)
  - MH(Marshall Islands)
  - MP(Northern Mariana Islands)
  - NC(New Caledonia)
  - NF(Norfolk Island)
  - NR(Nauru)
  - NU(Niue)
  - NZ(New Zealand)
  - PF(French Polynesia)
  - PG(Papua New Guinea)
  - PN(Pitcairn Islands)

- PW(Palau)
- SB(Solomon Islands)
- TK(Tokelau)
- TO(Tonga)
- TV(Tuvalu)
- VU(Vanuatu)
- WF(Wallis and Futuna)
- WS(Samoa)

Next steps

- Learn more about Geographic traffic routing method in Azure Traffic Manager.

Nested Traffic Manager profiles

2/1/2020 • 5 minutes to read • Edit Online

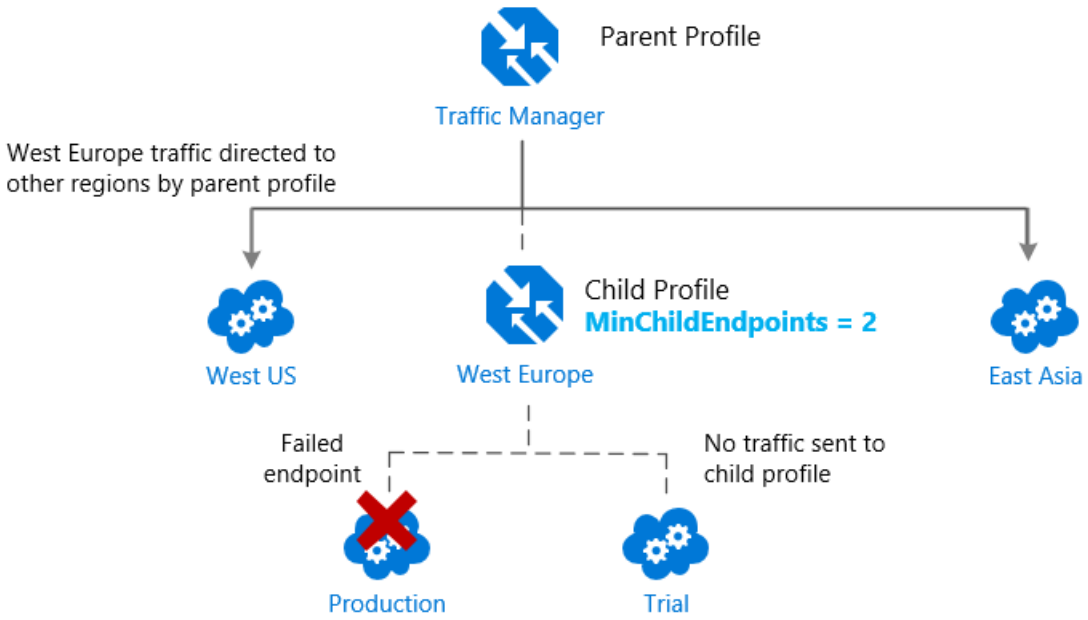
Traffic Manager includes a range of traffic-routing methods that allow you to control how Traffic Manager chooses which endpoint should receive traffic from each end user. For more information, see Traffic Manager traffic-routing methods.

Each Traffic Manager profile specifies a single traffic-routing method. However, there are scenarios that require more sophisticated traffic routing than the routing provided by a single Traffic Manager profile. You can nest Traffic Manager profiles to combine the benefits of more than one traffic-routing method. Nested profiles allow you to override the default Traffic Manager behavior to support larger and more complex application deployments.

The following examples illustrate how to use nested Traffic Manager profiles in various scenarios.

Example 1: Combining 'Performance' and 'Weighted' traffic routing

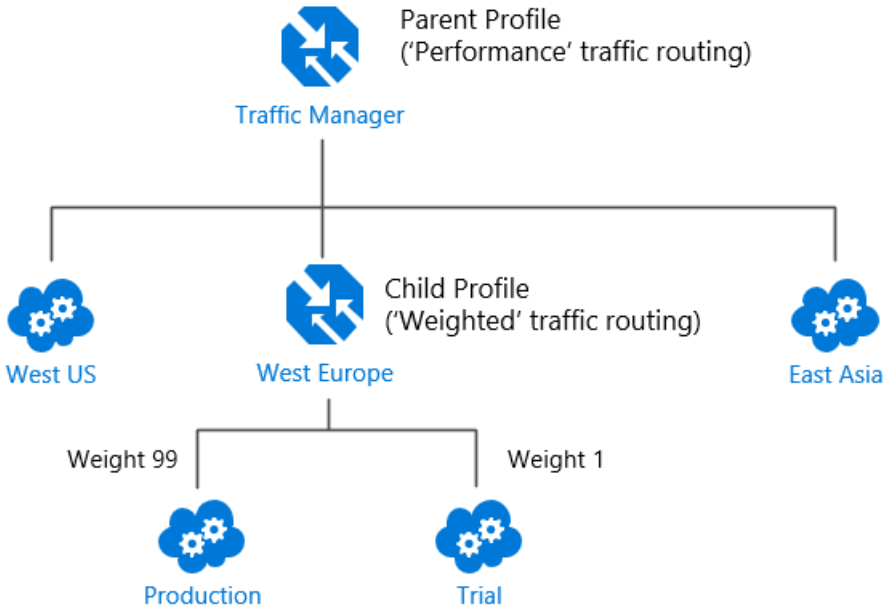
Suppose that you deployed an application in the following Azure regions: West US, West Europe, and East Asia. You use Traffic Manager's 'Performance' traffic-routing method to distribute traffic to the region closest to the user.



Now, suppose you wish to test an update to your service before rolling it out more widely. You want to use the 'weighted' traffic-routing method to direct a small percentage of traffic to your test deployment. You set up the test deployment alongside the existing production deployment in West Europe.

You cannot combine both 'Weighted' and 'Performance traffic-routing in a single profile. To support this scenario, you create a Traffic Manager profile using the two West Europe endpoints and the 'Weighted' traffic-routing method. Next, you add this 'child' profile as an endpoint to the 'parent' profile. The parent profile still uses the Performance traffic-routing method and contains the other global deployments as endpoints.

The following diagram illustrates this example:



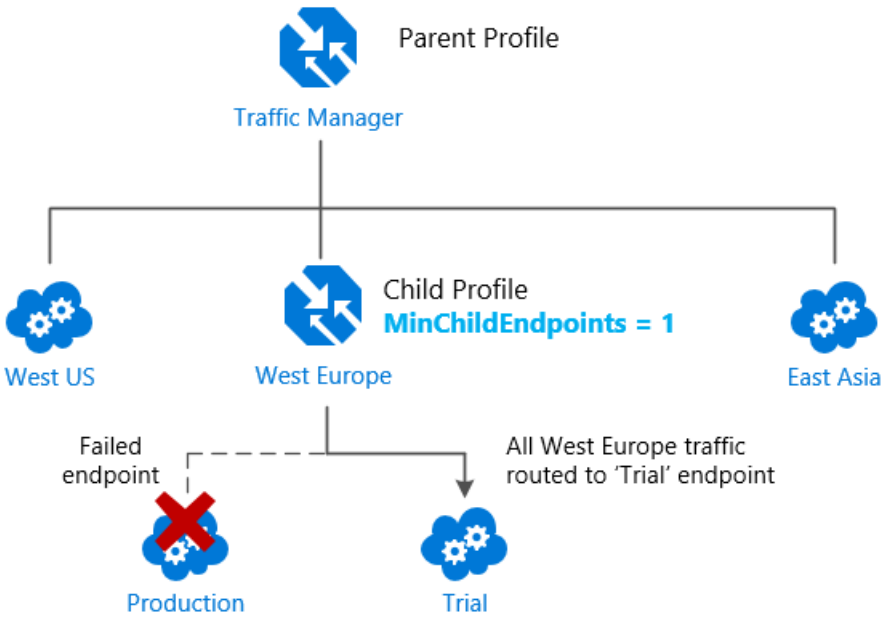
In this configuration, traffic directed via the parent profile distributes traffic across regions normally. Within West Europe, the nested profile distributes traffic to the production and test endpoints according to the weights assigned.

When the parent profile uses the 'Performance' traffic-routing method, each endpoint must be assigned a location. The location is assigned when you configure the endpoint. Choose the Azure region closest to your deployment. The Azure regions are the location values supported by the Internet Latency Table. For more information, see Traffic Manager 'Performance' traffic-routing method.

Example 2: Endpoint monitoring in Nested Profiles

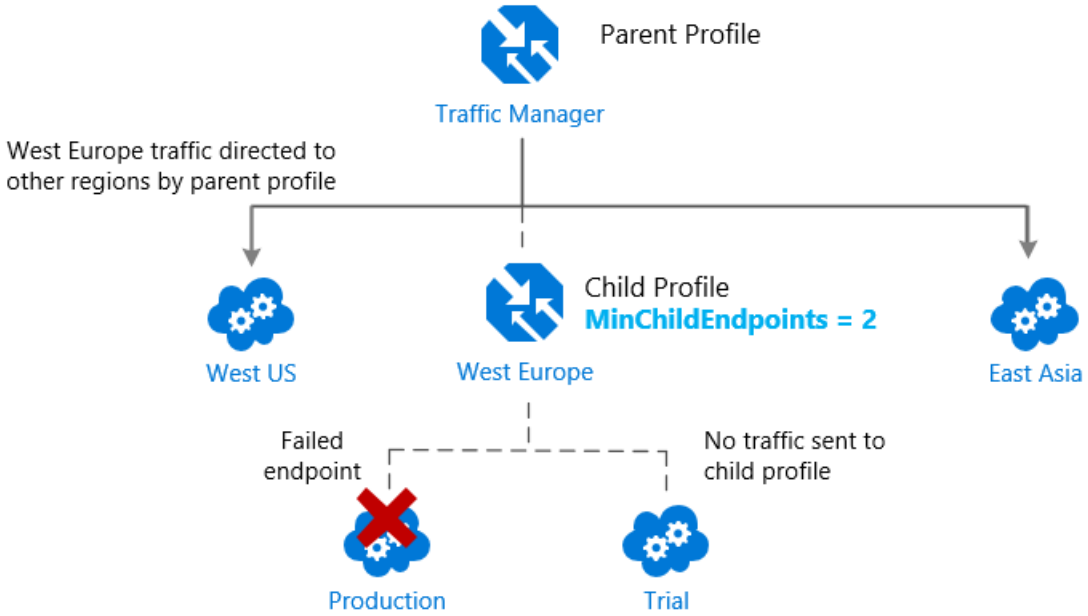
Traffic Manager actively monitors the health of each service endpoint. If an endpoint is unhealthy, Traffic Manager directs users to alternative endpoints to preserve the availability of your service. This endpoint monitoring and failover behavior applies to all traffic-routing methods. For more information, see Traffic Manager Endpoint Monitoring. Endpoint monitoring works differently for nested profiles. With nested profiles, the parent profile doesn't perform health checks on the child directly. Instead, the health of the child profile's endpoints is used to calculate the overall health of the child profile. This health information is propagated up the nested profile hierarchy. The parent profile uses this aggregated health to determine whether to direct traffic to the child profile. See the FAQ for full details on health monitoring of nested profiles.

Returning to the previous example, suppose the production deployment in West Europe fails. By default, the 'child' profile directs all traffic to the test deployment. If the test deployment also fails, the parent profile determines that the child profile should not receive traffic since all child endpoints are unhealthy. Then, the parent profile distributes traffic to the other regions.



You might be happy with this arrangement. Or you might be concerned that all traffic for West Europe is now going to the test deployment instead of a limited subset traffic. Regardless of the health of the test deployment, you want to fail over to the other regions when the production deployment in West Europe fails. To enable this failover, you can specify the 'MinChildEndpoints' parameter when configuring the child profile as an endpoint in the parent profile. The parameter determines the minimum number of available endpoints in the child profile. The default value is '1'. For this scenario, you set the MinChildEndpoints value to 2. Below this threshold, the parent profile considers the entire child profile to be unavailable and directs traffic to the other endpoints.

The following figure illustrates this configuration:



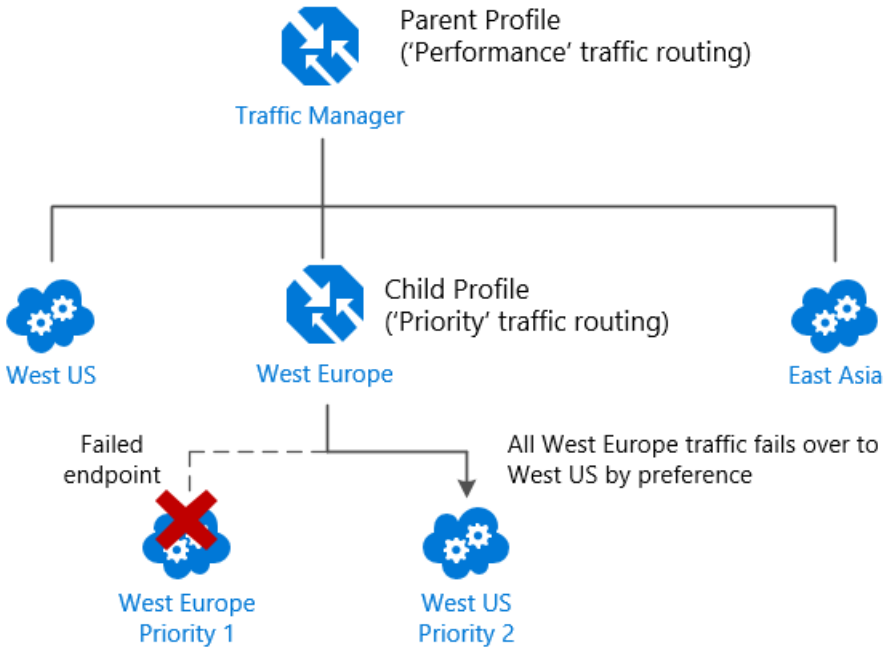
NOTE

The 'Priority' traffic-routing method distributes all traffic to a single endpoint. Thus there is little purpose in a MinChildEndpoints setting other than '1' for a child profile.

Example 3: Prioritized failover regions in 'Performance' traffic routing

The default behavior for the 'Performance' traffic-routing method is when you have endpoints in different geographic locations the end users are routed to the "closest" endpoint in terms of the lowest network latency.

However, suppose you prefer the West Europe traffic failover to West US, and only direct traffic to other regions when both endpoints are unavailable. You can create this solution using a child profile with the 'Priority' traffic-routing method.

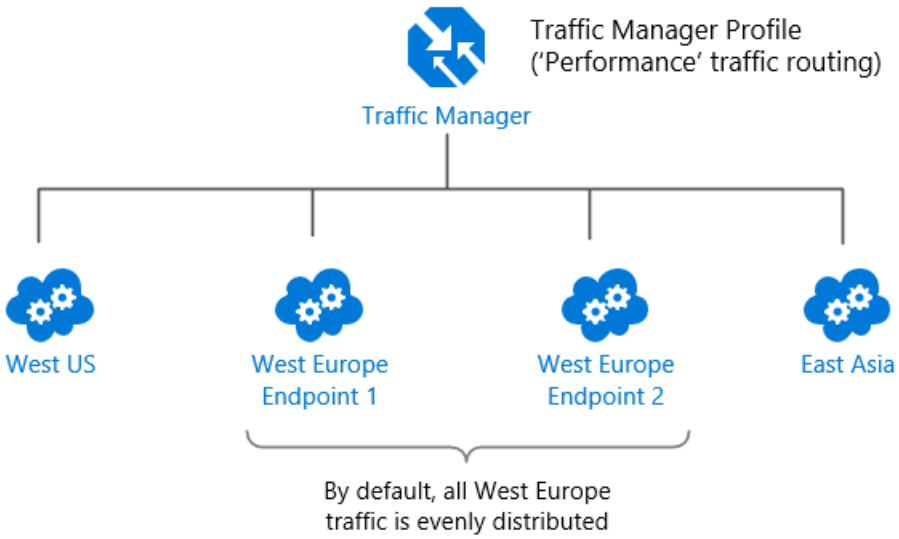


Since the West Europe endpoint has higher priority than the West US endpoint, all traffic is sent to the West Europe endpoint when both endpoints are online. If West Europe fails, its traffic is directed to West US. With the nested profile, traffic is directed to East Asia only when both West Europe and West US fail.

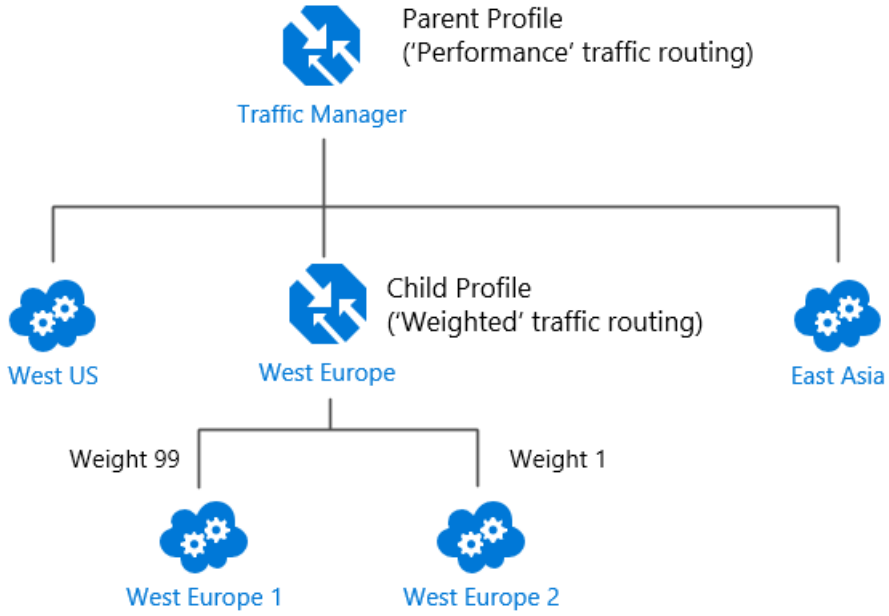
You can repeat this pattern for all regions. Replace all three endpoints in the parent profile with three child profiles, each providing a prioritized failover sequence.

Example 4: Controlling 'Performance' traffic routing between multiple endpoints in the same region

Suppose the 'Performance' traffic-routing method is used in a profile that has more than one endpoint in a particular region. By default, traffic directed to that region is distributed evenly across all available endpoints in that region.

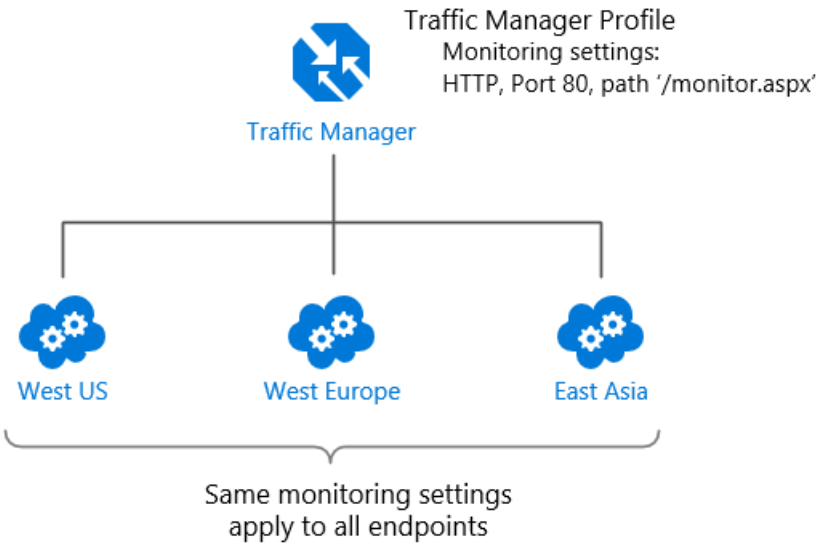


Instead of adding multiple endpoints in West Europe, those endpoints are enclosed in a separate child profile. The child profile is added to the parent as the only endpoint in West Europe. The settings on the child profile can control the traffic distribution with West Europe by enabling priority-based or weighted traffic routing within that region.

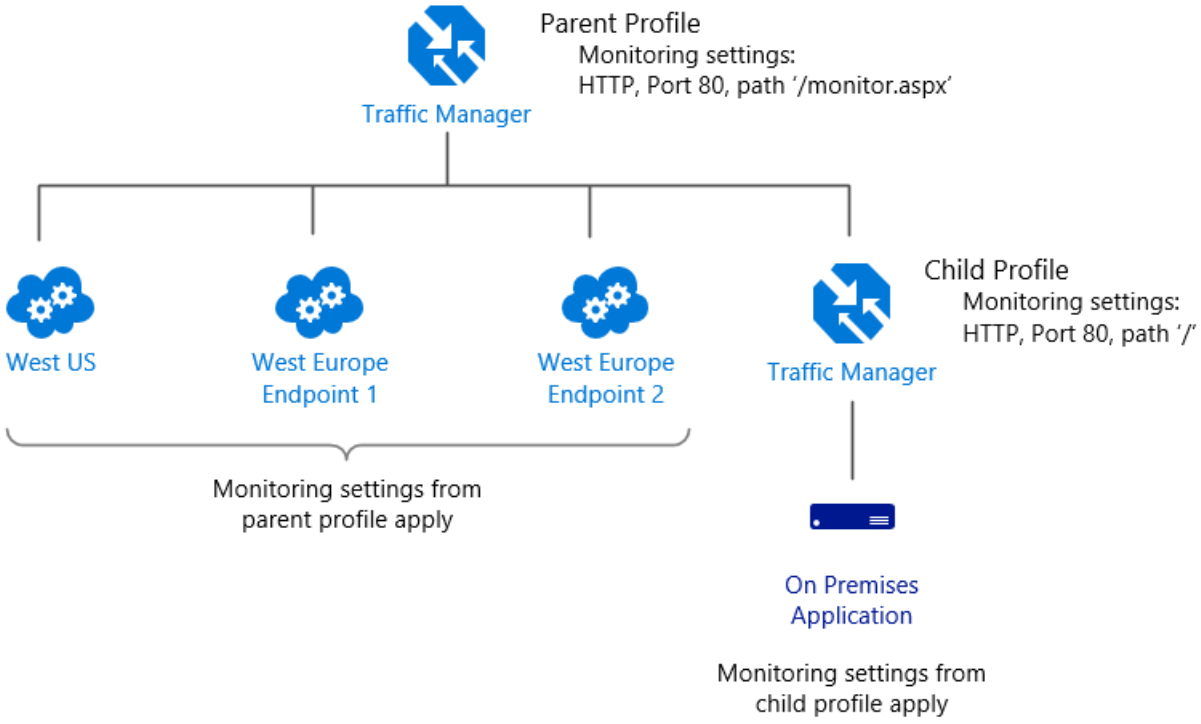


Example 5: Per-endpoint monitoring settings

Suppose you are using Traffic Manager to smoothly migrate traffic from a legacy on-premises web site to a new Cloud-based version hosted in Azure. For the legacy site, you want to use the home page URI to monitor site health. But for the new Cloud-based version, you are implementing a custom monitoring page (path '/monitor.aspx') that includes additional checks.



The monitoring settings in a Traffic Manager profile apply to all endpoints within a single profile. With nested profiles, you use a different child profile per site to define different monitoring settings.



FAQs

- How do I configure nested profiles?
- How many layers of nesting does Traffic Manger support?
- Can I mix other endpoint types with nested child profiles, in the same Traffic Manager profile?
- How does the billing model apply for Nested profiles?
- Is there a performance impact for nested profiles?
- How does Traffic Manager compute the health of a nested endpoint in a parent profile?

Next steps

Learn more about Traffic Manager profiles

Learn how to create a Traffic Manager profile

Traffic Manager endpoints

2/1/2020 • 6 minutes to read • Edit Online

Microsoft Azure Traffic Manager allows you to control how network traffic is distributed to application deployments running in different datacenters. You configure each application deployment as an 'endpoint' in Traffic Manager. When Traffic Manager receives a DNS request, it chooses an available endpoint to return in the DNS response. Traffic manager bases the choice on the current endpoint status and the traffic-routing method. For more information, see How Traffic Manager Works.

There are three types of endpoint supported by Traffic Manager:

- **Azure endpoints** are used for services hosted in Azure.
- **External endpoints** are used for IPv4/IPv6 addresses, FQDNs, or for services hosted outside Azure that can either be on-premises or with a different hosting provider.
- **Nested endpoints** are used to combine Traffic Manager profiles to create more flexible traffic-routing schemes to support the needs of larger, more complex deployments.

There is no restriction on how endpoints of different types are combined in a single Traffic Manager profile. Each profile can contain any mix of endpoint types.

The following sections describe each endpoint type in greater depth.

Azure endpoints

Azure endpoints are used for Azure-based services in Traffic Manager. The following Azure resource types are supported:

- PaaS cloud services.
- Web Apps
- Web App Slots
- PublicIPAddress resources (which can be connected to VMs either directly or via an Azure Load Balancer). The publicIpAddress must have a DNS name assigned to be used in a Traffic Manager profile.

PublicIPAddress resources are Azure Resource Manager resources. They do not exist in the classic deployment model. Thus they are only supported in Traffic Manager's Azure Resource Manager experiences. The other endpoint types are supported via both Resource Manager and the classic deployment model.

When using Azure endpoints, Traffic Manager detects when a Web App is stopped and started. This status is reflected in the endpoint status. See Traffic Manager endpoint monitoring for details. When the underlying service is stopped, Traffic Manager does not perform endpoint health checks or direct traffic to the endpoint. No Traffic Manager billing events occur for the stopped instance. When the service is restarted, billing resumes and the endpoint is eligible to receive traffic. This detection does not apply to PublicIpAddress endpoints.

External endpoints

External endpoints are used for either IPv4/IPv6 addresses, FQDNs, or for services outside of Azure. Use of IPv4/IPv6 address endpoints allows traffic manager to check the health of endpoints without requiring a DNS name for them. As a result, Traffic Manager can respond to queries with A/AAAA records when returning that endpoint in a response. Services outside of Azure can include a service hosted on-premises or with a different provider. External endpoints can be used individually or combined with Azure Endpoints in the same Traffic Manager profile except for endpoints specified as IPv4 or IPv6 addresses which can only be external endpoints.

Combining Azure endpoints with External endpoints enables various scenarios:

- Provide increased redundancy for an existing on-premises application in either an active-active or active-passive failover model using Azure.
- Route traffic to endpoints that do not have a DNS name associated with them. In addition, decrease the overall DNS lookup latency by removing the need to run a second DNS query to get an IP address of a DNS name returned.
- Reduce application latency for users around the world, extend an existing on-premises application to additional geographic locations in Azure. For more information, see Traffic Manager 'Performance' traffic routing.
- Provide additional capacity for an existing on-premises application, either continuously or as a 'burst-to-cloud' solution to meet a spike in demand using Azure.

In certain cases, it is useful to use External endpoints to reference Azure services (for examples, see the FAQ). In this case, health checks are billed at the Azure endpoints rate, not the External endpoints rate. However, unlike Azure endpoints, if you stop or delete the underlying service, health check billing continues until you disable or delete the endpoint in Traffic Manager.

Nested endpoints

Nested endpoints combine multiple Traffic Manager profiles to create flexible traffic-routing schemes and support the needs of larger, complex deployments. With Nested endpoints, a 'child' profile is added as an endpoint to a 'parent' profile. Both the child and parent profiles can contain other endpoints of any type, including other nested profiles. For more information, see nested Traffic Manager profiles.

Web Apps as endpoints

Some additional considerations apply when configuring Web Apps as endpoints in Traffic Manager:

1. Only Web Apps at the 'Standard' SKU or above are eligible for use with Traffic Manager. Attempts to add a Web App of a lower SKU fail. Downgrading the SKU of an existing Web App results in Traffic Manager no longer sending traffic to that Web App. For more information on supported plans see the App Service Plans
2. When an endpoint receives an HTTP request, it uses the 'host' header in the request to determine which Web App should service the request. The host header contains the DNS name used to initiate the request, for example 'contosoapp.azurewebsites.net'. To use a different DNS name with your Web App, the DNS name must be registered as a custom domain name for the App. When adding a Web App endpoint as an Azure endpoint, the Traffic Manager profile DNS name is automatically registered for the App. This registration is automatically removed when the endpoint is deleted.
3. Each Traffic Manager profile can have at most one Web App endpoint from each Azure region. To work around for this constraint, you can configure a Web App as an External endpoint. For more information, see the FAQ.

Enabling and disabling endpoints

Disabling an endpoint in Traffic Manager can be useful to temporarily remove traffic from an endpoint that is in maintenance mode or being redeployed. Once the endpoint is running again, it can be re-enabled.

Endpoints can be enabled and disabled via the Traffic Manager portal, PowerShell, CLI or REST API.

> **NOTE**
> Disabling an Azure endpoint has nothing to do with its deployment state in Azure. An Azure service (such as a VM or Web App remains running and able to receive traffic even when disabled in Traffic Manager. Traffic can be addressed directly to the service instance rather than via the Traffic Manager profile DNS name. For more information, see how Traffic Manager works.

The current eligibility of each endpoint to receive traffic depends on the following factors:

- The profile status (enabled/disabled)
- The endpoint status (enabled/disabled)
- The results of the health checks for that endpoint

For details, see Traffic Manager endpoint monitoring.

> **NOTE**
> Since Traffic Manager works at the DNS level, it is unable to influence existing connections to any endpoint. When an endpoint is unavailable, Traffic Manager directs new connections to another available endpoint. However, the host behind the disabled or unhealthy endpoint may continue to receive traffic via existing connections until those sessions are terminated. Applications should limit the session duration to allow traffic to drain from existing connections.

If all endpoints in a profile are disabled, or if the profile itself is disabled, then Traffic Manager sends an 'NXDOMAIN' response to a new DNS query.

FAQs

- Can I use Traffic Manager with endpoints from multiple subscriptions?
- Can I use Traffic Manager with Cloud Service 'Staging' slots?
- Does Traffic Manager support IPv6 endpoints?
- Can I use Traffic Manager with more than one Web App in the same region?
- How do I move my Traffic Manager profile's Azure endpoints to a different resource group?

Next steps

- Learn how Traffic Manager works.
- Learn about Traffic Manager endpoint monitoring and automatic failover.
- Learn about Traffic Manager traffic routing methods.

Traffic Manager endpoint monitoring

2/1/2020 • 15 minutes to read • Edit Online

Azure Traffic Manager includes built-in endpoint monitoring and automatic endpoint failover. This feature helps you deliver high-availability applications that are resilient to endpoint failure, including Azure region failures.

Configure endpoint monitoring

To configure endpoint monitoring, you must specify the following settings on your Traffic Manager profile:

- **Protocol**. Choose HTTP, HTTPS, or TCP as the protocol that Traffic Manager uses when probing your endpoint to check its health. HTTPS monitoring does not verify whether your SSL certificate is valid--it only checks that the certificate is present.
- **Port**. Choose the port used for the request.
- **Path**. This configuration setting is valid only for the HTTP and HTTPS protocols, for which specifying the path setting is required. Providing this setting for the TCP monitoring protocol results in an error. For HTTP and HTTPS protocol, give the relative path and the name of the webpage or the file that the monitoring accesses. A forward slash (/) is a valid entry for the relative path. This value implies that the file is in the root directory (default).
- **Custom header settings** This configuration setting helps you add specific HTTP headers to the health checks that Traffic Manager sends to endpoints under a profile. The custom headers can be specified at a profile level to be applicable for all endpoints in that profile and / or at an endpoint level applicable only to that endpoint. You can use custom headers for having health checks to endpoints in a multi-tenant environment be routed correctly to their destination by specifying a host header. You can also use this setting by adding unique headers that can be used to identify Traffic Manager originated HTTP(S) requests and processes them differently. You can specify up to eight header:value pairs seprated by a comma. For example, "header1:value1,header2:value2".
- **Expected status code ranges** This setting allows you to specify multiple success code ranges in the format 200-299, 301-301. If these status codes are received as response from an endpoint when a health check is initiated, Traffic Manager marks those endpoints as healthy. You can specify a maximum of 8 status code ranges. This setting is applicable only to HTTP and HTTPS protocol and to all endpoints. This setting is at the Traffic Manager profile level and by default the value 200 is defined as the success status code.
- **Probing interval**. This value specifies how often an endpoint is checked for its health from a Traffic Manager probing agent. You can specify two values here: 30 seconds (normal probing) and 10 seconds (fast probing). If no values are provided, the profile sets to a default value of 30 seconds. Visit the Traffic Manager Pricing page to learn more about fast probing pricing.
- **Tolerated number of failures**. This value specifies how many failures a Traffic Manager probing agent tolerates before marking that endpoint as unhealthy. Its value can range between 0 and 9. A value of 0 means a single monitoring failure can cause that endpoint to be marked as unhealthy. If no value is specified, it uses the default value of 3.
- **Probe timeout**. This property specifies the amount of time the Traffic Manager probing agent should wait before considering that check a failure when a health check probe is sent to the endpoint. If the Probing Interval is set to 30 seconds, then you can set the Timeout value between 5 and 10 seconds. If no value is specified, it uses a default value of 10 seconds. If the Probing Interval is set to 10 seconds, then you can set

the Timeout value between 5 and 9 seconds. If no Timeout value is specified, it uses a default value of 9 seconds.

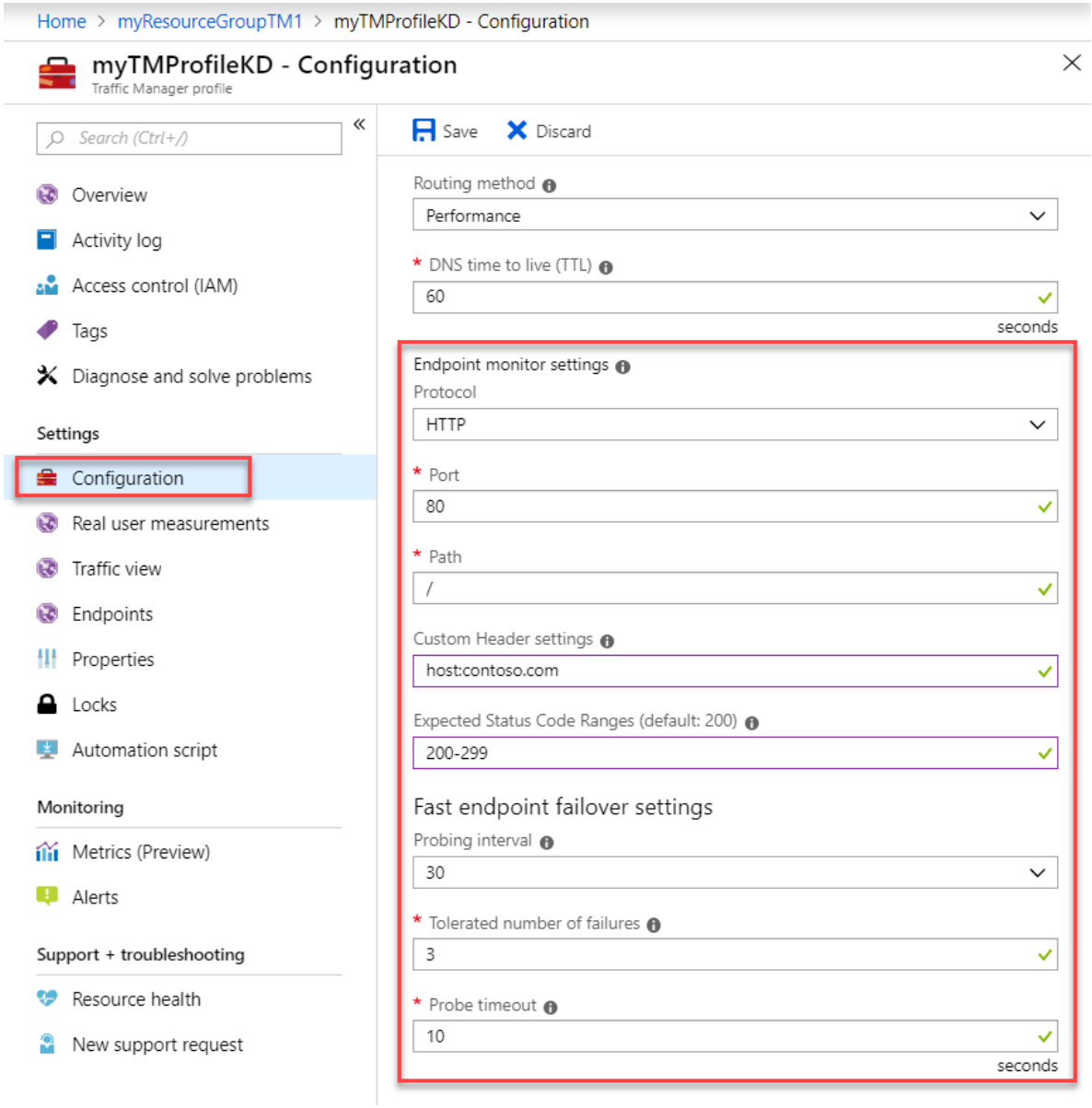


Figure: Traffic Manager endpoint monitoring

How endpoint monitoring works

If the monitoring protocol is set as HTTP or HTTPS, the Traffic Manager probing agent makes a GET request to the endpoint using the protocol, port, and relative path given. If it gets back a 200-OK response, or any of the responses configured in the **Expected status code *ranges**, then that endpoint is considered healthy. If the response is a different value, or, if no response is received within the timeout period specified, then the Traffic Manager probing agent re-attempts according to the Tolerated Number of Failures setting (no re-attempts are done if this setting is 0). If the number of consecutive failures is higher than the Tolerated Number of Failures setting, then that endpoint is marked as unhealthy.

If the monitoring protocol is TCP, the Traffic Manager probing agent initiates a TCP connection request using the port specified. If the endpoint responds to the request with a response to establish the connection, that health check is marked as a success and the Traffic Manager probing agent resets the TCP connection. If the response is a different value, or if no response is received within the timeout period specified, the Traffic Manager probing agent re-attempts according to the Tolerated Number of Failures setting (no re-attempts are made if this setting is 0). If the number of consecutive failures is higher than the Tolerated Number of Failures setting, then that endpoint is marked unhealthy.

In all cases, Traffic Manager probes from multiple locations and the consecutive failure determination happens within each region. This also means that endpoints are receiving health probes from Traffic Manager with a higher frequency than the setting used for Probing Interval.

> **NOTE**
> For HTTP or HTTPS monitoring protocol, a common practice on the endpoint side is to implement a custom page within your application - for example, /health.aspx. Using this path for monitoring, you can perform application-specific checks, such as checking performance counters or verifying database availability. Based on these custom checks, the page returns an appropriate HTTP status code.

All endpoints in a Traffic Manager profile share monitoring settings. If you need to use different monitoring settings for different endpoints, you can create nested Traffic Manager profiles.

Endpoint and profile status

You can enable and disable Traffic Manager profiles and endpoints. However, a change in endpoint status also might occur as a result of Traffic Manager automated settings and processes.

Endpoint status

You can enable or disable a specific endpoint. The underlying service, which might still be healthy, is unaffected. Changing the endpoint status controls the availability of the endpoint in the Traffic Manager profile. When an endpoint status is disabled, Traffic Manager does not check its health and the endpoint is not included in a DNS response.

Profile status

Using the profile status setting, you can enable or disable a specific profile. While endpoint status affects a single endpoint, profile status affects the entire profile, including all endpoints. When you disable a profile, the endpoints are not checked for health and no endpoints are included in a DNS response. An NXDOMAIN response code is returned for the DNS query.

Endpoint monitor status

Endpoint monitor status is a Traffic Manager-generated value that shows the status of the endpoint. You cannot change this setting manually. The endpoint monitor status is a combination of the results of endpoint monitoring and the configured endpoint status. The possible values of endpoint monitor status are shown in the following table:

PROFILE STATUS	ENDPOINT STATUS	ENDPOINT MONITOR STATUS	NOTES
Disabled	Enabled	Inactive	The profile has been disabled. Although the endpoint status is Enabled, the profile status (Disabled) takes precedence. Endpoints in disabled profiles are not monitored. An NXDOMAIN response code is returned for the DNS query.
<any>	Disabled	Disabled	The endpoint has been disabled. Disabled endpoints are not monitored. The endpoint is not included in DNS responses, therefore, it does not receive traffic.

PROFILE STATUS	ENDPOINT STATUS	ENDPOINT MONITOR STATUS	NOTES
Enabled	Enabled	Online	The endpoint is monitored and is healthy. It is included in DNS responses and can receive traffic.
Enabled	Enabled	Degraded	Endpoint monitoring health checks are failing. The endpoint is not included in DNS responses and does not receive traffic. An exception to this is if all endpoints are degraded, in which case all of them are considered to be returned in the query response).
Enabled	Enabled	CheckingEndpoint	The endpoint is monitored, but the results of the first probe have not been received yet. CheckingEndpoint is a temporary state that usually occurs immediately after adding or enabling an endpoint in the profile. An endpoint in this state is included in DNS responses and can receive traffic.
Enabled	Enabled	Stopped	The web app that the endpoint points to is not running. Check the web app settings. This can also happen if the endpoint is of type nested endpoint and the child profile is disabled or is inactive. An endpoint with a Stopped status is not monitored. It is not included in DNS responses and does not receive traffic. An exception to this is if all endpoints are degraded, in which case all of them will be considered to be returned in the query response.

For details about how endpoint monitor status is calculated for nested endpoints, see nested Traffic Manager profiles.

> **NOTE**
> A Stopped Endpoint monitor status can happen on App Service if your web application is not running in the Standard tier or above. For more information, see Traffic Manager integration with App Service.

Profile monitor status

The profile monitor status is a combination of the configured profile status and the endpoint monitor status

values for all endpoints. The possible values are described in the following table:

PROFILE STATUS (AS CONFIGURED)	ENDPOINT MONITOR STATUS	PROFILE MONITOR STATUS	NOTES
Disabled	<any> or a profile with no defined endpoints.	Disabled	The profile has been disabled.
Enabled	The status of at least one endpoint is Degraded.	Degraded	Review the individual endpoint status values to determine which endpoints require further attention.
Enabled	The status of at least one endpoint is Online. No endpoints have a Degraded status.	Online	The service is accepting traffic. No further action is required.
Enabled	The status of at least one endpoint is CheckingEndpoint. No endpoints are in Online or Degraded status.	CheckingEndpoints	This transition state occurs when a profile if created or enabled. The endpoint health is being checked for the first time.
Enabled	The statuses of all endpoints in the profile are either Disabled or Stopped, or the profile has no defined endpoints.	Inactive	No endpoints are active, but the profile is still Enabled.

Endpoint failover and recovery

Traffic Manager periodically checks the health of every endpoint, including unhealthy endpoints. Traffic Manager detects when an endpoint becomes healthy and brings it back into rotation.

An endpoint is unhealthy when any of the following events occur:

- If the monitoring protocol is HTTP or HTTPS:
  - A non-200 response, or a response that does not include the status range specified in the **Expected status code ranges** setting, is received (including a different 2xx code, or a 301/302 redirect).
- If the monitoring protocol is TCP:
  - A response other than ACK or SYN-ACK is received in response to the SYN request sent by Traffic Manager to attempt a connection establishment.
- Timeout.
- Any other connection issue resulting in the endpoint being not reachable.

For more information about troubleshooting failed checks, see Troubleshooting Degraded status on Azure Traffic Manager.

The timeline in the following figure is a detailed description of the monitoring process of Traffic Manager endpoint that has the following settings: monitoring protocol is HTTP, probing interval is 30 seconds, number of tolerated failures is 3, timeout value is 10 seconds, and DNS TTL is 30 seconds.

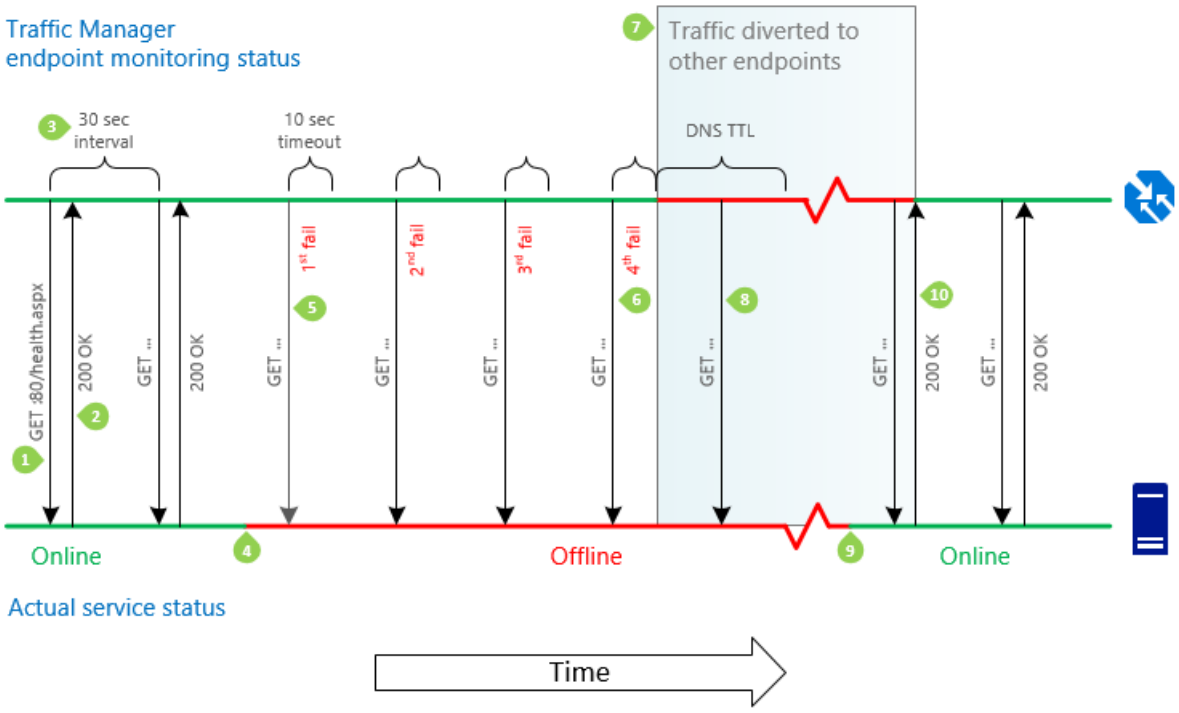


Figure: Traffic manager endpoint failover and recovery sequence

1. **GET**. For each endpoint, the Traffic Manager monitoring system performs a GET request on the path specified in the monitoring settings.
2. **200 OK or custom code range specified Traffic Manager profile monitoring settings** . The monitoring system expects an HTTP 200 OK or the or custom code range specified Traffic Manager profile monitoring settings message to be returned within 10 seconds. When it receives this response, it recognizes that the service is available.
3. **30 seconds between checks**. The endpoint health check is repeated every 30 seconds.
4. **Service unavailable**. The service becomes unavailable. Traffic Manager will not know until the next health check.
5. **Attempts to access the monitoring path**. The monitoring system performs a GET request, but does not receive a response within the timeout period of 10 seconds (alternatively, a non-200 response may be received). It then tries three more times, at 30-second intervals. If one of the tries is successful, then the number of tries is reset.
6. **Status set to Degraded**. After a fourth consecutive failure, the monitoring system marks the unavailable endpoint status as Degraded.
7. **Traffic is diverted to other endpoints**. The Traffic Manager DNS name servers are updated and Traffic Manager no longer returns the endpoint in response to DNS queries. New connections are directed to other, available endpoints. However, previous DNS responses that include this endpoint may still be cached by recursive DNS servers and DNS clients. Clients continue to use the endpoint until the DNS cache expires. As the DNS cache expires, clients make new DNS queries and are directed to different endpoints. The cache duration is controlled by the TTL setting in the Traffic Manager profile, for example, 30 seconds.
8. **Health checks continue**. Traffic Manager continues to check the health of the endpoint while it has a Degraded status. Traffic Manager detects when the endpoint returns to health.
9. **Service comes back online**. The service becomes available. The endpoint retains its Degraded status in

Traffic Manager until the monitoring system performs its next health check.

10. **Traffic to service resumes**. Traffic Manager sends a GET request and receives a 200 OK status response. The service has returned to a healthy state. The Traffic Manager name servers are updated, and they begin to hand out the service's DNS name in DNS responses. Traffic returns to the endpoint as cached DNS responses that return other endpoints expire, and as existing connections to other endpoints are terminated.

> **NOTE**
> Because Traffic Manager works at the DNS level, it cannot influence existing connections to any endpoint. When it directs traffic between endpoints (either by changed profile settings, or during failover or failback), Traffic Manager directs new connections to available endpoints. However, other endpoints might continue to receive traffic via existing connections until those sessions are terminated. To enable traffic to drain from existing connections, applications should limit the session duration used with each endpoint.

Traffic-routing methods

When an endpoint has a Degraded status, it is no longer returned in response to DNS queries. Instead, an alternative endpoint is chosen and returned. The traffic-routing method configured in the profile determines how the alternative endpoint is chosen.

- **Priority**. Endpoints form a prioritized list. The first available endpoint on the list is always returned. If an endpoint status is Degraded, then the next available endpoint is returned.
- **Weighted**. Any available endpoint is chosen at random based on their assigned weights and the weights of the other available endpoints.
- **Performance**. The endpoint closest to the end user is returned. If that endpoint is unavailable, Traffic Manager moves traffic to the endpoints in the next closest Azure region. You can configure alternative failover plans for performance traffic-routing by using nested Traffic Manager profiles.
- **Geographic**. The endpoint mapped to serve the geographic location based on the query request IP's is returned. If that endpoint is unavailable, another endpoint will not be selected to failover to, since a geographic location can be mapped only to one endpoint in a profile (more details are in the FAQ). As a best practice, when using geographic routing, we recommend customers to use nested Traffic Manager profiles with more than one endpoint as the endpoints of the profile.
- **MultiValue** Multiple endpoints mapped to IPv4/IPv6 addresses are returned. When a query is received for this profile, healthy endpoints are returned based on the **Maximum record count in response** value that you have specified. The default number of responses is two endpoints.
- **Subnet** The endpoint mapped to a set of IP address ranges is returned. When a request is received from that IP address, the endpoint returned is the one mapped for that IP address.

For more information, see Traffic Manager traffic-routing methods.

NOTE

One exception to normal traffic-routing behavior occurs when all eligible endpoints have a degraded status. Traffic Manager makes a "best effort" attempt and *responds as if all the Degraded status endpoints actually are in an online state*. This behavior is preferable to the alternative, which would be to not return any endpoint in the DNS response. Disabled or Stopped endpoints are not monitored, therefore, they are not considered eligible for traffic.

This condition is commonly caused by improper configuration of the service, such as:

- An access control list [ACL] blocking the Traffic Manager health checks.
- An improper configuration of the monitoring port or protocol in the Traffic manager profile.

The consequence of this behavior is that if Traffic Manager health checks are not configured correctly, it might appear from the traffic routing as though Traffic Manager *is* working properly. However, in this case, endpoint failover cannot happen which affects overall application availability. It is important to check that the profile shows an Online status, not a Degraded status. An Online status indicates that the Traffic Manager health checks are working as expected.

For more information about troubleshooting failed health checks, see Troubleshooting Degraded status on Azure Traffic Manager.

FAQs

- Is Traffic Manager resilient to Azure region failures?
- How does the choice of resource group location affect Traffic Manager?
- How do I determine the current health of each endpoint?
- Can I monitor HTTPS endpoints?
- Do I use an IP address or a DNS name when adding an endpoint?
- What types of IP addresses can I use when adding an endpoint?
- Can I use different endpoint addressing types within a single profile?
- What happens when an incoming query's record type is different from the record type associated with the addressing type of the endpoints?
- Can I use a profile with IPv4 / IPv6 addressed endpoints in a nested profile?
- I stopped an web application endpoint in my Traffic Manager profile but I am not receiving any traffic even after I restarted it. How can I fix this?
- Can I use Traffic Manager even if my application does not have support for HTTP or HTTPS?
- What specific responses are required from the endpoint when using TCP monitoring?
- How fast does Traffic Manager move my users away from an unhealthy endpoint?
- How can I specify different monitoring settings for different endpoints in a profile?
- How can I assign HTTP headers to the Traffic Manager health checks to my endpoints?
- What host header do endpoint health checks use?
- What are the IP addresses from which the health checks originate?
- How many health checks to my endpoint can I expect from Traffic Manager?
- How can I get notified if one of my endpoints goes down?

Next steps

Learn how Traffic Manager works

Learn more about the traffic-routing methods supported by Traffic Manager

Learn how to create a Traffic Manager profile

Troubleshoot Degraded status on a Traffic Manager endpoint

Traffic Manager Real User Measurements overview

2/1/2020 • 3 minutes to read • Edit Online

When you set up a Traffic Manager profile to use the performance routing method, the service looks at where the DNS query requests are coming from and makes routing decisions to direct those requestors to the Azure region that gives them the lowest latency. This is accomplished by utilizing the network latency intelligence that Traffic Manager maintains for different end-user networks.

Real User Measurements enables you to measure network latency measurements to Azure regions, from the client applications your end users use, and have Traffic Manager consider that information as well when making routing decisions. By choosing to use the Real User Measurements, you can increase the accuracy of the routing for requests coming from those networks where your end users reside.

How Real User Measurements work

Real User Measurements work by having your client applications measure latency to Azure regions as it is seen from the end-user networks in which they are used. For example, if you have a web page that is accessed by users across different locations (for example, in the North American regions), you can use Real User Measurements with performance routing method to get them to the best Azure region in which your server application is hosted.

It starts by embedding an Azure provided JavaScript (with a unique key in it) in your web pages. Once that is done, whenever a user visits that webpage, the JavaScript queries Traffic Manager to get information about the Azure regions it should measure. The service returns a set of endpoints to the script that then measure these regions consecutively by downloading a single pixel image hosted in those Azure regions and noting the latency between the time the request was sent and the time when the first byte was received. These measurements are then reported back to the Traffic Manager service.

Over time, this happens many times and across many networks leading to Traffic Manager getting more accurate information about the latency characteristics of the networks in which your end users reside. This information starts getting to be included in the routing decisions made by Traffic Manager. As a result, it leads to increased accuracy in those decisions based on the Real User Measurements sent.

When you use Real User Measurements, you are billed based on the number of measurements sent to Traffic Manager. For more details on the pricing, visit the Traffic Manager pricing page.

FAQs

- What are the benefits of using Real User Measurements?
- Can I use Real User Measurements with non-Azure regions?
- Which routing method benefits from Real User Measurements?
- Do I need to enable Real User Measurements each profile separately?
- How do I turn off Real User Measurements for my subscription?
- Can I use Real User Measurements with client applications other than web pages?
- How many measurements are made each time my Real User Measurements enabled web page is rendered?
- Is there a delay before Real User Measurements script runs in my webpage?
- Can I use Real User Measurements with only the Azure regions I want to measure?

- Can I limit the number of measurements made to a specific number?
- Can I see the measurements taken by my client application as part of Real User Measurements?
- Can I modify the measurement script provided by Traffic Manager?
- Will it be possible for others to see the key I use with Real User Measurements?
- Can others abuse my RUM key?
- Do I need to put the measurement JavaScript in all my web pages?
- Can information about my end users be identified by Traffic Manager if I use Real User Measurements?
- Does the webpage measuring Real User Measurements need to be using Traffic Manager for routing?
- Do I need to host any service on Azure regions to use with Real User Measurements?
- Will my Azure bandwidth usage increase when I use Real User Measurements?

Next steps

- Learn how to use Real User Measurements with web pages
- Learn how Traffic Manager works
- Learn more about Mobile Center
- Learn more about the traffic-routing methods supported by Traffic Manager
- Learn how to create a Traffic Manager profile

Traffic Manager Traffic View

2/1/2020 • 5 minutes to read • Edit Online

Traffic Manager provides you with DNS level routing so that your end users are directed to healthy endpoints based on the routing method specified when you created the profile. Traffic View provides Traffic Manager with a view of your user bases (at a DNS resolver granularity level) and their traffic pattern. When you enable Traffic View, this information is processed to provide you with actionable insights.

By using Traffic View, you can:

- understand where your user bases are located (up to a local DNS resolver level granularity).
- view the volume of traffic (observed as DNS queries handled by Azure Traffic Manager) originating from these regions.
- get insights into what is the representative latency experienced by these users.
- deep dive into the specific traffic patterns from each of these user bases to Azure regions where you have endpoints.

For example, you can use Traffic View to understand which regions have a large number of traffic but suffer from higher latencies. Next, you can use this information to plan your footprint expansion to new Azure regions so that these users can have a lower latency experience.

How Traffic View works

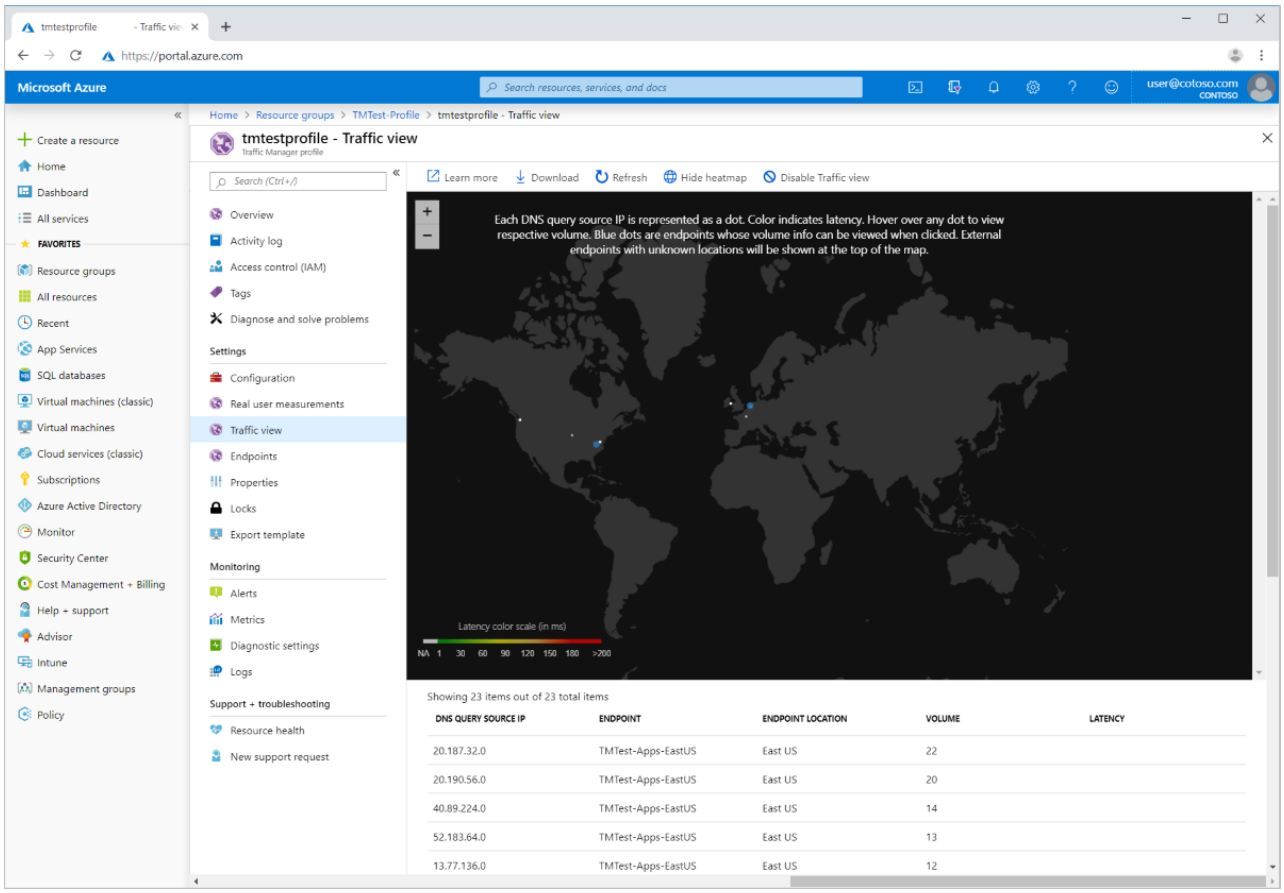
Traffic View works by having Traffic Manager look at the incoming queries received in the past seven days against a profile that has this feature enabled. From the incoming queries information, Traffic View extracts the source IP of the DNS resolver that is used as a representation of the location of the users. These are then grouped together at a DNS resolver level granularity to create user base regions by using the geographic information of IP addresses maintained by Traffic Manager. Traffic Manager then looks at the Azure regions to which the query was routed and constructs a traffic flow map for users from those regions.
In the next step, Traffic Manager correlates the user base region to Azure region mapping with the network intelligence latency tables that it maintains for different end-user networks to understand the average latency experienced by users from those regions when connecting to Azure regions. All these calculations are then combined at a per local DNS resolver IP level before it is presented to you. You can consume the information in various ways.

The frequency of Traffic view data update depends on multiple internal service variables. However, the data is usually updated once every 24 hours.

> **NOTE**
> The latency described in Traffic View is a representative latency between the end user and the Azure regions to which they had connected to, and is not the DNS lookup latency. Traffic View makes a best effort estimate of the latency between the local DNS resolver and the Azure region the query was routed to, if there is insufficient data available then the latency returned will be null.

Visual overview

When you navigate to the **Traffic View** section in your Traffic Manager page, you are presented with a geographical map with an overlay of Traffic View insights. The map provides information about the user base and endpoints for your Traffic Manager profile.



User base information

For those local DNS resolvers for which location information is available, they are shown in the map. The color of the DNS resolver denotes the average latency experienced by end users who used that DNS resolver for their Traffic Manager queries.

If you hover over a DNS resolver location in the map, it shows:

- the IP address of the DNS resolver
- the volume of DNS query traffic seen by Traffic Manager from it
- the endpoints to which traffic from the DNS resolver was routed, as a line between the endpoint and the DNS resolver
- the average latency from that location to the endpoint, represented as the color of the line connecting them

Endpoint information

The Azure regions in which the endpoints reside are shown as blue dots in the map. If your endpoint is external and doesn't have an Azure region mapped to it, it is shown at the top of the map. Click on any endpoint to see the different locations (based on the DNS resolver used) from where traffic was directed to that endpoint. The connections are shown as a line between the endpoint and the DNS resolver location and are colored according to the representative latency between that pair. In addition, you can see the name of the endpoint, the Azure region in which it runs, and the total volume of requests that were directed to it by this Traffic Manager profile.

Tabular listing and raw data download

You can view the Traffic View data in a tabular format in Azure portal. There is an entry for each DNS resolver IP / endpoint pair that shows the IP address of the DNS resolver, the name and geographical location of the Azure region in which the endpoint is located (if available), the volume of requests associated with that DNS resolver to that endpoint, and the representative latency associated with end users using that DNS (where available). You can also download the Traffic View data as a CSV file that can be used as a part of an analytics workflow of your choice.

Billing

When you use Traffic View, you are billed based on the number of data points used to create the insights presented. Currently, the only data point type used is the queries received against your Traffic Manager profile. For more details on the pricing, visit the Traffic Manager pricing page.

FAQs

- What does Traffic View do?
- How can I benefit from using Traffic View?
- How is Traffic View different from the Traffic Manager metrics available through Azure monitor?
- Does Traffic View use EDNS Client Subnet information?
- How many days of data does Traffic View use?
- How does Traffic View handle external endpoints?
- Do I need to enable Traffic View for each profile in my subscription?
- How can I turn off Traffic View?
- How does Traffic View billing work?

Next steps

- Learn how Traffic Manager works
- Learn more about the traffic-routing methods supported by Traffic Manager
- Learn how to create a Traffic Manager profile

Traffic Manager metrics and alerts

2/1/2020 • 2 minutes to read • Edit Online

Traffic Manager provides you with DNS-based load balancing that includes multiple routing methods and endpoint monitoring options. This article describes the metrics and associated alerts that are available to customers.

Metrics available in Traffic Manager

Traffic Manager provides the following metrics on a per profile basis that customers can use to understand their usage of Traffic manager and the status of their endpoints under that profile.

Queries by endpoint returned

Use this metric to view the number of queries that a Traffic Manager profile processes over a specified period. You can also view the same information at an endpoint level granularity that helps you understand how many times an endpoint was returned in the query responses from Traffic Manager.

In the following example, Figure 1 displays all the query responses that the Traffic Manager profile returns.

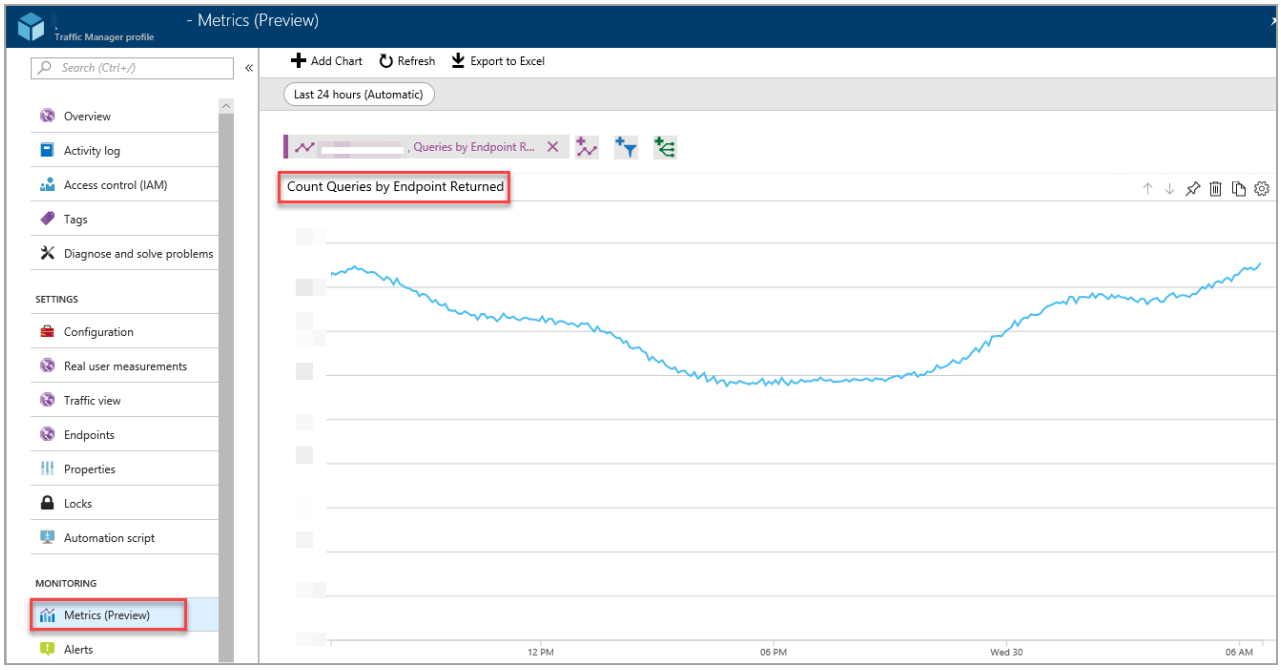


Figure 1: Aggregate view with all queries

Figure 2 displays the same information, however, it is split by endpoints. As a result, you can see the volume of query responses in which a specific endpoint was returned.

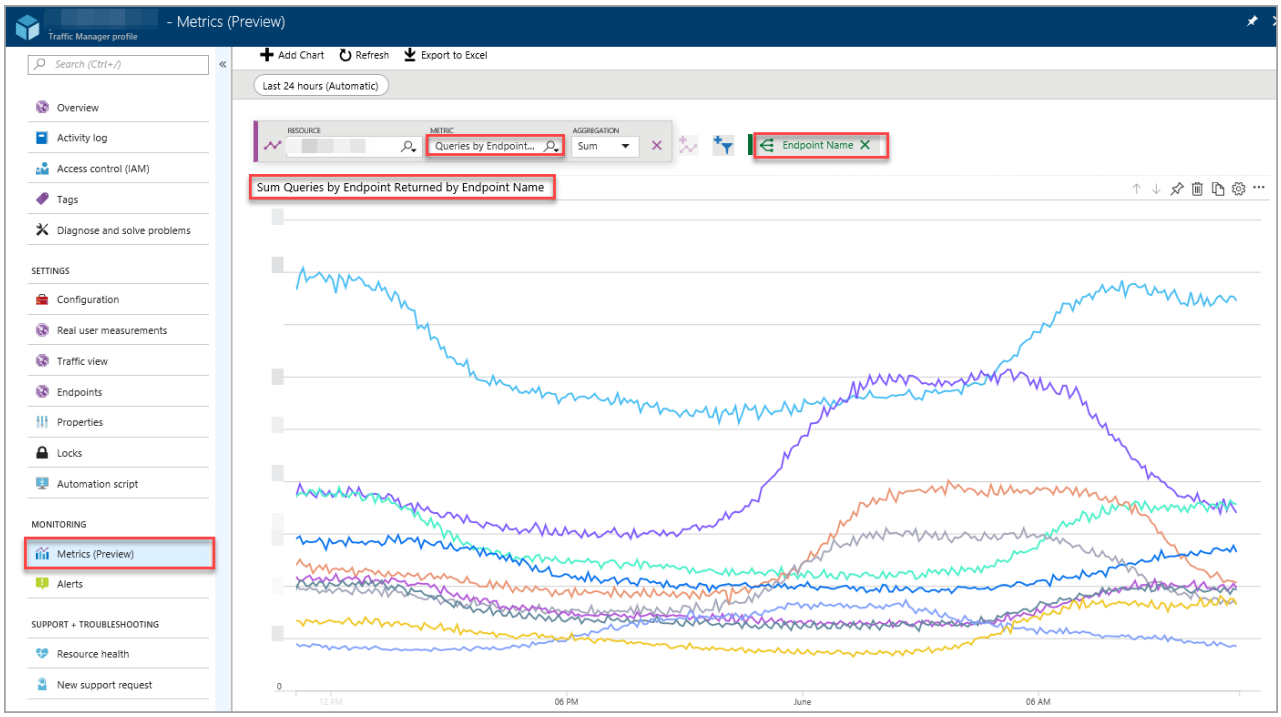


Figure 2: Split view with query volume shown per endpoint returned

Endpoint status by endpoint

Use this metric to understand the health status of the endpoints in the profile. It takes two values:

- use **1** if the endpoint is up.
- use **0** if the endpoint is down.

This metric can be shown either as an aggregate value representing the status of all the metrics (Figure 3), or, it can be split (see Figure 4) to show the status of specific endpoints. If the former, if the aggregation level is selected as **Avg**, the value of this metric is the arithmetic average of the status of all endpoints. For example, if a profile has two endpoints and only one is healthy, then this metric has a value of **0.50** as shown in Figure 3.

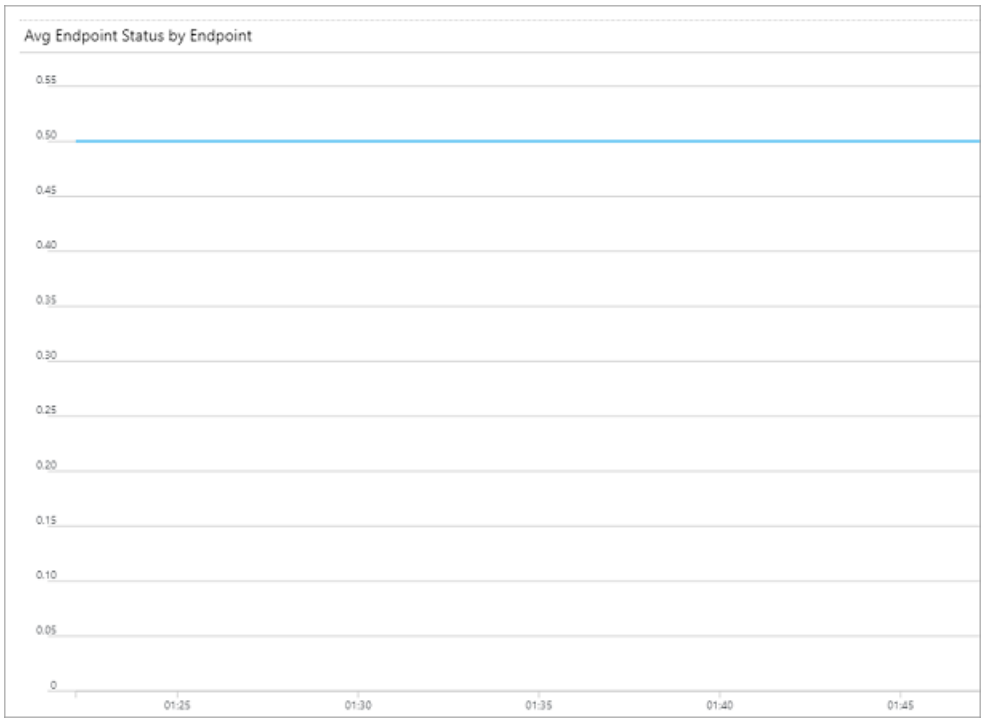


Figure 3: Composite view of endpoint status metric – "Avg" aggregation selected

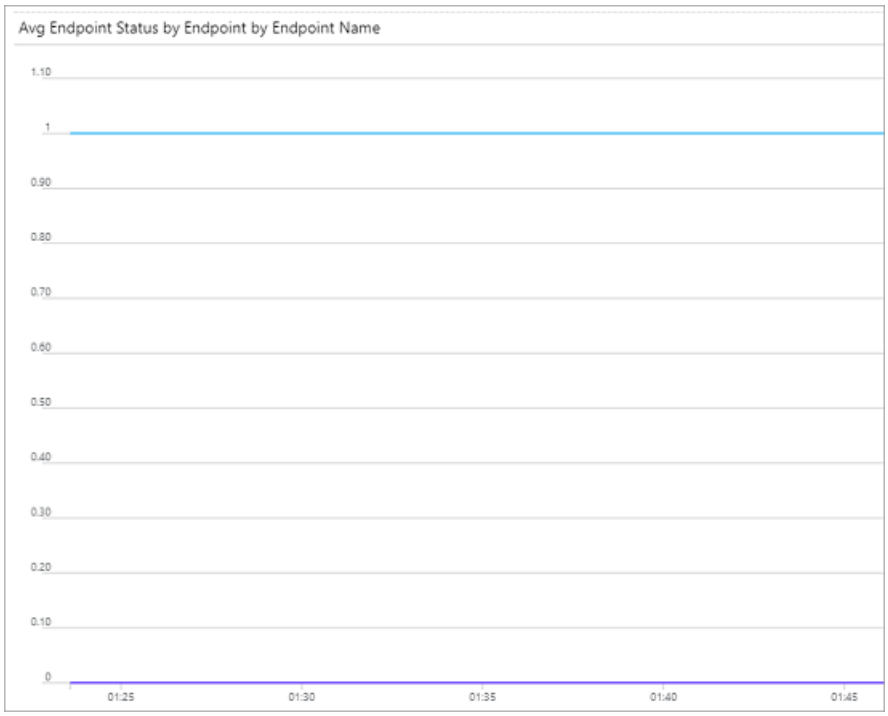


Figure 4: Split view of endpoint status metrics

You can consume these metrics through Azure Monitor service's portal, REST API, Azure CLI, and Azure PowerShell, or through the metrics section of Traffic Manager's portal experience.

Alerts on Traffic Manager metrics

In addition to processing and displaying metrics from Traffic Manager, Azure Monitor enables customers to configure and receive alerts associated with these metrics. You can choose what conditions need to be met in these metrics for an alert to occur, how often those conditions need to be monitored, and how the alerts should be sent to you. For more information, see Azure Monitor alerts documentation.

Next steps

- Learn more about Azure Monitor service
- Learn how to create a chart using Azure Monitor

Disaster recovery using Azure DNS and Traffic Manager

11/25/2019 • 10 minutes to read • Edit Online

Disaster recovery focuses on recovering from a severe loss of application functionality. In order to choose a disaster recovery solution, business and technology owners must first determine the level of functionality that is required during a disaster, such as - unavailable, partially available via reduced functionality, or delayed availability, or fully available. Most enterprise customers are choosing a multi-region architecture for resiliency against an application or infrastructure level failover. Customers can choose several approaches in the quest to achieve failover and high availability via redundant architecture. Here are some of the popular approaches:

- **Active-passive with cold standby**: In this failover solution, the VMs and other appliances that are running in the standby region are not active until there is a need for failover. However, the production environment is replicated in the form of backups, VM images, or Resource Manager templates, to a different region. This failover mechanism is cost-effective but takes a longer time to undertake a complete failover.

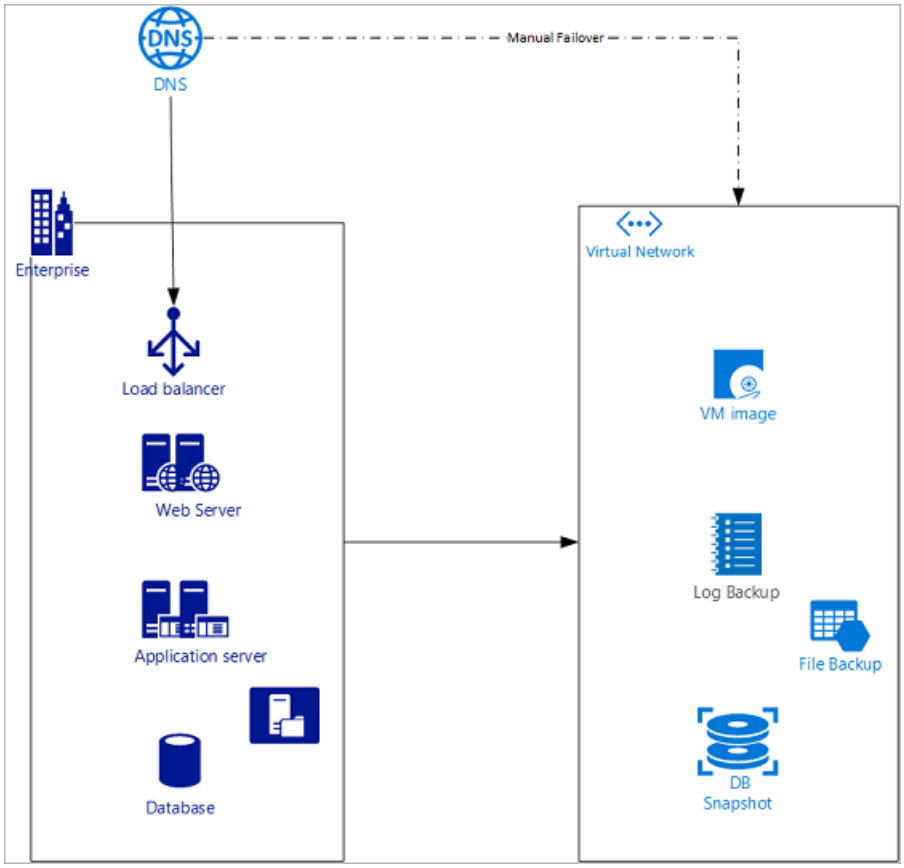


Figure - Active/Passive with cold standby disaster recovery configuration

- **Active/Passive with pilot light**: In this failover solution, the standby environment is set up with a minimal configuration. The setup has only the necessary services running to support only a minimal and critical set of applications. In its native form, this scenario can only execute minimal functionality but can scale up and spawn additional services to take bulk of the production load if a failover occurs.

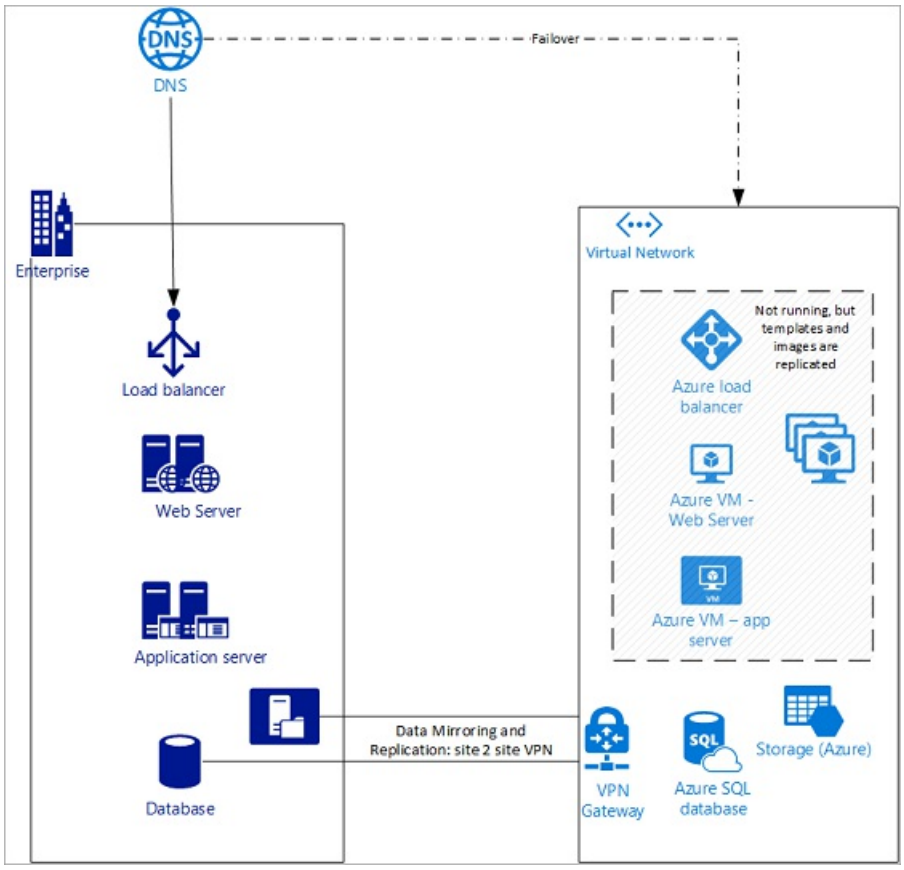


Figure: Active/Passive with pilot light disaster recovery configuration

- **Active/Passive with warm standby**: In this failover solution, the standby region is pre-warmed and is ready to take the base load, auto scaling is turned on, and all the instances are up and running. This solution is not scaled to take the full production load but is functional, and all services are up and running. This solution is an augmented version of the pilot light approach.

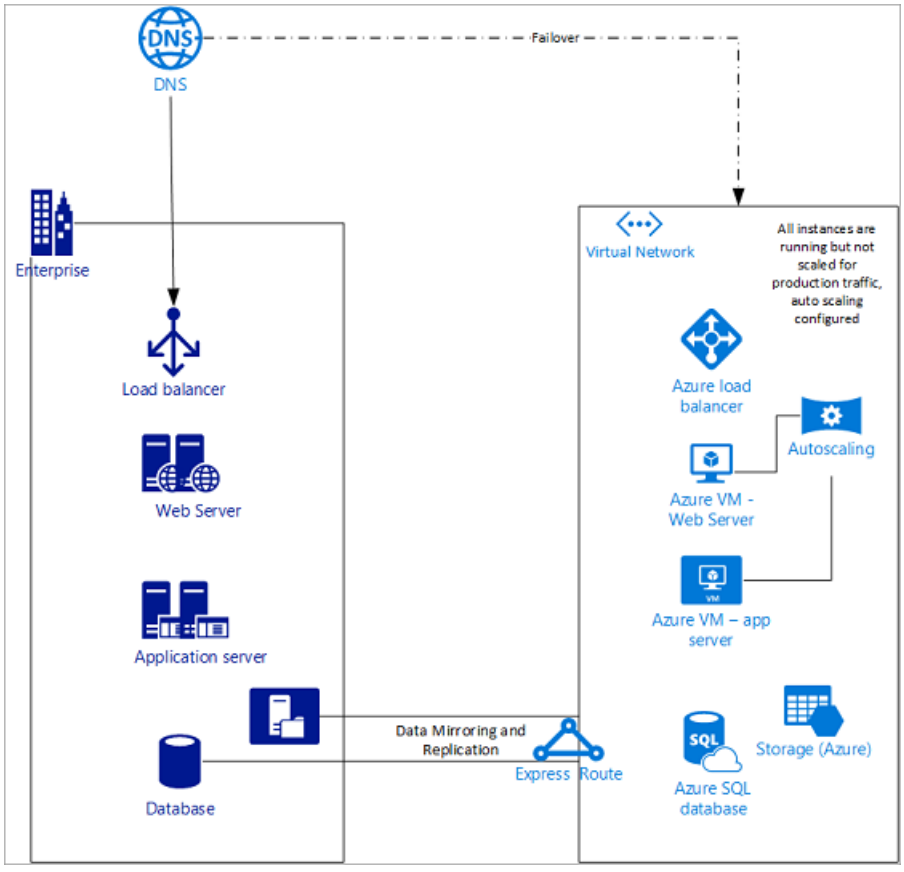


Figure: Active/Passive with warm standby disaster recovery configuration

To learn more about failover and high availability, see Disaster Recovery for Azure Applications.

Planning your disaster recovery architecture

There are two technical aspects towards setting up your disaster recovery architecture:

- Using a deployment mechanism to replicate instances, data, and configurations between primary and standby environments. This type of disaster recovery can be done natively via Azure Site-Recovery via Microsoft Azure partner appliances/services like Veritas or NetApp.
- Developing a solution to divert network/web traffic from the primary site to the standby site. This type of disaster recovery can be achieved via Azure DNS, Azure Traffic Manager(DNS), or third-party global load balancers.

This article is limited to approaches via Network and Web traffic redirection. For instructions to set up Azure Site Recovery, see Azure Site Recovery Documentation. DNS is one of the most efficient mechanisms to divert network traffic because DNS is often global and external to the data center and is insulated from any regional or availability zone (AZ) level failures. One can use a DNS-based failover mechanism and in Azure, two DNS services can accomplish the same in some fashion - Azure DNS (authoritative DNS) and Azure Traffic Manager (DNS-based smart traffic routing).

It is important to understand few concepts in DNS that are extensively used to discuss the solutions provided in this article:

- **DNS A Record** – A Records are pointers that point a domain to an IPv4 address.
- **CNAME or Canonical name** - This record type is used to point to another DNS record. CNAME doesn't respond with an IP address but rather the pointer to the record that contains the IP address.
- **Weighted Routing** – one can choose to associate a weight to service endpoints and then distribute the traffic based on the assigned weights. This routing method is one of the four traffic routing mechanisms available within Traffic Manager. For more information, see Weighted routing method.
- **Priority Routing** – Priority routing is based on health checks of endpoints. By default, Azure Traffic manager sends all traffic to the highest priority endpoint, and upon a failure or disaster, Traffic Manager routes the traffic to the secondary endpoint. For more information, see Priority routing method.

Manual failover using Azure DNS

The Azure DNS manual failover solution for disaster recovery uses the standard DNS mechanism to failover to the backup site. The manual option via Azure DNS works best when used in conjunction with the cold standby or the pilot light approach.

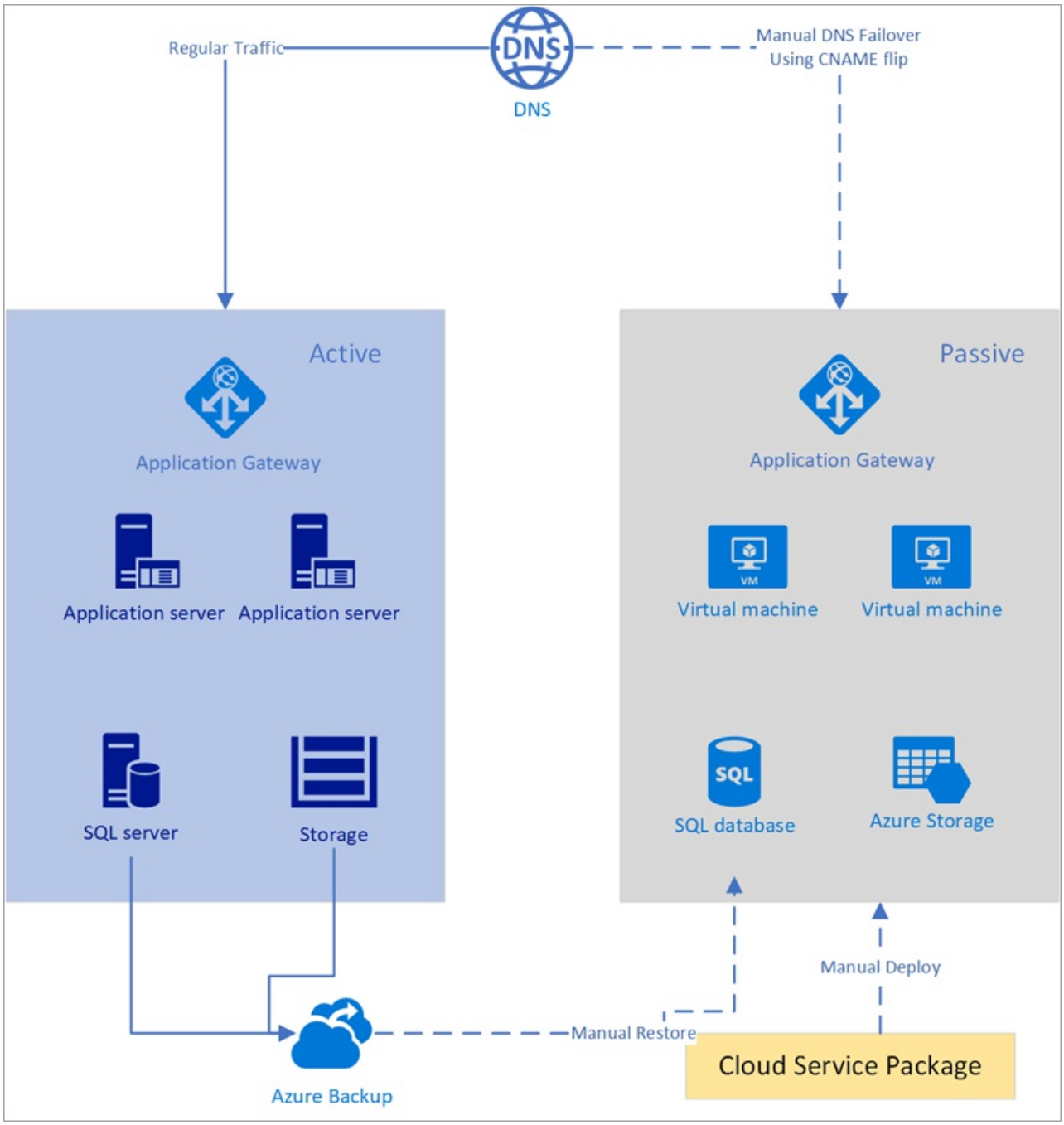


Figure - Manual failover using Azure DNS

The assumptions made for the solution are:

- Both primary and secondary endpoints have static IPs that don't change often. Say for the primary site the IP is 100.168.124.44 and the IP for the secondary site is 100.168.124.43.
- An Azure DNS zone exists for both the primary and secondary site. Say for the primary site the endpoint is prod.contoso.com and for the backup site is dr.contoso.com. A DNS record for the main application known as www.contoso.com also exists.
- The TTL is at or below the RTO SLA set in the organization. For example, if an enterprise sets the RTO of the application disaster response to be 60 mins, then the TTL should be less than 60 mins, preferably the lower the better. You can set up Azure DNS for manual failover as follows:
- Create a DNS zone
- Create DNS zone records
- Update CNAME record

Step 1: Create a DNS

Create a DNS zone (for example, www.contoso.com) as shown below:

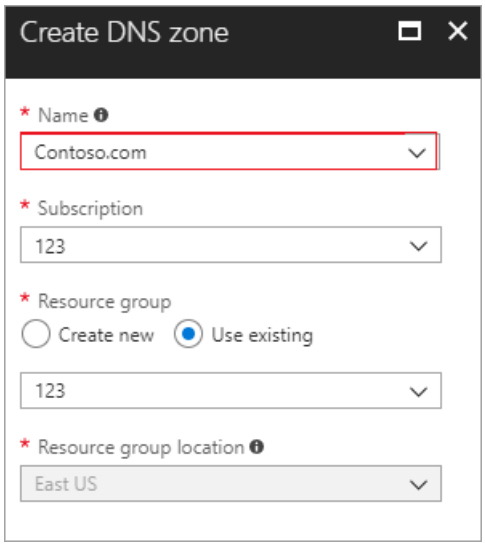


Figure - Create a DNS zone in Azure

Step 2: Create DNS zone records

Within this zone create three records (for example - www.contoso.com, prod.contoso.com and dr.consoto.com) as show below.

NAME	TYPE	TTL	VALUE
@	NS	172800	ns1-05.azure-dns.com. ns2-05.azure-dns.net. ns3-05.azure-dns.org. ns4-05.azure-dns.info.
@	SOA	3600	Email: azuredns-hostmaster.microsoft.com Host: ns1-05.azure-dns.com. Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 300 Serial number: 1
dr	A	300	100.168.124.43
prod	A	300	100.168.124.44
www	CNAME	1800	prod.contoso.com

Figure - Create DNS zone records in Azure

In this scenario, site, www.contoso.com has a TTL of 30 mins, which is well below the stated RTO, and is pointing to the production site prod.contoso.com. This configuration is during normal business operations. The TTL of prod.contoso.com and dr.contoso.com has been set to 300 seconds or 5 mins. You can use an Azure monitoring service such as Azure Monitor or Azure App Insights, or, any partner monitoring solutions such as Dynatrace, You can even use home grown solutions that can monitor or detect application or virtual infrastructure level failures.

Step 3: Update the CNAME record

Once failure is detected, change the record value to point to dr.contoso.com as shown below:

Figure - Update the CNAME record in Azure

Within 30 minutes, during which most resolvers will refresh the cached zone file, any query to www.contoso.com

will be redirected to dr.contoso.com. You can also run the following Azure CLI command to change the CNAME value:

```
az network dns record-set cname set-record \
--resource-group 123 \
--zone-name contoso.com \
--record-set-name www \
--cname dr.contoso.com
```

This step can be executed manually or via automation. It can be done manually via the console or by the Azure CLI. The Azure SDK and API can be used to automate the CNAME update so that no manual intervention is required. Automation can be built via Azure functions or within a third-party monitoring application or even from on-premises.

How manual failover works using Azure DNS

Since the DNS server is outside the failover or disaster zone, it is insulated against any downtime. This enables user to architect a simple failover scenario that is cost effective and will work all the time assuming that the operator has network connectivity during disaster and can make the flip. If the solution is scripted, then one must ensure that the server or service running the script should be insulated against the problem affecting the production environment. Also, keep in mind the low TTL that was set against the zone so that no resolver around the world keeps the endpoint cached for long and customers can access the site within the RTO. For a cold standby and pilot light, since some prewarming and other administrative activity may be required – one should also give enough time before making the flip.

Automatic failover using Azure Traffic Manager

When you have complex architectures and multiple sets of resources capable of performing the same function, you can configure Azure Traffic Manager (based on DNS) to check the health of your resources and route the traffic from the non-healthy resource to the healthy resource. In the following example, both the primary region and the secondary region have a full deployment. This deployment includes the cloud services and a synchronized database.

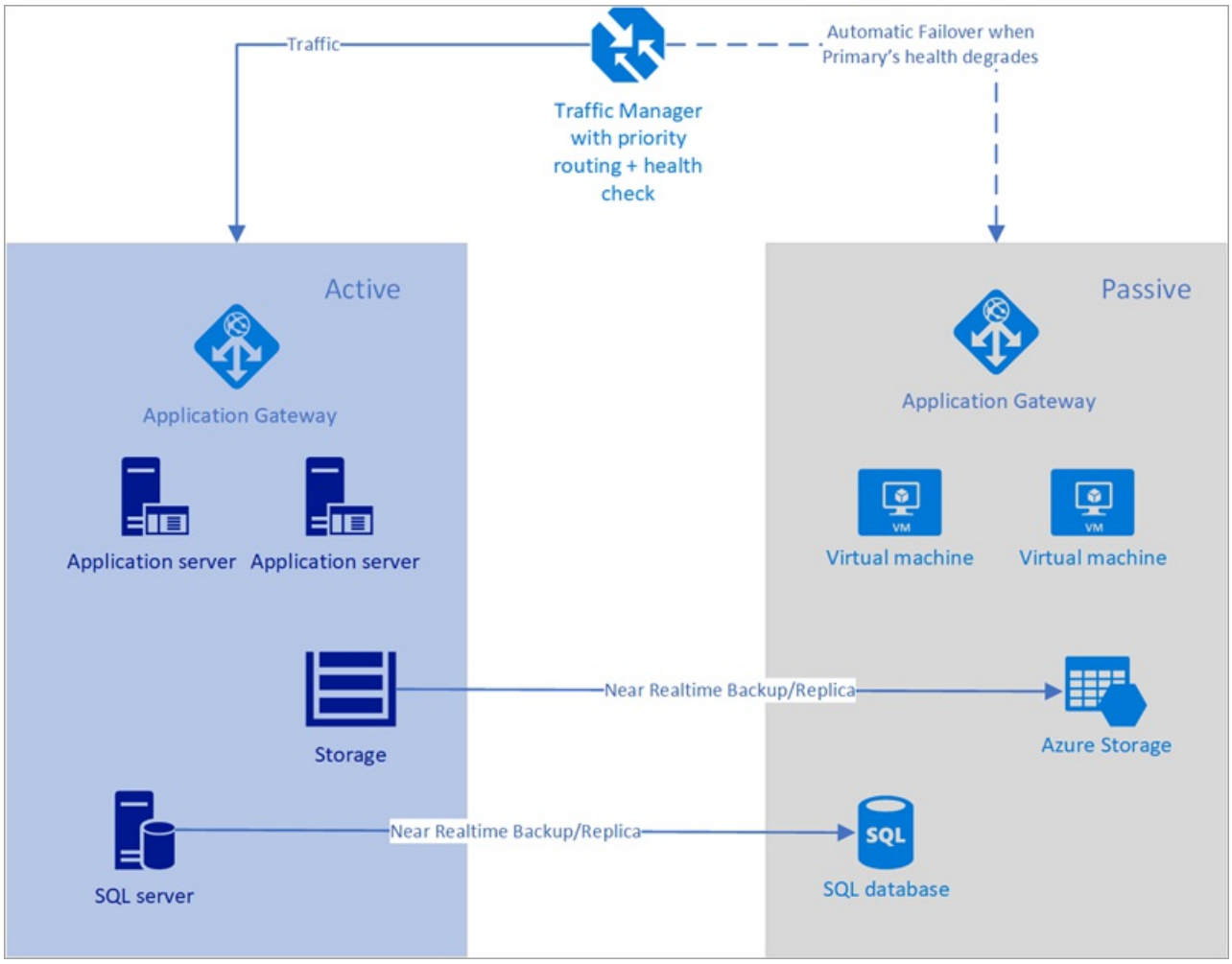


Figure - Automatic failover using Azure Traffic Manager

However, only the primary region is actively handling network requests from the users. The secondary region becomes active only when the primary region experiences a service disruption. In that case, all new network requests route to the secondary region. Since the backup of the database is near instantaneous, both the load balancers have IPs that can be health checked, and the instances are always up and running, this topology provides an option for going in for a low RTO and failover without any manual intervention. The secondary failover region must be ready to go-live immediately after failure of the primary region. This scenario is ideal for the use of Azure Traffic Manager that has inbuilt probes for various types of health checks including http / https and TCP. Azure Traffic manager also has a rule engine that can be configured to failover when a failure occurs as described below. Let's consider the following solution using Traffic Manager:

- Customer has the Region #1 endpoint known as prod.contoso.com with a static IP as 100.168.124.44 and a Region #2 endpoint known as dr.contoso.com with a static IP as 100.168.124.43.
- Each of these environments is fronted via a public facing property like a load balancer. The load balancer can be configured to have a DNS-based endpoint or a fully qualified domain name (FQDN) as shown above.
- All the instances in Region 2 are in near real-time replication with Region 1. Furthermore, the machine images are up-to-date, and all software/configuration data is patched and are in line with Region 1.
- Autoscaling is preconfigured in advance.

The steps taken to configure the failover with Azure Traffic Manager are as follows:

1. Create a new Azure Traffic Manager profile
2. Create endpoints within the Traffic Manager profile
3. Set up health check and failover configuration

Step 1: Create a new Azure Traffic Manager profile

Create a new Azure Traffic manager profile with the name contoso123 and select the Routing method as Priority. If

you have a pre-existing resource group that you want to associate with, then you can select an existing resource group, otherwise, create a new resource group.

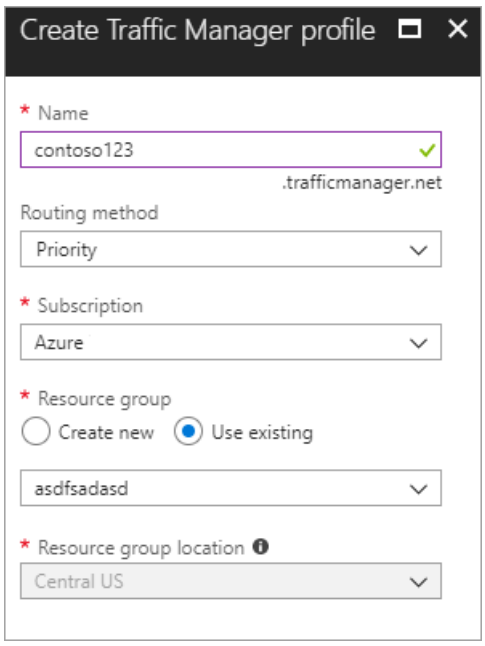


Figure - Create a Traffic Manager profile

Step 2: Create endpoints within the Traffic Manager profile

In this step, you create endpoints that point to the production and disaster recovery sites. Here, choose the **Type** as an external endpoint, but if the resource is hosted in Azure, then you can choose **Azure endpoint** as well. If you choose **Azure endpoint**, then select a **Target resource** that is either an **App Service** or a **Public IP** that is allocated by Azure. The priority is set as **1** since it is the primary service for Region 1. Similarly, create the disaster recovery endpoint within Traffic Manager as well.

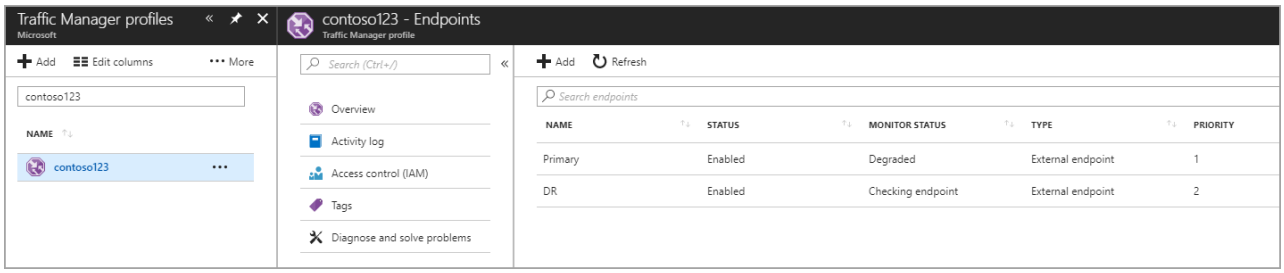


Figure - Create disaster recovery endpoints

Step 3: Set up health check and failover configuration

In this step, you set the DNS TTL to 10 seconds, which is honored by most internet-facing recursive resolvers. This configuration means that no DNS resolver will cache the information for more than 10 seconds. For the endpoint monitor settings, the path is current set at / or root, but you can customize the endpoint settings to evaluate a path, for example, prod.contoso.com/index. The example below shows the **https** as the probing protocol. However, you can choose **http** or **tcp** as well. The choice of protocol depends upon the end application. The probing interval is set to 10 seconds, which enables fast probing, and the retry is set to 3. As a result, Traffic Manager will failover to the second endpoint if three consecutive intervals register a failure. The following formula defines the total time for an automated failover: Time for failover = TTL + Retry * Probing interval And in this case, the value is 10 + 3 * 10 = 40 seconds (Max). If the Retry is set to 1 and TTL is set to 10 secs, then the time for failover 10 + 1 * 10 = 20 seconds. Set the Retry to a value greater than **1** to eliminate chances of failovers due to false positives or any minor network blips.

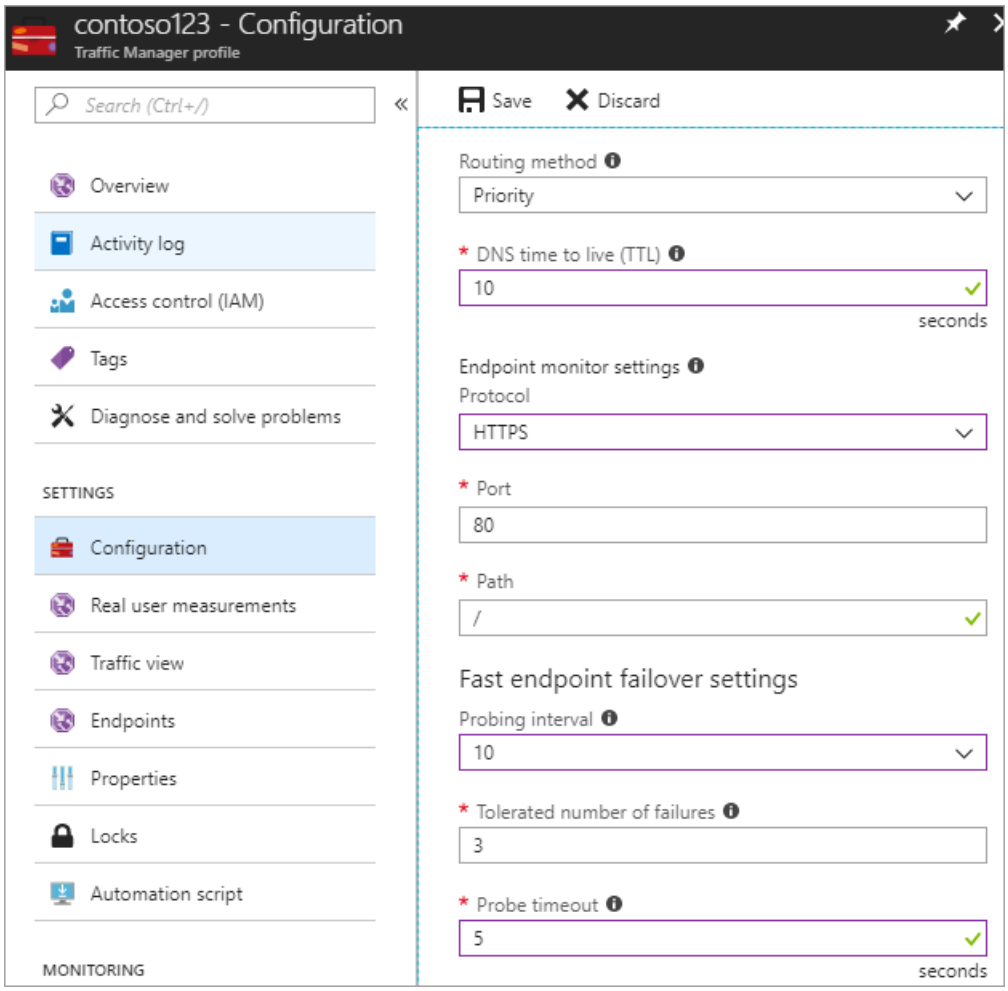


Figure - Set up health check and failover configuration

How automatic failover works using Traffic Manager

During a disaster, the primary endpoint gets probed and the status changes to **degraded** and the disaster recovery site remains **Online**. By default, Traffic Manager sends all traffic to the primary (highest-priority) endpoint. If the primary endpoint appears degraded, Traffic Manager routes the traffic to the second endpoint as long as it remains healthy. One has the option to configure more endpoints within Traffic Manager that can serve as additional failover endpoints, or, as load balancers sharing the load between endpoints.

Next steps

- Learn more about Azure Traffic Manager.
- Learn more about Azure DNS.

How Traffic Manager Works

2/1/2020 • 4 minutes to read • Edit Online

Azure Traffic Manager enables you to control the distribution of traffic across your application endpoints. An endpoint is any Internet-facing service hosted inside or outside of Azure.

Traffic Manager provides two key benefits:

- Distribution of traffic according to one of several traffic-routing methods
- Continuous monitoring of endpoint health and automatic failover when endpoints fail

When a client attempts to connect to a service, it must first resolve the DNS name of the service to an IP address. The client then connects to that IP address to access the service.

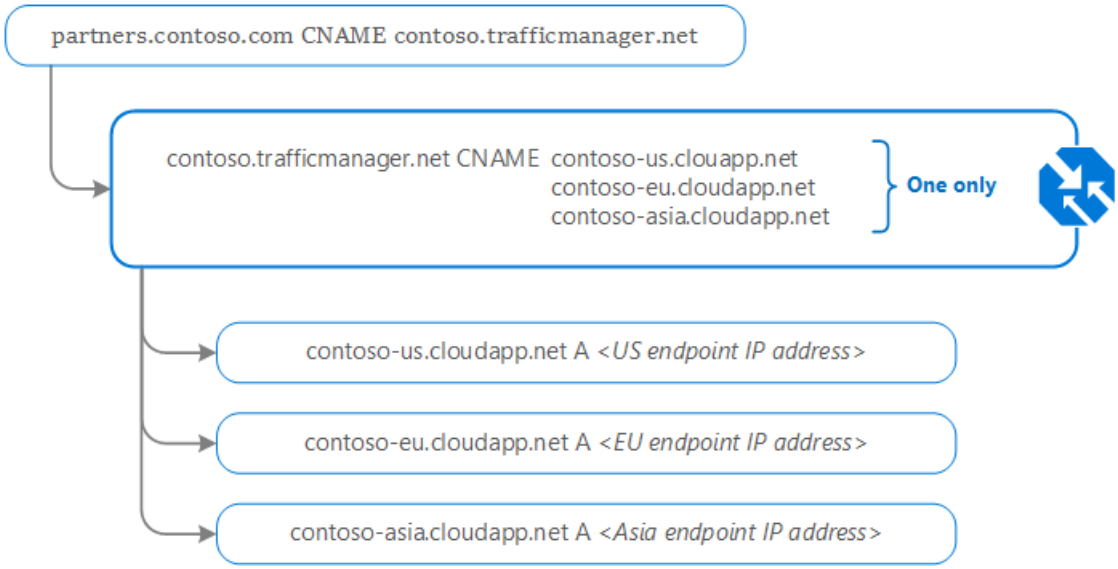
The most important point to understand is that Traffic Manager works at the DNS level. Traffic Manager uses DNS to direct clients to specific service endpoints based on the rules of the traffic-routing method. Clients connect to the selected endpoint **directly**. Traffic Manager is not a proxy or a gateway. Traffic Manager does not see the traffic passing between the client and the service.

Traffic Manager example

Contoso Corp have developed a new partner portal. The URL for this portal is https://partners.contoso.com/login.aspx. The application is hosted in three regions of Azure. To improve availability and maximize global performance, they use Traffic Manager to distribute client traffic to the closest available endpoint.

To achieve this configuration, they complete the following steps:

1. Deploy three instances of their service. The DNS names of these deployments are 'contoso-us.cloudapp.net', 'contoso-eu.cloudapp.net', and 'contoso-asia.cloudapp.net'.
2. Create a Traffic Manager profile, named 'contoso.trafficmanager.net', and configure it to use the 'Performance' traffic-routing method across the three endpoints.
3. Configure their vanity domain name, 'partners.contoso.com', to point to 'contoso.trafficmanager.net', using a DNS CNAME record.

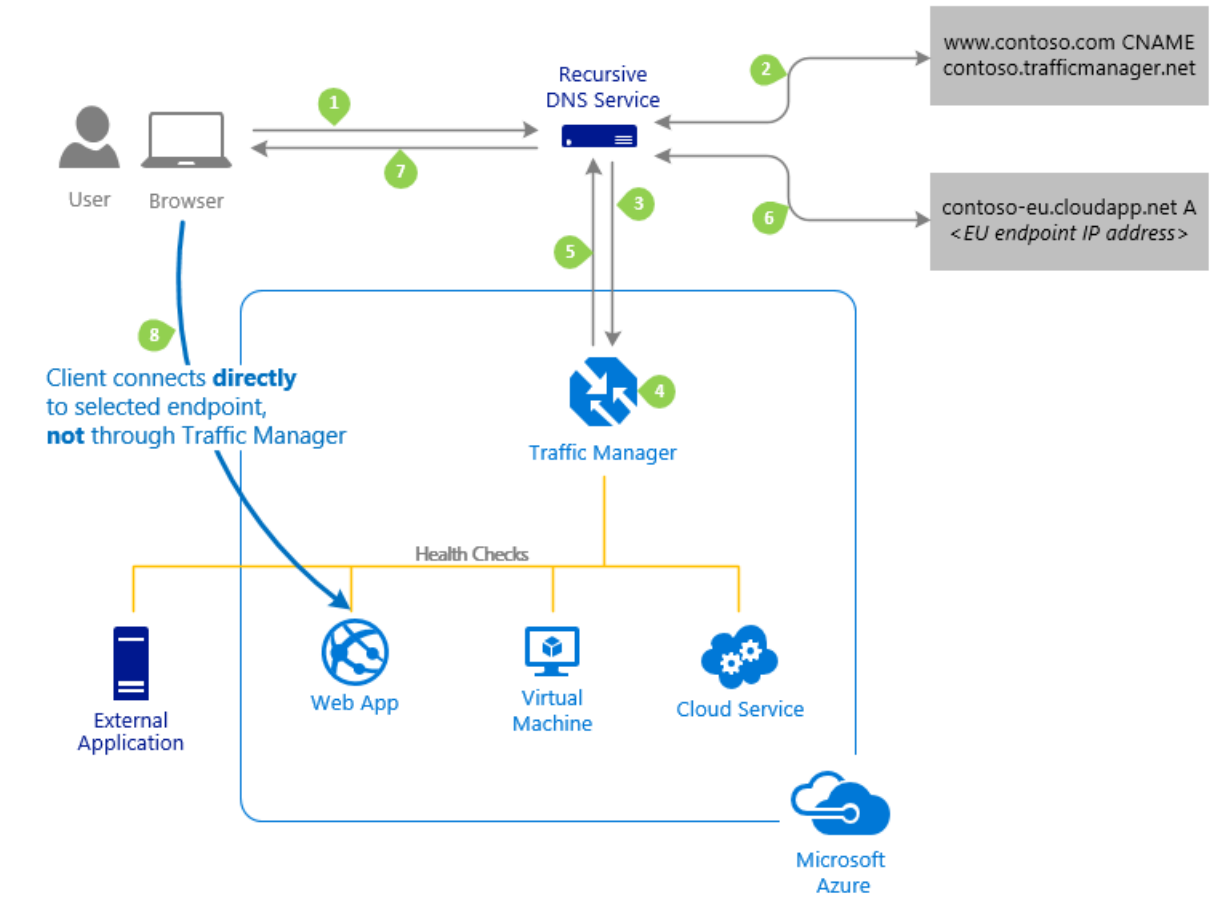


NOTE

When using a vanity domain with Azure Traffic Manager, you must use a CNAME to point your vanity domain name to your Traffic Manager domain name. DNS standards do not allow you to create a CNAME at the 'apex' (or root) of a domain. Thus you cannot create a CNAME for 'contoso.com' (sometimes called a 'naked' domain). You can only create a CNAME for a domain under 'contoso.com', such as 'www.contoso.com'. To work around this limitation, we recommend hosting your DNS domain on Azure DNS and using Alias records to point to your traffic manager profile. Alternatively you can use a simple HTTP redirect to direct requests for 'contoso.com' to an alternative name such as 'www.contoso.com'.

How clients connect using Traffic Manager

Continuing from the previous example, when a client requests the page https://partners.contoso.com/login.aspx, the client performs the following steps to resolve the DNS name and establish a connection:



1. The client sends a DNS query to its configured recursive DNS service to resolve the name 'partners.contoso.com'. A recursive DNS service, sometimes called a 'local DNS' service, does not host DNS domains directly. Rather, the client off-loads the work of contacting the various authoritative DNS services across the Internet needed to resolve a DNS name.
2. To resolve the DNS name, the recursive DNS service finds the name servers for the 'contoso.com' domain. It then contacts those name servers to request the 'partners.contoso.com' DNS record. The contoso.com DNS servers return the CNAME record that points to contoso.trafficmanager.net.
3. Next, the recursive DNS service finds the name servers for the 'trafficmanager.net' domain, which are provided by the Azure Traffic Manager service. It then sends a request for the 'contoso.trafficmanager.net' DNS record to those DNS servers.
4. The Traffic Manager name servers receive the request. They choose an endpoint based on:
    - The configured state of each endpoint (disabled endpoints are not returned)

- The current health of each endpoint, as determined by the Traffic Manager health checks. For more information, see Traffic Manager Endpoint Monitoring.
- The chosen traffic-routing method. For more information, see Traffic Manager Routing Methods.

5. The chosen endpoint is returned as another DNS CNAME record. In this case, let us suppose contoso-us.cloudapp.net is returned.
6. Next, the recursive DNS service finds the name servers for the 'cloudapp.net' domain. It contacts those name servers to request the 'contoso-us.cloudapp.net' DNS record. A DNS 'A' record containing the IP address of the US-based service endpoint is returned.
7. The recursive DNS service consolidates the results and returns a single DNS response to the client.
8. The client receives the DNS results and connects to the given IP address. The client connects to the application service endpoint directly, not through Traffic Manager. Since it is an HTTPS endpoint, the client performs the necessary SSL/TLS handshake, and then makes an HTTP GET request for the '/login.aspx' page.

The recursive DNS service caches the DNS responses it receives. The DNS resolver on the client device also caches the result. Caching enables subsequent DNS queries to be answered more quickly by using data from the cache rather than querying other name servers. The duration of the cache is determined by the 'time-to-live' (TTL) property of each DNS record. Shorter values result in faster cache expiry and thus more round-trips to the Traffic Manager name servers. Longer values mean that it can take longer to direct traffic away from a failed endpoint. Traffic Manager allows you to configure the TTL used in Traffic Manager DNS responses to be as low as 0 seconds and as high as 2,147,483,647 seconds (the maximum range compliant with RFC-1035), enabling you to choose the value that best balances the needs of your application.

FAQs

- What IP address does Traffic Manager use?
- What types of traffic can be routed using Traffic Manager?
- Does Traffic Manager support "sticky" sessions?
- Why am I seeing an HTTP error when using Traffic Manager?
- What is the performance impact of using Traffic Manager?
- What application protocols can I use with Traffic Manager?
- Can I use Traffic Manager with a "naked" domain name?
- Does Traffic Manager consider the client subnet address when handling DNS queries?
- What is DNS TTL and how does it impact my users?
- How high or low can I set the TTL for Traffic Manager responses?
- How can I understand the volume of queries coming to my profile?

Next steps

Learn more about Traffic Manager endpoint monitoring and automatic failover.

Learn more about Traffic Manager traffic routing methods.

Traffic Manager Frequently Asked Questions (FAQ)

2/1/2020 • 39 minutes to read • Edit Online

Traffic Manager basics

What IP address does Traffic Manager use?

As explained in How Traffic Manager Works, Traffic Manager works at the DNS level. It sends DNS responses to direct clients to the appropriate service endpoint. Clients then connect to the service endpoint directly, not through Traffic Manager.

Therefore, Traffic Manager does not provide an endpoint or IP address for clients to connect to. If you want static IP address for your service, that must be configured at the service, not in Traffic Manager.

What types of traffic can be routed using Traffic Manager?

As explained in How Traffic Manager Works, a Traffic Manager endpoint can be any internet facing service hosted inside or outside of Azure. Hence, Traffic Manager can route traffic that originates from the public internet to a set of endpoints that are also internet facing. If you have endpoints that are inside a private network (for example, an internal version of Azure Load Balancer) or have users making DNS requests from such internal networks, then you cannot use Traffic Manager to route this traffic.

Does Traffic Manager support "sticky" sessions?

As explained in How Traffic Manager Works, Traffic Manager works at the DNS level. It uses DNS responses to direct clients to the appropriate service endpoint. Clients connect to the service endpoint directly, not through Traffic Manager. Therefore, Traffic Manager does not see the HTTP traffic between the client and the server.

Additionally, the source IP address of the DNS query received by Traffic Manager belongs to the recursive DNS service, not the client. Therefore, Traffic Manager has no way to track individual clients and cannot implement 'sticky' sessions. This limitation is common to all DNS-based traffic management systems and is not specific to Traffic Manager.

Why am I seeing an HTTP error when using Traffic Manager?

As explained in How Traffic Manager Works, Traffic Manager works at the DNS level. It uses DNS responses to direct clients to the appropriate service endpoint. Clients then connect to the service endpoint directly, not through Traffic Manager. Traffic Manager does not see HTTP traffic between client and server. Therefore, any HTTP error you see must be coming from your application. For the client to connect to the application, all DNS resolution steps are complete. That includes any interaction that Traffic Manager has on the application traffic flow.

Further investigation should therefore focus on the application.

The HTTP host header sent from the client's browser is the most common source of problems. Make sure that the application is configured to accept the correct host header for the domain name you are using. For endpoints using the Azure App Service, see configuring a custom domain name for a web app in Azure App Service using Traffic Manager.

What is the performance impact of using Traffic Manager?

As explained in How Traffic Manager Works, Traffic Manager works at the DNS level. Since clients connect to your service endpoints directly, there is no performance impact incurred when using Traffic Manager once the connection is established.

Since Traffic Manager integrates with applications at the DNS level, it does require an additional DNS lookup to be inserted into the DNS resolution chain. The impact of Traffic Manager on DNS resolution time is minimal. Traffic

Manager uses a global network of name servers, and uses anycast networking to ensure DNS queries are always routed to the closest available name server. In addition, caching of DNS responses means that the additional DNS latency incurred by using Traffic Manager applies only to a fraction of sessions.

The Performance method routes traffic to the closest available endpoint. The net result is that the overall performance impact associated with this method should be minimal. Any increase in DNS latency should be offset by lower network latency to the endpoint.

What application protocols can I use with Traffic Manager?

As explained in How Traffic Manager Works, Traffic Manager works at the DNS level. Once the DNS lookup is complete, clients connect to the application endpoint directly, not through Traffic Manager. Therefore, the connection can use any application protocol. If you select TCP as the monitoring protocol, Traffic Manager's endpoint health monitoring can be done without using any application protocols. If you choose to have the health verified using an application protocol, the endpoint needs to be able to respond to either HTTP or HTTPS GET requests.

Can I use Traffic Manager with a "naked" domain name?

Yes. To learn how to create an alias record for your domain name apex to reference an Azure Traffic Manager profile, see Configure an alias record to support apex domain names with Traffic Manager.

Does Traffic Manager consider the client subnet address when handling DNS queries?

Yes, in addition to the source IP address of the DNS query it receives (which usually is the IP address of the DNS resolver), when performing lookups for Geographic, Performance, and Subnet routing methods, traffic manager also considers the client subnet address if it is included in the query by the resolver making the request on behalf of the end user.
Specifically, RFC 7871 – Client Subnet in DNS Queries that provides an Extension Mechanism for DNS (EDNS0) which can pass on the client subnet address from resolvers that support it.

What is DNS TTL and how does it impact my users?

When a DNS query lands on Traffic Manager, it sets a value in the response called time-to-live (TTL). This value, whose unit is in seconds, indicates to DNS resolvers downstream on how long to cache this response. While DNS resolvers are not guaranteed to cache this result, caching it enables them to respond to any subsequent queries off the cache instead of going to Traffic Manager DNS servers. This impacts the responses as follows:

- a higher TTL reduces the number of queries that land on the Traffic Manager DNS servers, which can reduce the cost for a customer since number of queries served is a billable usage.
- a higher TTL can potentially reduce the time it takes to do a DNS lookup.
- a higher TTL also means that your data does not reflect the latest health information that Traffic Manager has obtained through its probing agents.

How high or low can I set the TTL for Traffic Manager responses?

You can set, at a per profile level, the DNS TTL to be as low as 0 seconds and as high as 2,147,483,647 seconds (the maximum range compliant with RFC-1035). A TTL of 0 means that downstream DNS resolvers do not cache query responses and all queries are expected to reach the Traffic Manager DNS servers for resolution.

How can I understand the volume of queries coming to my profile?

One of the metrics provided by Traffic Manager is the number of queries responded by a profile. You can get this information at a profile level aggregation or you can split it up further to see the volume of queries where specific endpoints were returned. In addition, you can set up alerts to notify you if the query response volume crosses the conditions you have set. For more details, Traffic Manager metrics and alerts.

Traffic Manager Geographic traffic routing method

What are some use cases where geographic routing is useful?

Geographic routing type can be used in any scenario where an Azure customer needs to distinguish their users based on geographic regions. For example, using the Geographic traffic routing method, you can give users from specific regions a different user experience than those from other regions. Another example is complying with local data sovereignty mandates that require that users from a specific region be served only by endpoints in that region.

How do I decide if I should use Performance routing method or Geographic routing method?

The key difference between these two popular routing methods is that in Performance routing method your primary goal is to send traffic to the endpoint that can provide the lowest latency to the caller, whereas, in Geographic routing the primary goal is to enforce a geo fence for your callers so that you can deliberately route them to a specific endpoint. The overlap happens since there is a correlation between geographical closeness and lower latency, although this is not always true. There might be an endpoint in a different geography that can provide a better latency experience for the caller and in that case Performance routing will send the user to that endpoint but Geographic routing will always send them to the endpoint you have mapped for their geographic region. To further make it clear, consider the following example - with Geographic routing you can make uncommon mappings such as send all traffic from Asia to endpoints in the US and all US traffic to endpoints in Asia. In that case, Geographic routing will deliberately do exactly what you have configured it to do and performance optimization is not a consideration.

> **NOTE**
> There may be scenarios where you might need both performance and geographic routing capabilities, for these scenarios nested profiles can be great choice. For example, you can set up a parent profile with geographic routing where you send all traffic from North America to a nested profile that has endpoints in the US and use performance routing to send those traffic to the best endpoint within that set.

What are the regions that are supported by Traffic Manager for geographic routing?

The country/region hierarchy that is used by Traffic Manager can be found here. While this page is kept up-to-date with any changes, you can also programmatically retrieve the same information by using the Azure Traffic Manager REST API.

How does traffic manager determine where a user is querying from?

Traffic Manager looks at the source IP of the query (this most likely is a local DNS resolver doing the querying on behalf of the user) and uses an internal IP to region map to determine the location. This map is updated on an ongoing basis to account for changes in the internet.

Is it guaranteed that Traffic Manager can correctly determine the exact geographic location of the user in every case?

No, Traffic Manager cannot guarantee that the geographic region we infer from the source IP address of a DNS query will always correspond to the user's location due to the following reasons:

- First, as described in the previous FAQ, the source IP address we see is that of a DNS resolver doing the lookup on behalf of the user. While the geographic location of the DNS resolver is a good proxy for the geographic location of the user, it can also be different depending upon the footprint of the DNS resolver service and the specific DNS resolver service a customer has chosen to use. As an example, a customer located in Malaysia could specify in their device's settings use a DNS resolver service whose DNS server in Singapore might get picked to handle the query resolutions for that user/device. In that case, Traffic Manager can only see the resolver's IP address that corresponds to the Singapore location. Also, see the earlier FAQ regarding client subnet address support on this page.
- Second, Traffic Manager uses an internal map to do the IP address to geographic region translation. While this map is validated and updated on an ongoing basis to increase its accuracy and account for the evolving nature of the internet, there is still the possibility that our information is not an exact representation of the geographic location of all the IP addresses.

Does an endpoint need to be physically located in the same region as the one it is configured with for geographic routing?

No, the location of the endpoint imposes no restrictions on which regions can be mapped to it. For example, an endpoint in US-Central Azure region can have all users from India directed to it.

Can I assign geographic regions to endpoints in a profile that is not configured to do geographic routing?

Yes, if the routing method of a profile is not geographic, you can use the Azure Traffic Manager REST API to assign geographic regions to endpoints in that profile. In the case of non-geographic routing type profiles, this configuration is ignored. If you change such a profile to geographic routing type at a later time, Traffic Manager can use those mappings.

Why am I getting an error when I try to change the routing method of an existing profile to Geographic?

All the endpoints under a profile with geographic routing need to have at least one region mapped to it. To convert an existing profile to geographic routing type, you first need to associate geographic regions to all its endpoints using the Azure Traffic Manager REST API before changing the routing type to geographic. If using portal, first delete the endpoints, change the routing method of the profile to geographic and then add the endpoints along with their geographic region mapping.

Why is it strongly recommended that customers create nested profiles instead of endpoints under a profile with geographic routing enabled?

A region can be assigned to only one endpoint within a profile if it is using the geographic routing method. If that endpoint is not a nested type with a child profile attached to it, if that endpoint going unhealthy, Traffic Manager continues to send traffic to it since the alternative of not sending any traffic isn't any better. Traffic Manager does not failover to another endpoint, even when the region assigned is a "parent" of the region assigned to the endpoint that went unhealthy (for example, if an endpoint that has region Spain goes unhealthy, we do not failover to another endpoint that has the region Europe assigned to it). This is done to ensure that Traffic Manager respects the geographic boundaries that a customer has setup in their profile. To get the benefit of failing over to another endpoint when an endpoint goes unhealthy, it is recommended that geographic regions be assigned to nested profiles with multiple endpoints within it instead of individual endpoints. In this way, if an endpoint in the nested child profile fails, traffic can failover to another endpoint inside the same nested child profile.

Are there any restrictions on the API version that supports this routing type?

Yes, only API version 2017-03-01 and newer supports the Geographic routing type. Any older API versions cannot be used to created profiles of Geographic routing type or assign geographic regions to endpoints. If an older API version is used to retrieve profiles from an Azure subscription, any profile of Geographic routing type is not returned. In addition, when using older API versions, any profile returned that has endpoints with a geographic region assignment, does not have its geographic region assignment shown.

Traffic Manager Subnet traffic routing method

What are some use cases where subnet routing is useful?

Subnet routing allows you to differentiate the experience you deliver for specific sets of users identified by the source IP of their DNS requests IP address. An example would be showing different content if users are connecting to a website from your corporate HQ. Another would be restricting users from certain ISPs to only access endpoints that support only IPv4 connections if those ISPs have sub-par performance when IPv6 is used. Another reason to use Subnet routing method is in conjunction with other profiles in a nested profile set. For example, if you want to use Geographic routing method for geo-fencing your users, but for a specific ISP you want to do a different routing method, you can have a profile withy Subnet routing method as the parent profile and override that ISP to use a specific child profile and have the standard Geographic profile for everyone else.

How does Traffic Manager know the IP address of the end user?

End user devices typically use a DNS resolver to do the DNS lookup on their behalf. The outgoing IP of such resolvers is what Traffic Manager sees as the source IP. In addition, Subnet routing method also looks to see if

there is EDNS0 Extended Client Subnet (ECS) information that was passed with the request. If ECS information is present, that is the address used to determine the routing. In the absence of ECS information, the source IP of the query is used for routing purposes.

How can I specify IP addresses when using Subnet routing?

The IP addresses to associate with an endpoint can be specified in two ways. First, you can use the quad dotted decimal octet notation with a start and end addresses to specify the range (for example, 1.2.3.4-5.6.7.8 or 3.4.5.6-3.4.5.6). Second, you can use the CIDR notation to specify the range (for example, 1.2.3.0/24). You can specify multiple ranges and can use both notation types in a range set. A few restrictions apply.

- You cannot have overlap of address ranges since each IP needs to be mapped to only a single endpoint
- The start address cannot be more than the end address
- In the case of the CIDR notation, the IP address before the '/' should be the start address of that range (for example, 1.2.3.0/24 is valid but 1.2.3.4.4/24 is NOT valid)

How can I specify a fallback endpoint when using Subnet routing?

In a profile with Subnet routing, if you have an endpoint with no subnets mapped to it, any request that does not match with other endpoints will be directed to here. It is highly recommended that you have such a fallback endpoint in your profile since Traffic Manager will return a NXDOMAIN response if a request comes in and it is not mapped to any endpoints or if it is mapped to an endpoint but that endpoint is unhealthy.

What happens if an endpoint is disabled in a Subnet routing type profile?

In a profile with Subnet routing, if you have an endpoint with that is disabled, Traffic Manager will behave as if that endpoint and the subnet mappings it has does not exist. If a query that would've matched with its IP address mapping is received and the endpoint is disabled, Traffic Manager will return a fallback endpoint (one with no mappings) or if such an endpoint is not present, will return a NXDOMAIN response.

Traffic Manager MultiValue traffic routing method

What are some use cases where MultiValue routing is useful?

MultiValue routing returns multiple healthy endpoints in a single query response. The main advantage of this is that, if an endpoint is unhealthy, the client has more options to retry without making another DNS call (which might return the same value from an upstream cache). This is applicable for availability sensitive applications that want to minimize the downtime. Another use for MultiValue routing method is if an endpoint is "dual-homed" to both IPv4 and IPv6 addresses and you want to give the caller both options to choose from when it initiates a connection to the endpoint.

How many endpoints are returned when MultiValue routing is used?

You can specify the maximum number of endpoints to be returned and MultiValue will return no more than that many healthy endpoints when a query is received. The maximum possible value for this configuration is 10.

Will I get the same set of endpoints when MultiValue routing is used?

We cannot guarantee that the same set of endpoints will be returned in each query. This is also affected by the fact that some of the endpoints might go unhealthy at which point they will not be included in the response

Real User Measurements

What are the benefits of using Real User Measurements?

When you use performance routing method, Traffic Manager picks the best Azure region for your end user to connect to by inspecting the source IP and EDNS Client Subnet (if passed in) and checking it against the network latency intelligence the service maintains. Real User Measurements enhances this for your end user base by having their experience contribute to this latency table in addition to ensuring that this table adequately spans the end user networks from where your end users connect to Azure. This leads to an increased accuracy in the routing of your

end user.

Can I use Real User Measurements with non-Azure regions?

Real User Measurements measures and reports on only the latency to reach Azure regions. If you are using performance-based routing with endpoints hosted in non-Azure regions, you can still benefit from this feature by having increased latency information about the representative Azure region you had selected to be associated with this endpoint.

Which routing method benefits from Real User Measurements?

The additional information gained through Real User Measurements are applicable only for profiles that use the performance routing method. The Real User Measurements link is available from all the profiles when you view it through the Azure portal.

Do I need to enable Real User Measurements each profile separately?

No, you only need to enable it once per subscription and all the latency information measured and reported are available to all profiles.

How do I turn off Real User Measurements for my subscription?

You can stop accruing charges related to Real User Measurements when you stop collecting and sending back latency measurements from your client application. For example, when measurement JavaScript embedded in web pages, you can stop using this feature by removing the JavaScript or by turning off its invocation when the page is rendered.

You can also turn off Real User Measurements by deleting your key. Once you delete the key, any measurements sent to Traffic Manager with that key are discarded.

Can I use Real User Measurements with client applications other than web pages?

Yes, Real User Measurements is designed to ingest data collected through different type of end user clients. This FAQ will be updated as new types of client applications get supported.

How many measurements are made each time my Real User Measurements enabled web page is rendered?

When Real User Measurements is used with the measurement JavaScript provided, each page rendering results in six measurements being taken. These are then reported back to the Traffic Manager service. You are charged for this feature based on the number of measurements reported to Traffic Manager service. For example, if the user navigates away from your webpage while the measurements are being taken but before it was reported, those measurements are not considered for billing purposes.

Is there a delay before Real User Measurements script runs in my webpage?

No, there is no programmed delay before the script is invoked.

Can I use Real User Measurements with only the Azure regions I want to measure?

No, each time it is invoked, the Real User Measurements script measures a set of six Azure regions as determined by the service. This set changes between different invocations and when a large number of such invocations happen, the measurement coverage spans across different Azure regions.

Can I limit the number of measurements made to a specific number?

The measurement JavaScript is embedded within your webpage and you are in complete control over when to start and stop using it. As long as the Traffic Manager service receives a request for a list of Azure regions to be measured, a set of regions are returned.

Can I see the measurements taken by my client application as part of Real User Measurements?

Since the measurement logic is run from your client application, you are in full control of what happens including seeing the latency measurements. Traffic Manager does not report an aggregate view of the measurements received under the key linked to your subscription.

Can I modify the measurement script provided by Traffic Manager?

While you are in control of what is embedded on your web page, we strongly discourage you from making any changes to the measurement script to ensure that it measures and reports the latencies correctly.

Will it be possible for others to see the key I use with Real User Measurements?

When you embed the measurement script to a web page it will be possible for others to see the script and your Real User Measurements (RUM) key. But it is important to know that this key is different from your subscription id and is generated by Traffic Manager to be used only for this purpose. Knowing your RUM key will not compromise your Azure account safety.

Can others abuse my RUM key?

While it is possible for others to use your key to send wrong information to Azure, a few wrong measurements will not change the routing since it is taken into account along with all the other measurements we receive. If you need to change your keys, you can re-generate the key at which point the old key becomes discarded.

Do I need to put the measurement JavaScript in all my web pages?

Real User Measurements delivers more value as the number of measurements increase. Having said that, it is your decision as to whether you need to put it in all your web pages or a select few. Our recommendation is to start by putting it in your most visited page where a user is expected to stay on that page five seconds or more.

Can information about my end users be identified by Traffic Manager if I use Real User Measurements?

When the provided measurement JavaScript is used, Traffic Manager will have visibility into the client IP address of the end user and the source IP address of the local DNS resolver they use. Traffic Manager uses the client IP address only after having it truncated to not be able to identify the specific end user who sent the measurements.

Does the webpage measuring Real User Measurements need to be using Traffic Manager for routing?

No, it doesn't need to use Traffic Manager. The routing side of Traffic Manager operates separately from the Real User Measurement part and although it is a great idea to have them both in the same web property, they don't need to be.

Do I need to host any service on Azure regions to use with Real User Measurements?

No, you don't need to host any server-side component on Azure for Real User Measurements to work. The single pixel image downloaded by the measurement JavaScript and the service running it in different Azure regions is hosted and managed by Azure.

Will my Azure bandwidth usage increase when I use Real User Measurements?

As mentioned in the previous answer, the server-side components of Real User Measurements are owned and managed by Azure. This means your Azure bandwidth usage will not increase because you use Real User Measurements. This does not include any bandwidth usage outside of what Azure charges. We minimize the bandwidth used by downloading only a single pixel image to measurement the latency to an Azure region.

Traffic View

What does Traffic View do?

Traffic View is a feature of Traffic Manager that helps you understand more about your users and how their experience is. It uses the queries received by Traffic Manager and the network latency intelligence tables that the service maintains to provide you with the following:

- The regions from where your users are connecting to your endpoints in Azure.
- The volume of users connecting from these regions.
- The Azure regions to which they are getting routed to.
- Their latency experience to these Azure regions.

This information is available for you to consume through geographical map overlay and tabular views in the portal

in addition to being available as raw data for you to download.

How can I benefit from using Traffic View?

Traffic View gives you the overall view of the traffic your Traffic Manager profiles receive. In particular, it can be used to understand where your user base connects from and equally importantly what their average latency experience is. You can then use this information to find areas in which you need to focus, for example, by expanding your Azure footprint to a region that can serve those users with lower latency. Another insight you can derive from using Traffic View is to see the patterns of traffic to different regions which in turn can help you make decisions on increasing or decreasing invent in those regions.

How is Traffic View different from the Traffic Manager metrics available through Azure monitor?

Azure Monitor can be used to understand at an aggregate level the traffic received by your profile and its endpoints. It also enables you to track the health status of the endpoints by exposing the health check results. When you need to go beyond these and understand your end user's experience connecting to Azure at a regional level, Traffic View can be used to achieve that.

Does Traffic View use EDNS Client Subnet information?

The DNS queries served by Azure Traffic Manager do consider ECS information to increase the accuracy of the routing. But when creating the data set that shows where the users are connecting from, Traffic View is using only the IP address of the DNS resolver.

How many days of data does Traffic View use?

Traffic View creates its output by processing the data from the seven days preceding the day before when it is viewed by you. This is a moving window and the latest data will be used each time you visit.

How does Traffic View handle external endpoints?

When you use external endpoints hosted outside Azure regions in a Traffic Manager profile you can choose to have it mapped to an Azure region which is a proxy for its latency characteristics (this is in fact needed if you use performance routing method). If it has this Azure region mapping, that Azure region's latency metrics will be used when creating the Traffic View output. If no Azure region is specified, the latency information will be empty in the data for those external endpoints.

Do I need to enable Traffic View for each profile in my subscription?

During the preview period, Traffic View was enabled at a subscription level. As part of the improvements we made before the general availability, you can now enable Traffic View at a profile level, allowing you to have more granular enabling of this feature. By default, Traffic View will be disabled for a profile.

> **NOTE**
> If you enabled Traffic View at a subscription level during the preview time, you now need to re-enable it for each of the profile under that subscription.

How can I turn off Traffic View?

You can turn off Traffic View for any profile using the Portal or REST API.

How does Traffic View billing work?

Traffic View pricing is based on the number of data points used to create the output. Currently, the only data type supported is the queries your profile receives. In addition, you are only billed for the processing that was done when you have Traffic View enabled. This means that, if you enable Traffic View for some time period in a month and turn it off during other times, only the data points processed while you had the feature enabled count towards your bill.

Traffic Manager endpoints

Can I use Traffic Manager with endpoints from multiple subscriptions?

Using endpoints from multiple subscriptions is not possible with Azure Web Apps. Azure Web Apps requires that any custom domain name used with Web Apps is only used within a single subscription. It is not possible to use Web Apps from multiple subscriptions with the same domain name.

For other endpoint types, it is possible to use Traffic Manager with endpoints from more than one subscription. In Resource Manager, endpoints from any subscription can be added to Traffic Manager, as long as the person configuring the Traffic Manager profile has read access to the endpoint. These permissions can be granted using Azure Resource Manager role-based access control (RBAC).

Can I use Traffic Manager with Cloud Service 'Staging' slots?

Yes. Cloud Service 'staging' slots can be configured in Traffic Manager as External endpoints. Health checks are still be charged at the Azure Endpoints rate.

Does Traffic Manager support IPv6 endpoints?

Traffic Manager does not currently provide IPv6-addressable name servers. However, Traffic Manager can still be used by IPv6 clients connecting to IPv6 endpoints. A client does not make DNS requests directly to Traffic Manager. Instead, the client uses a recursive DNS service. An IPv6-only client sends requests to the recursive DNS service via IPv6. Then the recursive service should be able to contact the Traffic Manager name servers using IPv4.

Traffic Manager responds with the DNS name or IP address of the endpoint. To support an IPv6 endpoint, there are two options. You can add the endpoint as a DNS name that has an associated AAAA record and Traffic Manager will health check that endpoint and return it as a CNAME record type in the query response. You can also add that endpoint directly using the IPv6 address and Traffic Manager will return a AAAA type record in the query response.

Can I use Traffic Manager with more than one Web App in the same region?

Typically, Traffic Manager is used to direct traffic to applications deployed in different regions. However, it can also be used where an application has more than one deployment in the same region. The Traffic Manager Azure endpoints do not permit more than one Web App endpoint from the same Azure region to be added to the same Traffic Manager profile.

How do I move my Traffic Manager profile's Azure endpoints to a different resource group or subscription?

Azure endpoints that are associated with a Traffic Manager profile are tracked using their resource IDs. When an Azure resource that is being used as an endpoint (for example, Public IP, Classic Cloud Service, WebApp, or another Traffic Manager profile used in a nested manner) is moved to a different resource group or subscription, its resource ID changes. In this scenario, currently, you must update the Traffic Manager profile by first deleting and then adding back the endpoints to the profile.

Traffic Manager endpoint monitoring

Is Traffic Manager resilient to Azure region failures?

Traffic Manager is a key component of the delivery of highly available applications in Azure. To deliver high availability, Traffic Manager must have an exceptionally high level of availability and be resilient to regional failure.

By design, Traffic Manager components are resilient to a complete failure of any Azure region. This resilience applies to all Traffic Manager components: the DNS name servers, the API, the storage layer, and the endpoint monitoring service.

In the unlikely event of an outage of an entire Azure region, Traffic Manager is expected to continue to function normally. Applications deployed in multiple Azure regions can rely on Traffic Manager to direct traffic to an available instance of their application.

How does the choice of resource group location affect Traffic Manager?

Traffic Manager is a single, global service. It is not regional. The choice of resource group location makes no

difference to Traffic Manager profiles deployed in that resource group.

Azure Resource Manager requires all resource groups to specify a location, which determines the default location for resources deployed in that resource group. When you create a Traffic Manager profile, it is created in a resource group. All Traffic Manager profiles use **global** as their location, overriding the resource group default.

How do I determine the current health of each endpoint?

The current monitoring status of each endpoint, in addition to the overall profile, is displayed in the Azure portal. This information also is available via the Traffic Monitor REST API, PowerShell cmdlets, and cross-platform Azure CLI.

You can also use Azure Monitor to track the health of your endpoints and see a visual representation of them. For more about using Azure Monitor, see the Azure Monitoring documentation.

Can I monitor HTTPS endpoints?

Yes. Traffic Manager supports probing over HTTPS. Configure **HTTPS** as the protocol in the monitoring configuration.

Traffic manager cannot provide any certificate validation, including:

- Server-side certificates are not validated
- SNI server-side certificates are not validated
- Client certificates are not supported

Do I use an IP address or a DNS name when adding an endpoint?

Traffic Manager supports adding endpoints using three ways to refer them – as a DNS name, as an IPv4 address and as an IPv6 address. If the endpoint is added as an IPv4 or IPv6 address the query response will be of record type A or AAAA, respectively. If the endpoint was added as a DNS name, then the query response will be of record type CNAME. Adding endpoints as IPv4 or IPv6 address is permitted only if the endpoint is of type **External**. All routing methods and monitoring settings are supported by the three endpoint addressing types.

What types of IP addresses can I use when adding an endpoint?

Traffic Manager allows you to use IPv4 or IPv6 addresses to specify endpoints. There are a few restrictions which are listed below:

- Addresses that correspond to reserved private IP address spaces are not allowed. These addresses include those called out in RFC 1918, RFC 6890, RFC 5737, RFC 3068, RFC 2544 and RFC 5771
- The address must not contain any port numbers (you can specify the ports to be used in the profile configuration settings)
- No two endpoints in the same profile can have the same target IP address

Can I use different endpoint addressing types within a single profile?

No, Traffic Manager does not allow you to mix endpoint addressing types within a profile, except for the case of a profile with MultiValue routing type where you can mix IPv4 and IPv6 addressing types

What happens when an incoming query's record type is different from the record type associated with the addressing type of the endpoints?

When a query is received against a profile, Traffic Manager first finds the endpoint that needs to be returned as per the routing method specified and the health status of the endpoints. It then looks at the record type requested in the incoming query and the record type associated with the endpoint before returning a response based on the table below.

For profiles with any routing method other than MultiValue:

INCOMING QUERY REQUEST	ENDPOINT TYPE	RESPONSE PROVIDED
ANY	A / AAAA / CNAME	Target Endpoint
A	A / CNAME	Target Endpoint
A	AAAA	NODATA
AAAA	AAAA / CNAME	Target Endpoint
AAAA	A	NODATA
CNAME	CNAME	Target Endpoint
CNAME	A / AAAA	NODATA

For profiles with routing method set to MultiValue:

INCOMING QUERY REQUEST	ENDPOINT TYPE	RESPONSE PROVIDED
ANY	Mix of A and AAAA	Target Endpoints
A	Mix of A and AAAA	Only Target Endpoints of type A
AAAA	Mix of A and AAAA	Only Target Endpoints of type AAAA
CNAME	Mix of A and AAAA	NODATA

Can I use a profile with IPv4 / IPv6 addressed endpoints in a nested profile?

Yes, you can with the exception that a profile of type MultiValue cannot be a parent profile in a nested profile set.

I stopped an web application endpoint in my Traffic Manager profile but I am not receiving any traffic even after I restarted it. How can I fix this?

When an Azure web application endpoint is stopped Traffic Manager stops checking its health and restarts the health checks only after it detects that the endpoint has restarted. To prevent this delay, disable and then reenable that endpoint in the Traffic Manager profile after you restart the endpoint.

Can I use Traffic Manager even if my application does not have support for HTTP or HTTPS?

Yes. You can specify TCP as the monitoring protocol and Traffic Manager can initiate a TCP connection and wait for a response from the endpoint. If the endpoint replies to the connection request with a response to establish the connection, within the timeout period, then that endpoint is marked as healthy.

What specific responses are required from the endpoint when using TCP monitoring?

When TCP monitoring is used, Traffic Manager starts a three-way TCP handshake by sending a SYN request to endpoint at the specified port. It then waits for a SYN-ACK response from the endpoint for a period of time (specified in the timeout settings).

- If a SYN-ACK response is received within the timeout period specified in the monitoring settings, then that endpoint is considered healthy. A FIN or FIN-ACK is the expected response from the Traffic Manager when it regularly terminates a socket.
- If a SYN-ACK response is received after the specified timeout, the Traffic Manager will respond with an RST to reset the connection.

How fast does Traffic Manager move my users away from an unhealthy endpoint?

Traffic Manager provides multiple settings that can help you to control the failover behavior of your Traffic Manager profile as follows:

- you can specify that the Traffic Manager probes the endpoints more frequently by setting the Probing Interval at 10 seconds. This ensures that any endpoint going unhealthy can be detected as soon as possible.
- you can specify how long to wait before a health check request times out (minimum time out value is 5 sec).
- you can specify how many failures can occur before the endpoint is marked as unhealthy. This value can be low as 0, in which case the endpoint is marked unhealthy as soon as it fails the first health check. However, using the minimum value of 0 for the tolerated number of failures can lead to endpoints being taken out of rotation due to any transient issues that may occur at the time of probing.
- you can specify the time-to-live (TTL) for the DNS response to be as low as 0. Doing so means that DNS resolvers cannot cache the response and each new query gets a response that incorporates the most up-to-date health information that the Traffic Manager has.

By using these settings, Traffic Manager can provide failovers under 10 seconds after an endpoint goes unhealthy and a DNS query is made against the corresponding profile.

How can I specify different monitoring settings for different endpoints in a profile?

Traffic Manager monitoring settings are at a per profile level. If you need to use a different monitoring setting for only one endpoint, it can be done by having that endpoint as a nested profile whose monitoring settings are different from the parent profile.

How can I assign HTTP headers to the Traffic Manager health checks to my endpoints?

Traffic Manager allows you to specify custom headers in the HTTP(S) health checks it initiates to your endpoints. If you want to specify a custom header, you can do that at the profile level (applicable to all endpoints) or specify it at the endpoint level. If a header is defined at both levels, then the one specified at the endpoint level will override the profile level one. One common use case for this is specifying host headers so that Traffic Manager requests may get routed correctly to an endpoint hosted in a multi-tenant environment. Another use case of this is to identify Traffic Manager requests from an endpoint's HTTP(S) request logs

What host header do endpoint health checks use?

If no custom host header setting is provided, the host header used by Traffic Manager is the DNS name of the endpoint target configured in the profile, if that is available.

What are the IP addresses from which the health checks originate?

Click here to view the JSON file that lists the IP addresses from which Traffic Manager health checks can originate. Review the IPs listed in the JSON file to ensure that incoming connections from these IP addresses are allowed at the endpoints to check its health status.

How many health checks to my endpoint can I expect from Traffic Manager?

The number of Traffic Manager health checks reaching your endpoint depends on the following:

- the value that you have set for the monitoring interval (smaller interval means more requests landing on your endpoint in any given time period).
- the number of locations from where the health checks originate (the IP addresses from where you can expect these checks is listed in the preceding FAQ).

How can I get notified if one of my endpoints goes down?

One of the metrics provided by Traffic Manager is the health status of endpoints in a profile. You can see this as an aggregate of all endpoints inside a profile (for example, 75% of your endpoints are healthy), or, at a per endpoint level. Traffic Manager metrics are exposed through Azure Monitor and you can use its alerting capabilities to get notifications when there is a change in the health status of your endpoint. For more details, see Traffic Manager metrics and alerts.

Traffic Manager nested profiles

How do I configure nested profiles?

Nested Traffic Manager profiles can be configured using both the Azure Resource Manager and the classic Azure REST APIs, Azure PowerShell cmdlets and cross-platform Azure CLI commands. They are also supported via the new Azure portal.

How many layers of nesting does Traffic Manger support?

You can nest profiles up to 10 levels deep. 'Loops' are not permitted.

Can I mix other endpoint types with nested child profiles, in the same Traffic Manager profile?

Yes. There are no restrictions on how you combine endpoints of different types within a profile.

How does the billing model apply for Nested profiles?

There is no negative pricing impact of using nested profiles.

Traffic Manager billing has two components: endpoint health checks and millions of DNS queries

- Endpoint health checks: There is no charge for a child profile when configured as an endpoint in a parent profile. Monitoring of the endpoints in the child profile is billed in the usual way.
- DNS queries: Each query is only counted once. A query against a parent profile that returns an endpoint from a child profile is counted against the parent profile only.

For full details, see the Traffic Manager pricing page.

Is there a performance impact for nested profiles?

No. There is no performance impact incurred when using nested profiles.

The Traffic Manager name servers traverse the profile hierarchy internally when processing each DNS query. A DNS query to a parent profile can receive a DNS response with an endpoint from a child profile. A single CNAME record is used whether you are using a single profile or nested profiles. There is no need to create a CNAME record for each profile in the hierarchy.

How does Traffic Manager compute the health of a nested endpoint in a parent profile?

The parent profile doesn't perform health checks on the child directly. Instead, the health of the child profile's endpoints are used to calculate the overall health of the child profile. This information is propagated up the nested profile hierarchy to determine the health of the nested endpoint. The parent profile uses this aggregated health to determine whether the traffic can be directed to the child.

The following table describes the behavior of Traffic Manager health checks for a nested endpoint.

CHILD PROFILE MONITOR STATUS	PARENT ENDPOINT MONITOR STATUS	NOTES
Disabled. The child profile has been disabled.	Stopped	The parent endpoint state is Stopped, not Disabled. The Disabled state is reserved for indicating that you have disabled the endpoint in the parent profile.
Degraded. At least one child profile endpoint is in a Degraded state.	Online: the number of Online endpoints in the child profile is at least the value of MinChildEndpoints. CheckingEndpoint: the number of Online plus CheckingEndpoint endpoints in the child profile is at least the value of MinChildEndpoints. Degraded: otherwise.	Traffic is routed to an endpoint of status CheckingEndpoint. If MinChildEndpoints is set too high, the endpoint is always degraded.

CHILD PROFILE MONITOR STATUS	PARENT ENDPOINT MONITOR STATUS	NOTES
Online. At least one child profile endpoint is an Online state. No endpoint is in the Degraded state.	See above.	
CheckingEndpoints. At least one child profile endpoint is 'CheckingEndpoint'. No endpoints are 'Online' or 'Degraded'	Same as above.	
Inactive. All child profile endpoints are either Disabled or Stopped, or this profile has no endpoints.	Stopped	

Next steps:

- Learn more about Traffic Manager endpoint monitoring and automatic failover.
- Learn more about Traffic Manager traffic routing methods.

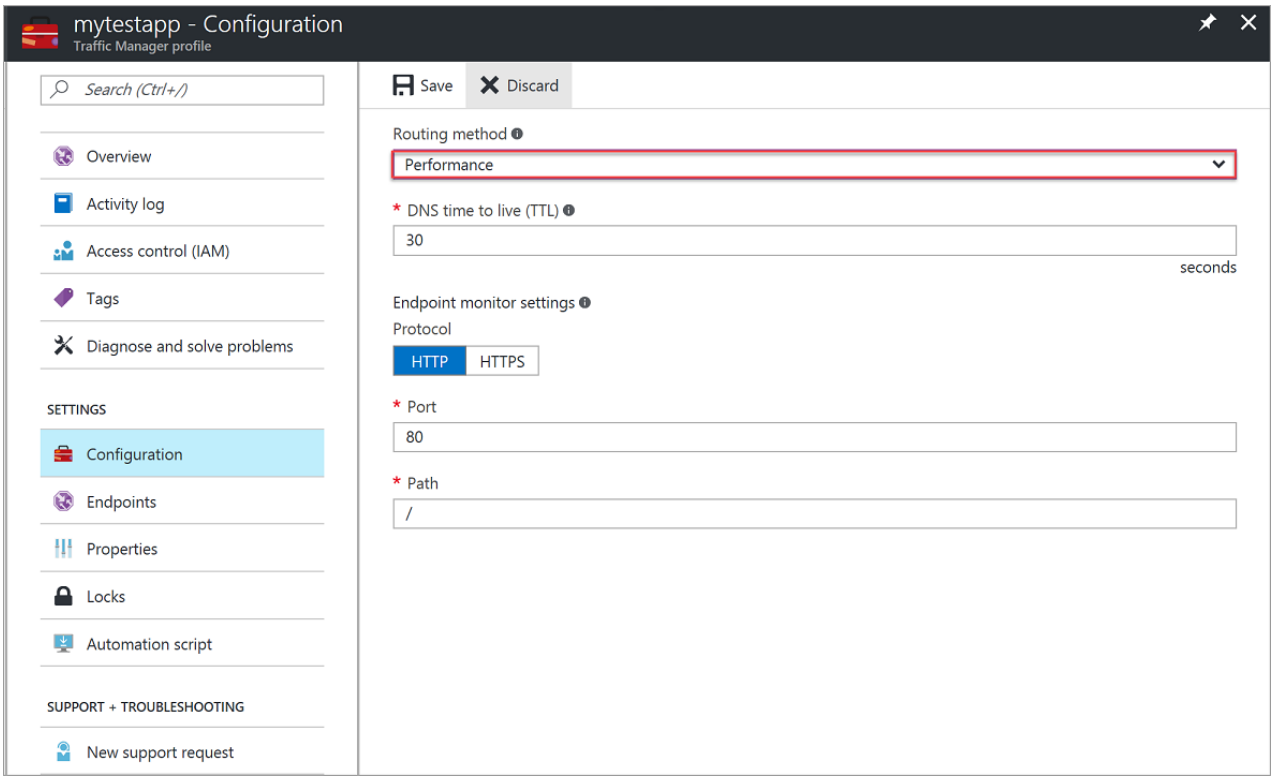
Configure the performance traffic routing method

2/1/2020 • 2 minutes to read • Edit Online

The Performance traffic routing method allows you to direct traffic to the endpoint with the lowest latency from the client's network. Typically, the datacenter with the lowest latency is the closest in geographic distance. This traffic routing method cannot account for real-time changes in network configuration or load.

To configure performance routing method

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. In the portal's search bar, search for the **Traffic Manager profiles** and then click the profile name that you want to configure the routing method for.
3. In the **Traffic Manager profile** blade, verify that both the cloud services and websites that you want to include in your configuration are present.
4. In the **Settings** section, click **Configuration**, and in the **Configuration** blade, complete as follows:
   a. For **traffic routing method settings**, for **Routing method** select **Performance**.
   b. Set the **Endpoint monitor settings** identical for all every endpoint within this profile as follows:
      a. Select the appropriate **Protocol**, and specify the **Port** number.
      b. For **Path** type a forward slash /. To monitor endpoints, you must specify a path and filename. A forward slash "/" is a valid entry for the relative path and implies that the file is in the root directory (default).
      c. At the top of the page, click **Save**.
5. Test the changes in your configuration as follows:
   a. In the portal's search bar, search for the Traffic Manager profile name and click the Traffic Manager profile in the results that the displayed.
   b. In the **Traffic Manager** profile blade, click **Overview**.
   c. The **Traffic Manager profile** blade displays the DNS name of your newly created Traffic Manager profile. This can be used by any clients (for example, by navigating to it using a web browser) to get routed to the right endpoint as determined by the routing type. In this case all requests are routed to the endpoint with the lowest latency from the client's network.
6. Once your Traffic Manager profile is working, edit the DNS record on your authoritative DNS server to point your company domain name to the Traffic Manager domain name.



Next steps

- Learn about weighted traffic routing method.
- Learn about priority routing method.
- Learn about geographic routing method.
- Learn how to test Traffic Manager settings.

Configure MultiValue routing method in Traffic Manager

2/1/2020 • 2 minutes to read • Edit Online

This article describes how to configure the MultiValue traffic-routing method. The **Multivalue** traffic routing method allows you to return multiple healthy endpoints and helps increase the reliability of your application since clients have more options to retry without having to do another DNS lookup. MultiValue routing is enabled only for profiles which have all their endpoints specified using IPv4 or IPv6 addresses. When a query is received for this profile, all healthy endpoints are returned based on the configurable maximum return count specified.

> **NOTE**
> At this time adding endpoints using IPv4 or IPv6 addresses is supported only for endpoints of type **External** and hence MultiValue routing is also supported only for such endpoints.

Sign in to Azure

Sign in to the Azure portal at https://portal.azure.com.

Create a resource group

Create a resource group for the Traffic Manager profile.

1. On the left pane of the Azure portal, select **Resource groups**.
2. In **Resource groups**, on the top of the page, select **Add**.
3. In **Resource group name**, type a name *myResourceGroupTM1*. For **Resource group location**, select **East US**, and then select **OK**.

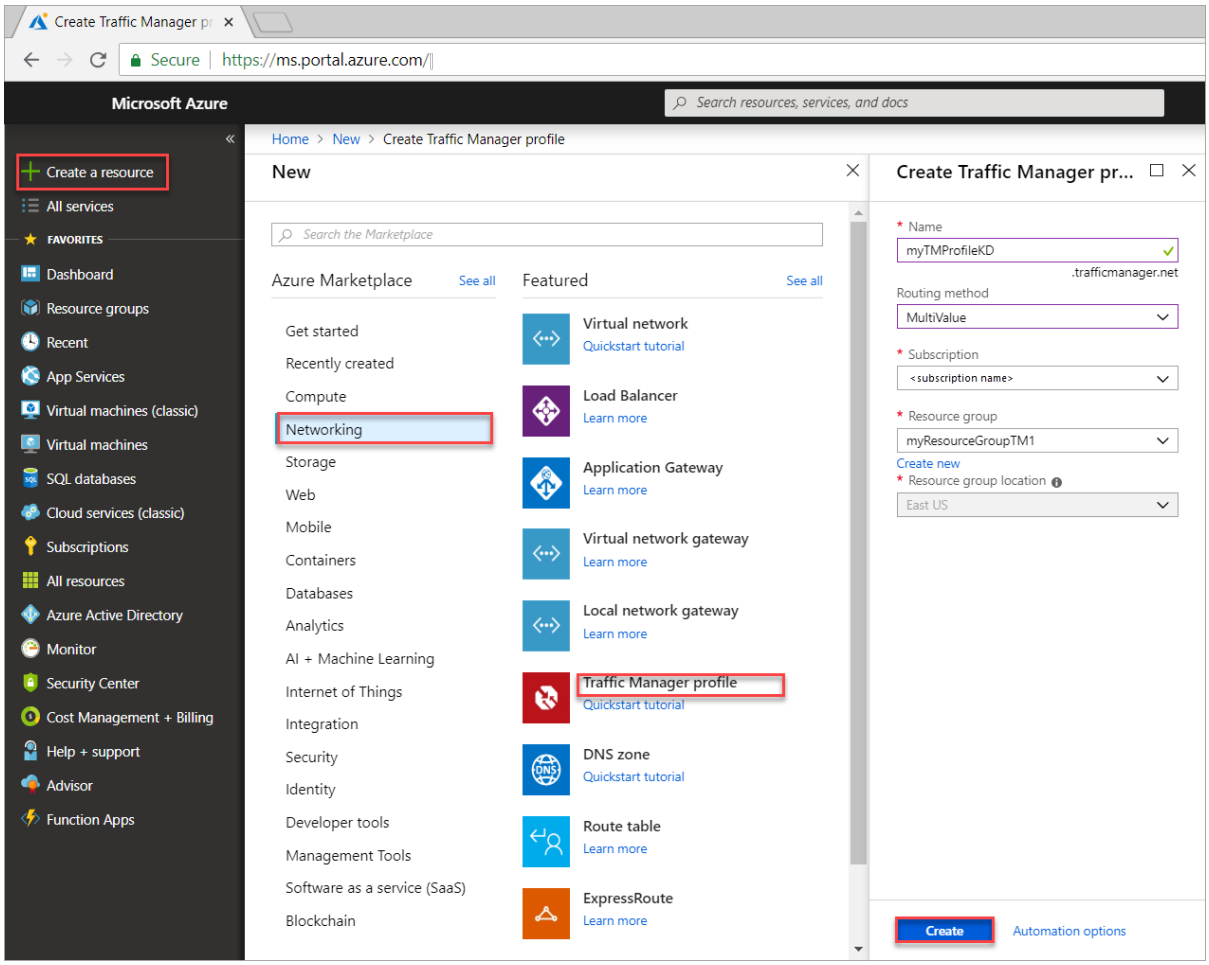
Create a Traffic Manager profile

Create a Traffic Manager profile that directs user traffic by sending them to the endpoint with lowest latency.

1. On the top left-hand side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
2. In **Create Traffic Manager profile**, enter or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	This name needs to be unique within the trafficmanager.net zone and results in the DNS name, trafficmanager.net that is used to access your Traffic Manager profile.
Routing method	Select the **Multivalue** routing method.
Subscription	Select your subscription.

SETTING	VALUE
Resource group	Select *myResourceGroupTM1*.
Location	This setting refers to the location of the resource group, and has no impact on the Traffic Manager profile that will be deployed globally.



Add Traffic Manager endpoints

Add two IP addresses as external endpoints to the MultiValue Traffic Manager profile that you created in the preceding step.

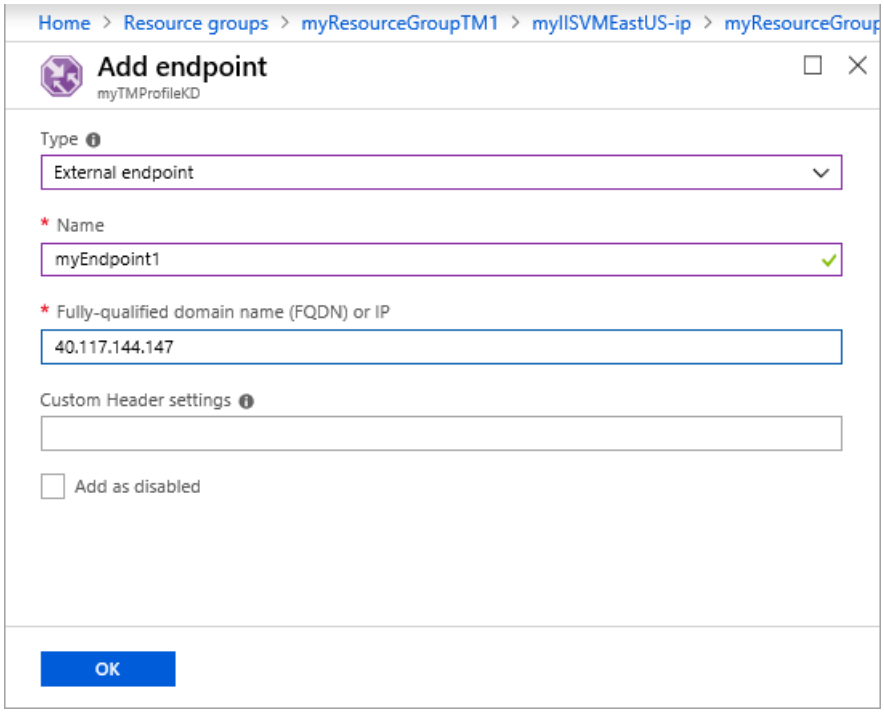
1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section and select the profile in the results that the displayed.
2. In **Traffic Manager profile**, in the **Settings** section, click **Endpoints**, and then click **Add**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

SETTING	VALUE
Type	External endpoint
Name	myEndpoint1

SETTING	VALUE
Fully qualified domain name (FQDN) or IP	Type the Public IP address of the endpoint that you want to add to this Traffic Manager profile

4. Repeat steps 2 and 3 to add another endpoint named *myEndpoint2*, for **Fully qualified domain name (FQDN) or IP**, enter the public IP address of the second endpoint.

5. When the addition of both endpoints is complete, they are displayed in **Traffic Manager profile** along with their monitoring status as **Online**.



Next steps

- Learn about weighted traffic routing method.
- Learn about priority routing method.
- Learn more about performance routing method
- Learn about geographic routing method.

Direct traffic to specific endpoints based on user subnet using Traffic Manager

2/1/2020 • 10 minutes to read • Edit Online

This article describes how to configure the subnet traffic-routing method. The **Subnet** traffic-routing method allows you to map a set of IP address ranges to specific endpoints and when a request is received by Traffic Manager, it inspects the source IP of the request and returns the endpoint associated with it.

In the scenario discussed in this article, using subnet routing, depending on the IP address of the user's query, traffic is either routed to an internal website or a production website.

If you don't have an Azure subscription, create a free account before you begin.

Prerequisites

In order to see the Traffic Manager in action, this tutorial requires that you deploy the following:

- two basic websites running in different Azure regions - **East US** (serves as internal website) and **West Europe** (serves as production website).
- two test VMs for testing the Traffic Manager - one VM in **East US** and the second VM in **West Europe**.

The test VMs are used to illustrate how Traffic Manager routes user traffic to the internal website or the production website based on subnet from where the user query originates.

Sign in to Azure

Sign in to the Azure portal at https://portal.azure.com.

Create websites

In this section, you create two website instances that provide the two service endpoints for the Traffic Manager profile in two Azure regions. Creating the two websites includes the following steps:

1. Create two VMs for running a basic website - one in **East US**, and the other in **West Europe**.
2. Install IIS server on each VM and update the default website page that describes the VM name that a user is connected to when visiting the website.

Create VMs for running websites

In this section, you create two VMs *myEndpointVMEastUS* and *myEndpointVMWEurope* in the **East US** and **West Europe** Azure regions.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2016 VM**.
2. Enter, or select, the following information for **Basics**, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	myIISVMEastUS
User name	Enter a user name of your choosing.

SETTING	VALUE
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
Resource group	Select **New** and then type *myResourceGroupTM1*.
Location	Select **East US**.

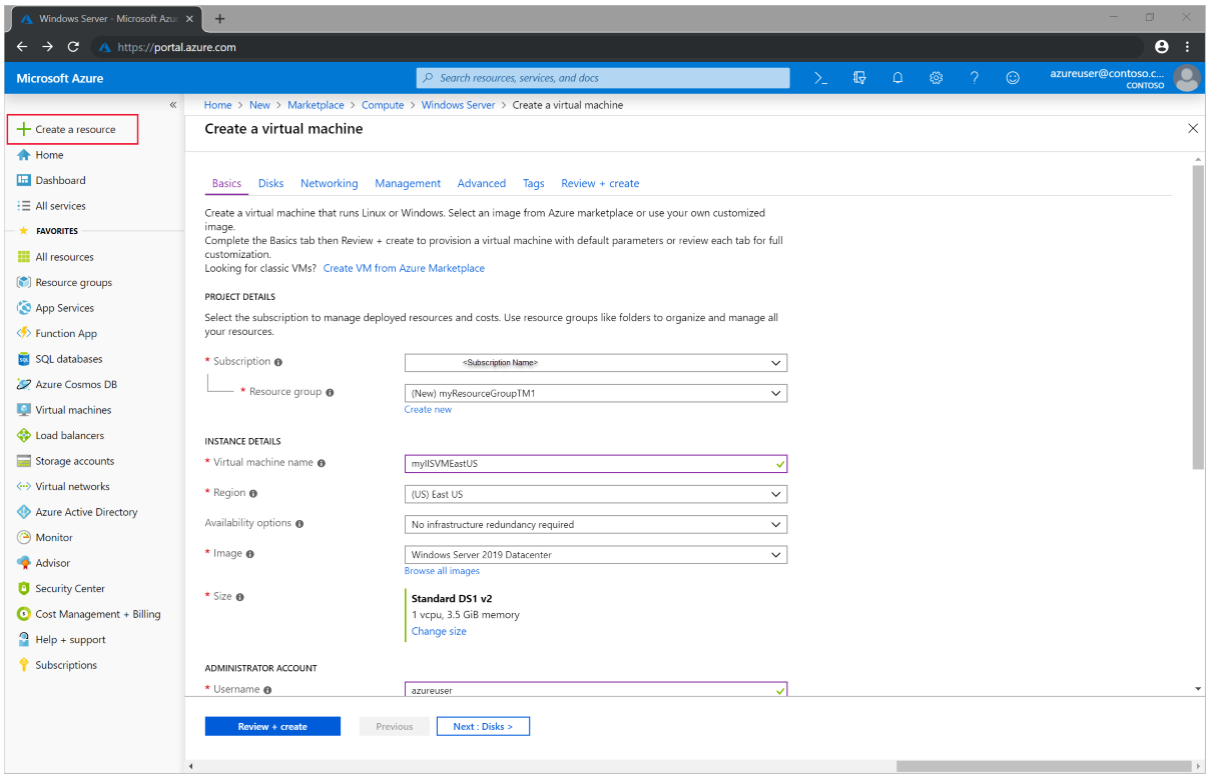
3. Select a VM size under **Choose a size**.
4. Select the following values for **Settings**, then select **OK**:

SETTING	VALUE
Virtual network	Select **Virtual network**, in **Create virtual network**, for **Name**, enter *myVNet1*, for subnet, enter *mySubnet*.
Network Security Group	Select **Basic**, and in **Select public inbound ports** drop-down, select **HTTP** and **RDP**
Boot diagnostics	Select **Disabled**.

5. Under **Create** in the **Summary**, select **Create** to start the VM deployment.
6. Complete steps 1-6 again, with the following changes:

SETTING	VALUE
Resource group	Select **New**, and then type *myResourceGroupTM2*
Location	West Europe
VM Name	myIISVMWEurope
Virtual network	Select **Virtual network**, in **Create virtual network**, for **Name**, enter *myVNet2*, for subnet, enter *mySubnet*.

7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.



Install IIS and customize the default web page

In this section, you install the IIS server on the two VMs - *myIISVMEastUS* & *myIISVMWEurope*, and then update the default website page. The customized website page shows the name of the VM that you are connecting to when you visit the website from a web browser.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.
3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.
5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.
6. On the server desktop, navigate to **Windows Administrative Tools**>**Server Manager**.
7. Launch Windows PowerShell on *myIISVMEastUS* and using the following commands to install IIS server and update the default htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from my test website server -
" + $env:computername)
```

8. Close the RDP connection with *myIISVMEastUS*.
9. Repeat steps 1-6 with by creating an RDP connection with the VM *myIISVMWEurope* within the *myResourceGroupTM2* resource group to install IIS and customize its default web page.
10. Launch Windows PowerShell on *myIISVMWEurope* and using the following commands to install IIS server and update the default htm file.

```
# Install IIS
Install-WindowsFeature -name Web-Server -IncludeManagementTools

# Remove default htm file
remove-item C:\inetpub\wwwroot\iisstart.htm

#Add custom htm file
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from my production website
server - " + $env:computername)
```

Configure DNS names for the VMs running IIS

Traffic Manager routes user traffic based on DNS name of the service endpoints. In this section, you configure the DNS names for the IIS servers - *myIISVMEastUS* and *myIISVMWEurope*.

1. Click **All resources** in the left-hand menu, and then from the resources list, select *myIISVMEastUS* that is located in the *myResourceGroupTM1* resource group.
2. On the **Overview** page, under **DNS name**, select **Configure**.
3. On the **Configuration** page, under DNS name label, add a unique name, and then select **Save**.
4. Repeat steps 1-3, for the VM named *myIISVMWEurope* that is located in the *myResourceGroupTM1* resource group.

Create test VMs

In this section, you create a VM (*mVMEastUS* and *myVMWestEurope*) in each Azure region (**East US** and **West Europe**. You will use these VMs to test how Traffic Manager routes traffic to the nearest IIS server when you browse to the website.

1. On the upper, left corner of the Azure portal, select **Create a resource** > **Compute** > **Windows Server 2016 VM**.

2. Enter, or select, the following information for **Basics**, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	myVMEastUS
User name	Enter a user name of your choosing.
Password	Enter a password of your choosing. The password must be at least 12 characters long and meet the defined complexity requirements.
Resource group	Select **Existing** and then select *myResourceGroupTM1*.

3. Select a VM size under **Choose a size**.

4. Select the following values for **Settings**, then select **OK**:

SETTING	VALUE
Virtual network	Select **Virtual network**, in **Create virtual network**, for **Name**, enter *myVNet3*, for subnet, enter *mySubnet3*.

SETTING	VALUE
Network Security Group	Select **Basic**, and in **Select public inbound ports** drop-down, select **HTTP** and **RDP**
Boot diagnostics	Select **Disabled**.

5. Under **Create** in the **Summary**, select **Create** to start the VM deployment.
6. Complete steps 1-5 again, with the following changes:

SETTING	VALUE
VM Name	*myVMWEurope*
Resource group	Select **Existing**, and then type *myResourceGroupTM2*
Virtual network	Select **Virtual network**, in **Create virtual network**, for **Name**, enter *myVNet4*, for subnet, enter *mySubnet4*.

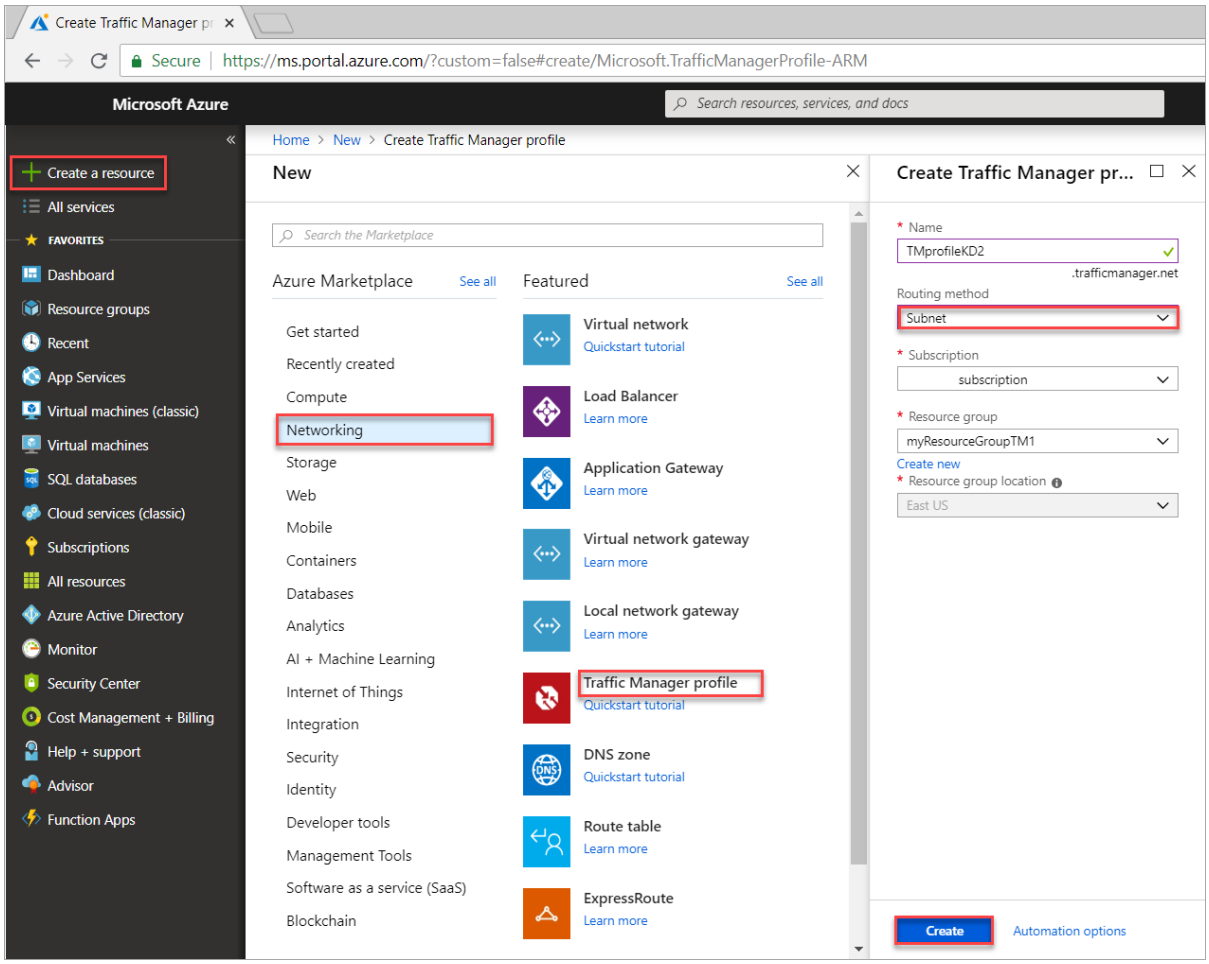
7. The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.

Create a Traffic Manager profile

Create a Traffic Manager profile that allows you to return specific endpoints based on the source IP of the request.

1. On the top left-hand side of the screen, select **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
2. In the **Create Traffic Manager profile**, enter or select, the following information, accept the defaults for the remaining settings, and then select **Create**:

SETTING	VALUE
Name	This name needs to be unique within the trafficmanager.net zone and results in the DNS name, trafficmanager.net that is used to access your Traffic Manager profile.
Routing method	Select the **Subnet** routing method.
Subscription	Select your subscription.
Resource group	Select **Existing** and enter *myResourceGroupTM1*.



Add Traffic Manager endpoints

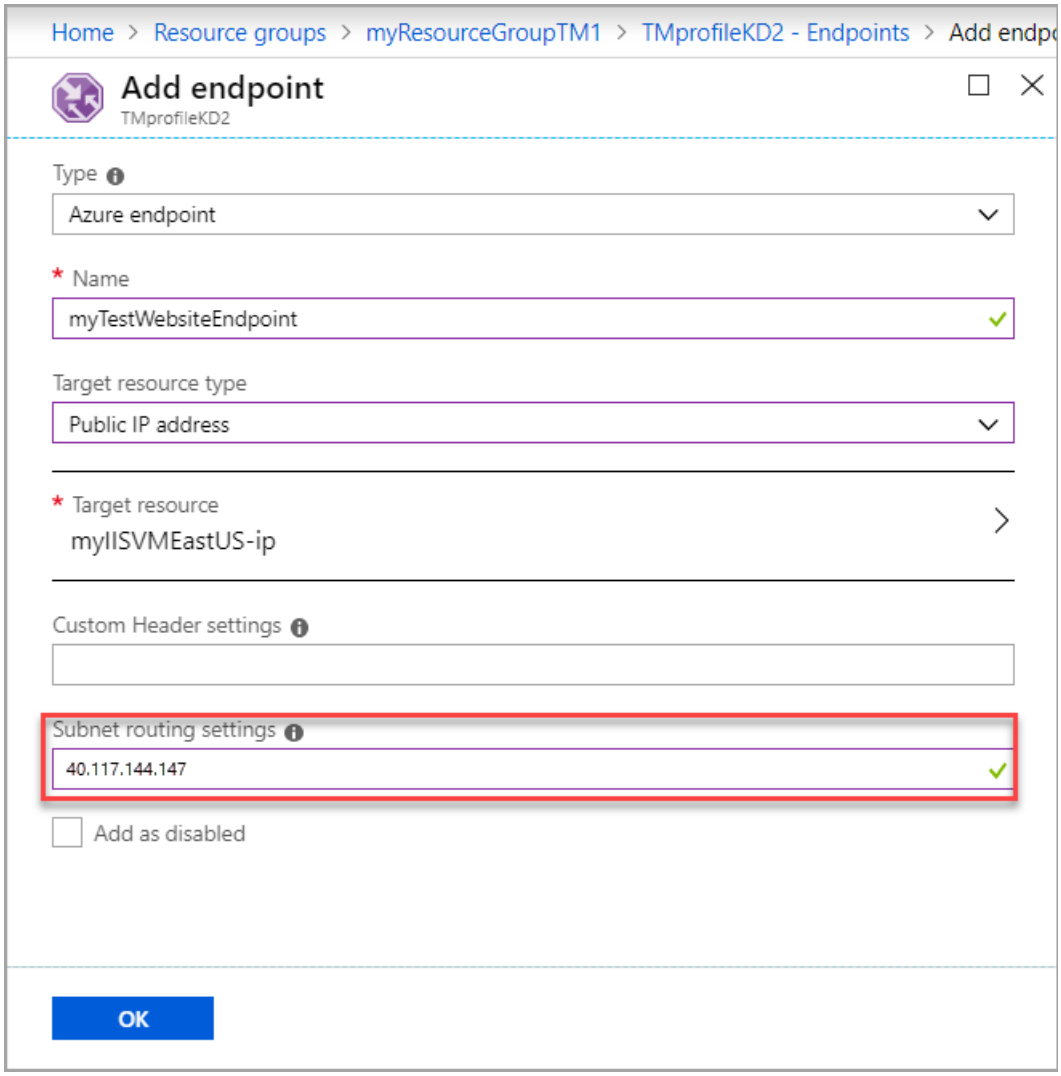
Add the two VMs running the IIS servers - *myIISVMEastUS* & *myIISVMWEurope* to route user traffic based on the subnet of the user's query.

1. In the portal's search bar, search for the Traffic Manager profile name that you created in the preceding section and select the profile in the results that the displayed.
2. In **Traffic Manager profile**, in the **Settings** section, click **Endpoints**, and then click **Add**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **OK**:

SETTING	VALUE
Type	Azure endpoint
Name	myTestWebSiteEndpoint
Target resource type	Public IP Address
Target resource	**Choose a Public IP address** to show the listing of resources with Public IP addresses under the same subscription. In **Resource**, select the public IP address named *myIISVMEastUS-ip*. This is the public IP address of the IIS server VM in East US.
Subnet routing settings	Add the IP address of *myVMEastUS* test VM. Any user query originating from this VM will be directed to the *myTestWebSiteEndpoint*.

4. Repeat steps 2 and 3 to add another endpoint named *myProductionEndpoint* for the public IP address *myIISVMWEurope-ip* that is associated with the IIS server VM named *myIISVMWEurope*. For **Subnet routing settings**, add the IP address of the test VM - *myVMWestEurope*. Any user query from this test VM will be routed to the endpoint - *myProductionWebsiteEndpoint*.

5. When the addition of both endpoints is complete, they are displayed in **Traffic Manager profile** along with their monitoring status as **Online**.



Test Traffic Manager profile

In this section, you test how the Traffic Manager routes user traffic from a given subnet to a specific endpoint. To view the Traffic Manager in action, complete the following steps:

1. Determine the DNS name of your Traffic Manager profile.
2. View Traffic Manager in action as follows:
   - From the test VM (*myVMEastUS*) that is located in the **East US** region, in a web browser, browse to the DNS name of your Traffic Manager profile.
   - From the test VM (*myVMEastUS*) that is located in the **West Europe** region, in a web browser, browse to the DNS name of your Traffic Manager profile.

Determine DNS name of Traffic Manager profile

In this tutorial, for simplicity, you use the DNS name of the Traffic Manager profile to visit the websites.

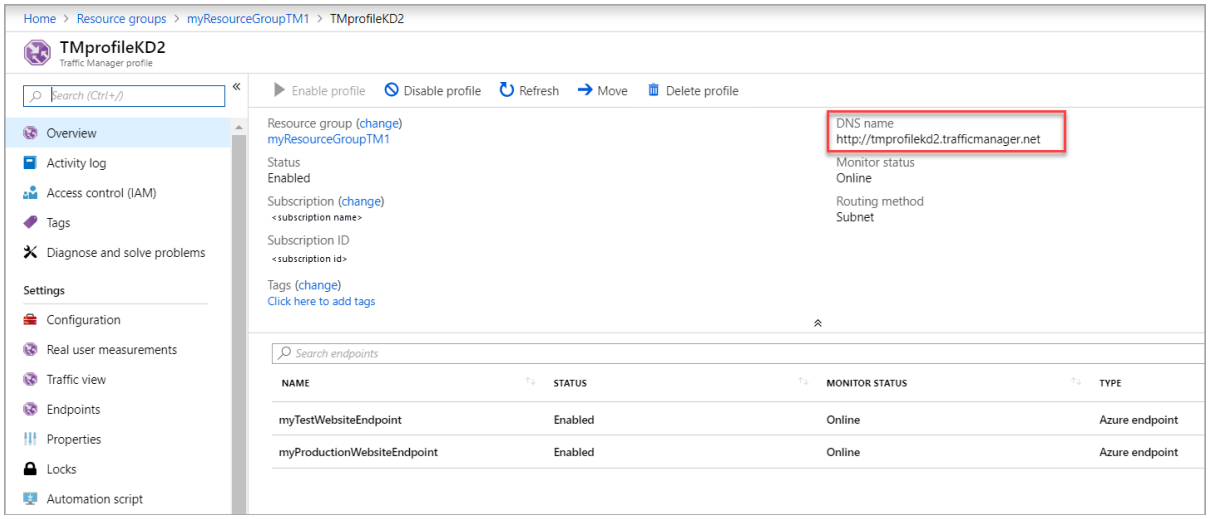
You can determine the DNS name of the Traffic Manager profile as follows:

1. In the portal's search bar, search for the **Traffic Manager profile** name that you created in the preceding

section. In the results that are displayed, click the traffic manager profile.

2. Click **Overview**.

3. The **Traffic Manager profile** displays the DNS name of your newly created Traffic Manager profile. In production deployments, you configure a vanity domain name to point to the Traffic Manager domain name, using a DNS CNAME record.



View Traffic Manager in action

In this section, you can see the Traffic Manager is action.

1. Select **All resources** in the left-hand menu, and then from the resources list click *myVMEastUS* that is located in the *myResourceGroupTM1* resource group.

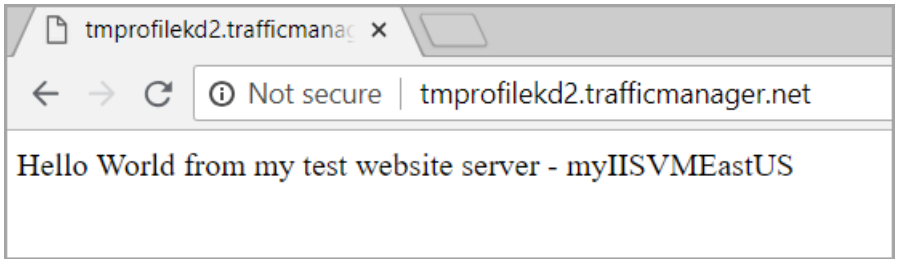
2. On the **Overview** page, click **Connect**, and then in **Connect to virtual machine**, select **Download RDP file**.

3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.

4. Select **OK**.

5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue**, to proceed with the connection.

6. In a web browser on the VM *myVMEastUS*, type the DNS name of your Traffic Manager profile to view your website. Since the VM *myVMEastUS* IP address is associated with the endpoint *myIISVMEastUS*, the web browser launches the Test website server - *myIISVMEastUS*.



7. Next, connect to the VM *myVMWestEurope* located in **West Europe** using steps 1-5 and browse to the Traffic Manager profile domain name from this VM. Since the VM *myVMWestEurope* IP address is associated with the endpoint *myIISVMEastUS*, the web browser launches the Test website server - *myIISVMWEurope*.

Delete the Traffic Manager profile

When no longer needed, delete the resource groups (**ResourceGroupTM1** and **ResourceGroupTM2**). To do so, select the resource group (**ResourceGroupTM1** or **ResourceGroupTM2**), and then select **Delete**.

Next steps

- Learn about weighted traffic routing method.
- Learn about priority routing method.
- Learn about geographic routing method.

How to send Real User Measurements to Traffic Manager with Visual Studio Mobile Center

2/1/2020 • 2 minutes to read • Edit Online

You can set up your mobile application developed using Visual Studio Mobile Center to send Real User Measurements to Traffic Manager by following the steps:

> **NOTE**
>
> Currently, sending Real User Measurements to Traffic manager is only supported for Android.

To configure Real User Measurements, you need to obtain a key and instrument your app with the RUM package.

Step 1: Obtain a key

The measurements you take and sent to Traffic Manager from your client application are identified by the service using a unique string, called the Real User Measurements (RUM) Key. You can get a RUM key using the Azure portal, a REST API or by using the PowerShell / CLI interfaces.

To obtain the RUM Key using Azure portal using the following procedure:

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. In the portal's search bar, search for the Traffic Manager profile name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
3. In the Traffic Manager profile page, click **Real User Measurements** under **Settings**.
4. Click **Generate Key** to create a new RUM Key.

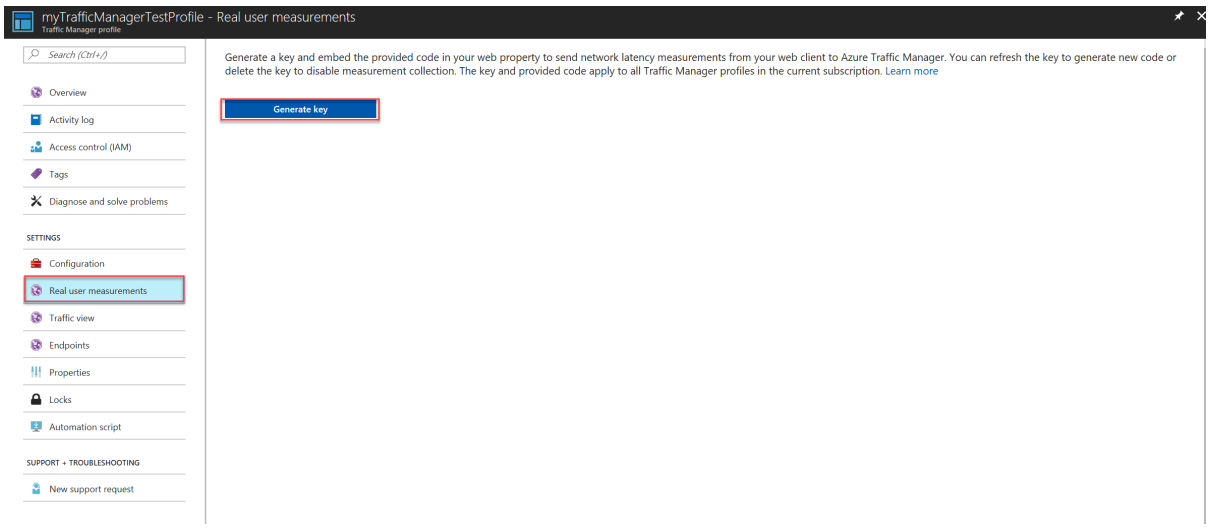


Figure 1: Real User Measurements key generation

5. The page displays the RUM Key that is generated and a JavaScript code snippet that needs to be embedded into your HTML page.

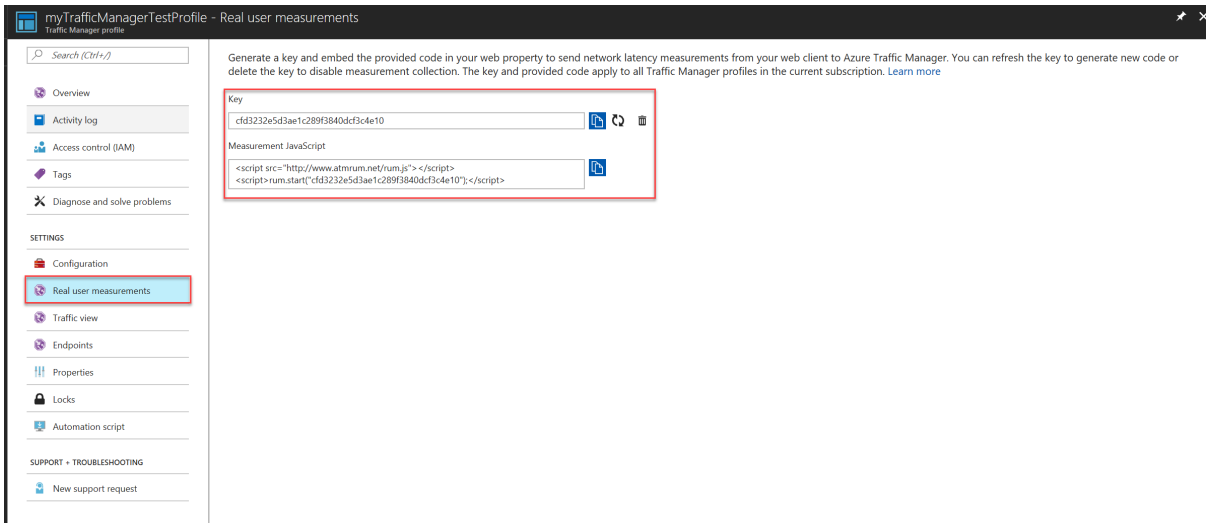


Figure 2: Real User Measurements Key and Measurement JavaScript

6. Click the **Copy** button to copy the RUM Key.

Step 2: Instrument your app with the RUM package of Mobile Center SDK

If you're new to Visual Studio Mobile Center, visit its website. For detailed instructions on SDK integration, see Getting Started with the Android SDK.

To use Real User Measurements, complete the following procedure:

1. Add the SDK to the project

   During the preview of the ATM RUM SDK, you need to explicitly reference the package repository.

   In your **app/build.gradle** file add the following lines:

```
repositories {
    maven {
        url "https://dl.bintray.com/mobile-center/mobile-center-snapshot"
    }
}
```

   In your **app/build.gradle** file add the following lines:

```
dependencies {

    def mobileCenterSdkVersion = '0.12.1-16+3fe5b08'
    compile "com.microsoft.azure.mobile:mobile-center-rum:${mobileCenterSdkVersion}"
}
```

2. Start the SDK

   Open your app's main activity class and add the following import statements:

```
import com.microsoft.azure.mobile.MobileCenter;
import com.microsoft.azure.mobile.rum.RealUserMeasurements;
```

   Look for the `onCreate` callback in the same file and add the following code:

```
RealUserMeasurements.setRumKey("<Your RUM Key>");
MobileCenter.start(getApplication(), "<Your Mobile Center AppSecret>", RealUserMeasurements.class);
```

Next steps

- Learn more about Real User Measurements
- Learn how Traffic Manager works
- Learn more about Mobile Center
- Sign up for Mobile Center
- Learn more about the traffic-routing methods supported by Traffic Manager
- Learn how to create a Traffic Manager profile

How to send Real User Measurements to Azure Traffic Manager using web pages

2/1/2020 • 2 minutes to read • Edit Online

You can configure your web pages to send Real User Measurements to Traffic Manager by obtaining a Real User Measurements (RUM) key and embedding the generated code to web page.

Obtain a Real User Measurements key

The measurements you take and send to Traffic Manager from your client application are identified by the service using a unique string, called the **Real User Measurements (RUM) Key**. You can get a RUM key using the Azure portal, a REST API, or by using the PowerShell or Azure CLI.

To obtain the RUM Key using Azure portal:

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. In the portal's search bar, search for the Traffic Manager profile name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
3. In the Traffic Manager profile blade, click **Real User Measurements** under **Settings**.
4. Click **Generate Key** to create a new RUM Key.

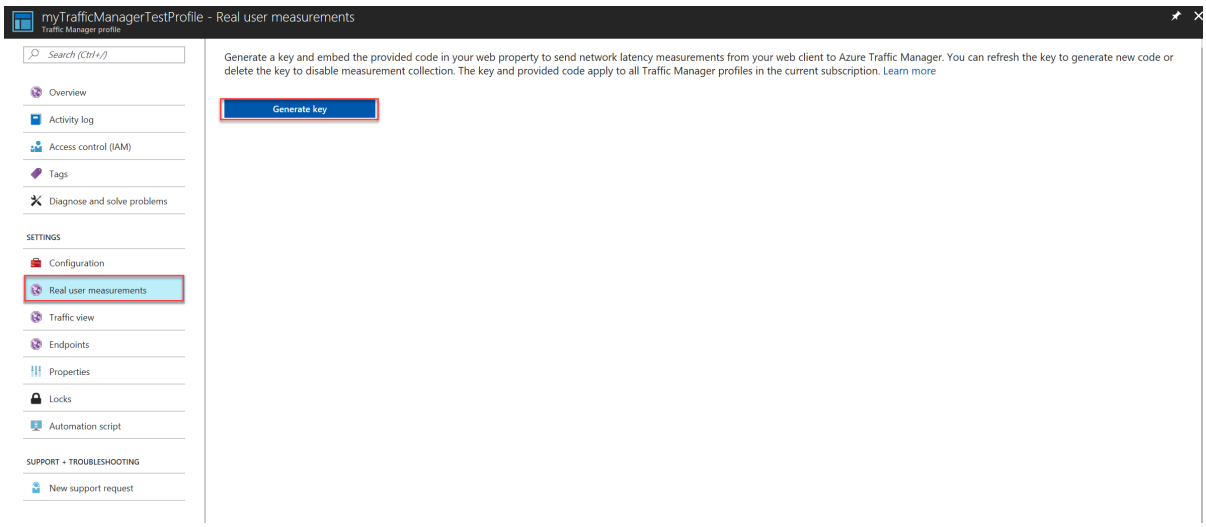


Figure 1: Real User Measurements Key Generation

5. The blade now displays the RUM Key generated and a JavaScript code snippet that needs to be embedded into your HTML page.

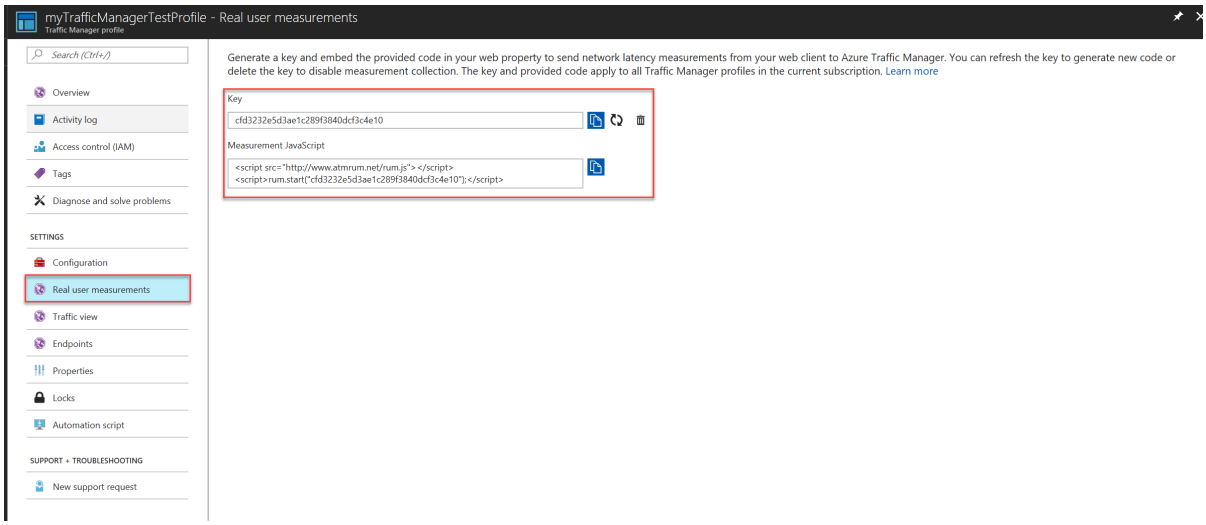


Figure 2: Real User Measurements Key and Measurement JavaScript

6. Click the **Copy** button to copy the JavaScript code.

> **IMPORTANT**
>
> Use the generated JavaScript for Real User Measurements feature to function properly. Any changes to this script or the scripts used by Real User Measurements can lead to unpredictable behavior.

Embed the code to an HTML web page

After you have obtained the RUM key, the next step is to embed this copied JavaScript into an HTML page that your end users visit. Editing HTML can be done in many ways and using different tools and workflows. This example shows how to update an HTML page to add this script. You can use this guidance to adapt it to your HTML source management workflow.

1. Open the HTML page in a text editor
2. Paste the JavaScript code you had copied in the earlier step to the BODY section of the HTML (the copied code is on line 8 & 9, see figure 3).

```
1  <HTML>
2  <HEAD>
3  <TITLE>Webpage powered by Azure</TITLE>
4  </HEAD>
5  <BODY BGCOLOR="FFFFFF">
6  <H1>Welcome</H1>
7  <P> <B>Hello!</B>
8  <script src="//www.atmrum.net/rum.js"></script>
9  <script>rum.start("0123456789abcdef0123456789abcdff");</script>
10 </BODY>
11 </HTML>
```

Figure 3: Simple HTML with embedded Real User Measurements JavaScript

3. Save the HTML file and host it on a webserver connected to the internet.
4. Next time this page is rendered on a web browser, the JavaScript referenced is downloaded and the script will execute the measurement and reporting operations.

Next steps

- Learn more about Real User Measurements
- Learn how Traffic Manager works
- Learn more about the traffic-routing methods supported by Traffic Manager
- Learn how to create a Traffic Manager profile

Add, disable, enable, or delete endpoints

2/1/2020 • 3 minutes to read • Edit Online

The Web Apps feature in Azure App Service already provides failover and round-robin traffic routing functionality for websites within a datacenter, regardless of the website mode. Azure Traffic Manager allows you to specify failover and round-robin traffic routing for websites and cloud services in different datacenters. The first step necessary to provide that functionality is to add the cloud service or website endpoint to Traffic Manager.

You can also disable individual endpoints that are part of a Traffic Manager profile. Disabling an endpoint leaves it as part of the profile, but the profile acts as if the endpoint is not included in it. This action is useful for temporarily removing an endpoint that is in maintenance mode or being redeployed. Once the endpoint is up and running again, it can be enabled.

> **NOTE**
> Disabling an endpoint has nothing to do with its deployment state in Azure. A healthy endpoint remains up and able to receive traffic even when disabled in Traffic Manager. Additionally, disabling an endpoint in one profile does not affect its status in another profile.

To add a cloud service or an App service endpoint to a Traffic Manager profile

1. From a browser, sign in to the Azure portal.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
3. In the **Traffic Manager profile** blade, in the **Settings** section, click **Endpoints**.
4. In the **Endpoints** blade that is displayed, click **Add**.
5. In the **Add endpoint** blade, complete as follows:
   a. For **Type**, click **Azure endpoint**.
   b. Provide a **Name** by which you want to recognize this endpoint.
   c. For **Target resource type**, from the drop-down, choose the appropriate resource type.
   d. For **Target resource**, click the **Choose...** selector to list resources under the same subscription in the **Resources blade**. In the **Resource** blade that is displayed, pick the service that you want to add as the first endpoint.
   e. For **Priority**, select as **1**. This results in all traffic going to this endpoint if it is healthy.
   f. Keep **Add as disabled** unchecked.
   g. Click **OK**
6. Repeat steps 4 and 5 to add the next Azure endpoint. Make sure to add it with its **Priority** value set at **2**.
7. When the addition of both endpoints is complete, they are displayed in the **Traffic Manager profile** blade along with their monitoring status as **Online**.

> **NOTE**
> After you add or remove an endpoint from a profile using the *Failover* traffic routing method, the failover priority list may not be ordered the way you want. You can adjust the order of the Failover Priority List on the Configuration page. For more information, see Configure Failover traffic routing.

To disable an endpoint

1. From a browser, sign in to the Azure portal.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that are displayed.
3. In the **Traffic Manager profile** blade, in the **Settings** section, click **Endpoints**.
4. Click the endpoint that you want to disable.
5. In the **Endpoint** blade, change the endpoint status to **Disabled**, and then click **Save**.
6. Clients continue to send traffic to the endpoint for the duration of Time-to-Live (TTL). You can change the TTL on the Configuration page of the Traffic Manager profile.

To enable an endpoint

1. From a browser, sign in to the Azure portal.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that are displayed.
3. In the **Traffic Manager profile** blade, in the **Settings** section, click **Endpoints**.
4. Click the endpoint that you want to enable.
5. In the **Endpoint** blade, change the endpoint status to **Enabled**, and then click **Save**.
6. Clients continue to send traffic to the endpoint for the duration of Time-to-Live (TTL). You can change the TTL on the Configuration page of the Traffic Manager profile.

To delete an endpoint

1. From a browser, sign in to the Azure portal.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that are displayed.
3. In the **Traffic Manager profile** blade, in the **Settings** section, click **Endpoints**.
4. Click the endpoint that you want to delete.
5. In the **Endpoint** blade, click **Delete**

Next steps

- Manage Traffic Manager profiles
- Configure routing methods
- Troubleshooting Traffic Manager degraded state
- Traffic Manager performance considerations
- Operations on Traffic Manager (REST API Reference)

Manage an Azure Traffic Manager profile

2/1/2020 • 3 minutes to read • Edit Online

Traffic Manager profiles use traffic-routing methods to control the distribution of traffic to your cloud services or website endpoints. This article explains how to create and manage these profiles.

Create a Traffic Manager profile

You can create a Traffic Manager profile by using the Azure portal. After creating your profile, you can configure endpoints, monitoring, and other settings in the Azure portal. Traffic Manager supports up to 200 endpoints per profile. However, most usage scenarios require only a few of endpoints.

To create a Traffic Manager profile

1. From a browser, sign in to the Azure portal. If you don't already have an account, you can sign up for a free one-month trial.
2. Click **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
3. In the **Create Traffic Manager profile**, complete as follows:
   a. In **Name**, provide a name for your profile. This name needs to be unique within the trafficmanager.net zone and results in the DNS name `<name>`, trafficmanager.net, that is used to access your Traffic Manager profile.
   b. In **Routing method**, select the **Priority** routing method.
   c. In **Subscription**, select the subscription you want to create this profile under
   d. In **Resource Group**, create a new resource group to place this profile under.
   e. In **Resource group location**, select the location of the resource group. This setting refers to the location of the resource group, and has no impact on the Traffic Manager profile that will be deployed globally.
   f. Click **Create**.
   g. When the global deployment of your Traffic Manager profile is complete, it is listed in respective resource group as one of the resources.

Disable, enable, or delete a profile

You can disable an existing profile so that Traffic Manager does not refer user requests to the configured endpoints. When you disable a Traffic Manager profile, the profile and the information contained in the profile remain intact and can be edited in the Traffic Manager interface. Referrals resume when you re-enable the profile. When you create a Traffic Manager profile in the Azure portal, it's automatically enabled. If you decide a profile is no longer necessary, you can delete it.

To disable a profile

1. If you are using a custom domain name, change the CNAME record on your Internet DNS server so that it no longer points to your Traffic Manager profile.
2. Traffic stops being directed to the endpoints through the Traffic Manager profile settings.
3. From a browser, sign in to the Azure portal.
4. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
5. Click **Overview** > **Disable**.
6. Confirm to disable the Traffic Manager profile.

To enable a profile

1. From a browser, sign in to the Azure portal.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
3. Click **Overview** > **Enable**.
4. If you are using a custom domain name, create a CNAME resource record on your Internet DNS server to point to the domain name of your Traffic Manager profile.
5. Traffic is directed to the endpoints again.

To delete a profile

1. Ensure that the DNS resource record on your Internet DNS server no longer uses a CNAME resource record that points to the domain name of your Traffic Manager profile.
2. In the portal's search bar, search for the **Traffic Manager profile** name that you want to modify, and then click the Traffic Manager profile in the results that the displayed.
3. Click **Overview** > **Delete**.
4. Confirm to delete the Traffic Manager profile.

Next steps

- Add an endpoint
- Configure Priority routing method
- Configure Geographic routing method
- Configure Weighted routing method
- Configure Performance routing method

Verify Traffic Manager settings

2/1/2020 • 3 minutes to read • Edit Online

To test your Traffic Manager settings, you need to have multiple clients, in various locations, from which you can run your tests. Then, bring the endpoints in your Traffic Manager profile down one at a time.

- Set the DNS TTL value low so that changes propagate quickly (for example, 30 seconds).
- Know the IP addresses of your Azure cloud services and websites in the profile you are testing.
- Use tools that let you resolve a DNS name to an IP address and display that address.

You are checking to see that the DNS names resolve to IP addresses of the endpoints in your profile. The names should resolve in a manner consistent with the traffic routing method defined in the Traffic Manager profile. You can use the tools like **nslookup** or **dig** to resolve DNS names.

The following examples help you test your Traffic Manager profile.

Check Traffic Manager profile using nslookup and ipconfig in Windows

1. Open a command or Windows PowerShell prompt as an administrator.
2. Type `ipconfig /flushdns` to flush the DNS resolver cache.
3. Type `nslookup <your Traffic Manager domain name>`. For example, the following command checks the domain name with the prefix *myapp.contoso*

   ```
   nslookup myapp.contoso.trafficmanager.net
   ```

   A typical result shows the following information:

   - The DNS name and IP address of the DNS server being accessed to resolve this Traffic Manager domain name.
   - The Traffic Manager domain name you typed on the command line after "nslookup" and the IP address to which the Traffic Manager domain resolves. The second IP address is the important one to check. It should match a public virtual IP (VIP) address for one of the cloud services or websites in the Traffic Manager profile you are testing.

How to test the failover traffic routing method

1. Leave all endpoints up.
2. Using a single client, request DNS resolution for your company domain name using nslookup or a similar utility.
3. Ensure that the resolved IP address matches the primary endpoint.
4. Bring down your primary endpoint or remove the monitoring file so that Traffic Manager thinks that the application is down.
5. Wait for the DNS Time-to-Live (TTL) of the Traffic Manager profile plus an additional two minutes. For example, if your DNS TTL is 300 seconds (5 minutes), you must wait for seven minutes.
6. Flush your DNS client cache and request DNS resolution using nslookup. In Windows, you can flush your DNS cache with the ipconfig /flushdns command.
7. Ensure that the resolved IP address matches your secondary endpoint.
8. Repeat the process, bringing down each endpoint in turn. Verify that the DNS returns the IP address of the next endpoint in the list. When all endpoints are down, you should obtain the IP address of the primary

endpoint again.

How to test the weighted traffic routing method

1. Leave all endpoints up.
2. Using a single client, request DNS resolution for your company domain name using nslookup or a similar utility.
3. Ensure that the resolved IP address matches one of your endpoints.
4. Flush your DNS client cache and repeat steps 2 and 3 for each endpoint. You should see different IP addresses returned for each of your endpoints.

How to test the performance traffic routing method

To effectively test a performance traffic routing method, you must have clients located in different parts of the world. You can create clients in different Azure regions that can be used to test your services. If you have a global network, you can remotely sign in to clients in other parts of the world and run your tests from there.

Alternatively, there are free web-based DNS lookup and dig services available. Some of these tools give you the ability to check DNS name resolution from various locations around the world. Do a search on "DNS lookup" for examples. Third-party services like Gomez or Keynote can be used to confirm that your profiles are distributing traffic as expected.

Next steps

- About Traffic Manager traffic routing methods
- Traffic Manager performance considerations
- Troubleshooting Traffic Manager degraded state

Using load-balancing services in Azure

2/1/2020 • 10 minutes to read • Edit Online

Introduction

Microsoft Azure provides multiple services for managing how network traffic is distributed and load balanced. You can use these services individually or combine their methods, depending on your needs, to build the optimal solution.

In this tutorial, we first define a customer use case and see how it can be made more robust and performant by using the following Azure load-balancing portfolio: Traffic Manager, Application Gateway, and Load Balancer. We then provide step-by-step instructions for creating a deployment that is geographically redundant, distributes traffic to VMs, and helps you manage different types of requests.

At a conceptual level, each of these services plays a distinct role in the load-balancing hierarchy.

- **Traffic Manager** provides global DNS load balancing. It looks at incoming DNS requests and responds with a healthy endpoint, in accordance with the routing policy the customer has selected. Options for routing methods are:
  - Performance routing to send the requestor to the closest endpoint in terms of latency.
  - Priority routing to direct all traffic to an endpoint, with other endpoints as backup.
  - Weighted round-robin routing, which distributes traffic based on the weighting that is assigned to each endpoint.
  - Geography-based routing to distribute the traffic to your application endpoints based on geographic location of the user.
  - Subnet-based routing to distribute the traffic to your application endpoints based on the subnet (IP address range) of the user.
  - Multi Value routing that enable you to send IP addresses of more than one application endpoints in a single DNS response.

  The client connects directly to the endpoint returned by Traffic Manager. Azure Traffic Manager detects when an endpoint is unhealthy and then redirects the clients to another healthy instance. Refer to Azure Traffic Manager documentation to learn more about the service.
- **Application Gateway** provides application delivery controller (ADC) as a service, offering various Layer 7 load-balancing capabilities for your application. It allows customers to optimize web farm productivity by offloading CPU-intensive SSL termination to the application gateway. Other Layer 7 routing capabilities include round-robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single application gateway. Application Gateway can be configured as an Internet-facing gateway, an internal-only gateway, or a combination of both. Application Gateway is fully Azure managed, scalable, and highly available. It provides a rich set of diagnostics and logging capabilities for better manageability.
- **Load Balancer** is an integral part of the Azure SDN stack, providing high-performance, low-latency Layer 4 load-balancing services for all UDP and TCP protocols. It manages inbound and outbound connections. You can configure public and internal load-balanced endpoints and define rules to map inbound connections to back-end pool destinations by using TCP and HTTP health-probing options to manage service availability.

Scenario

In this example scenario, we use a simple website that serves two types of content: images and dynamically

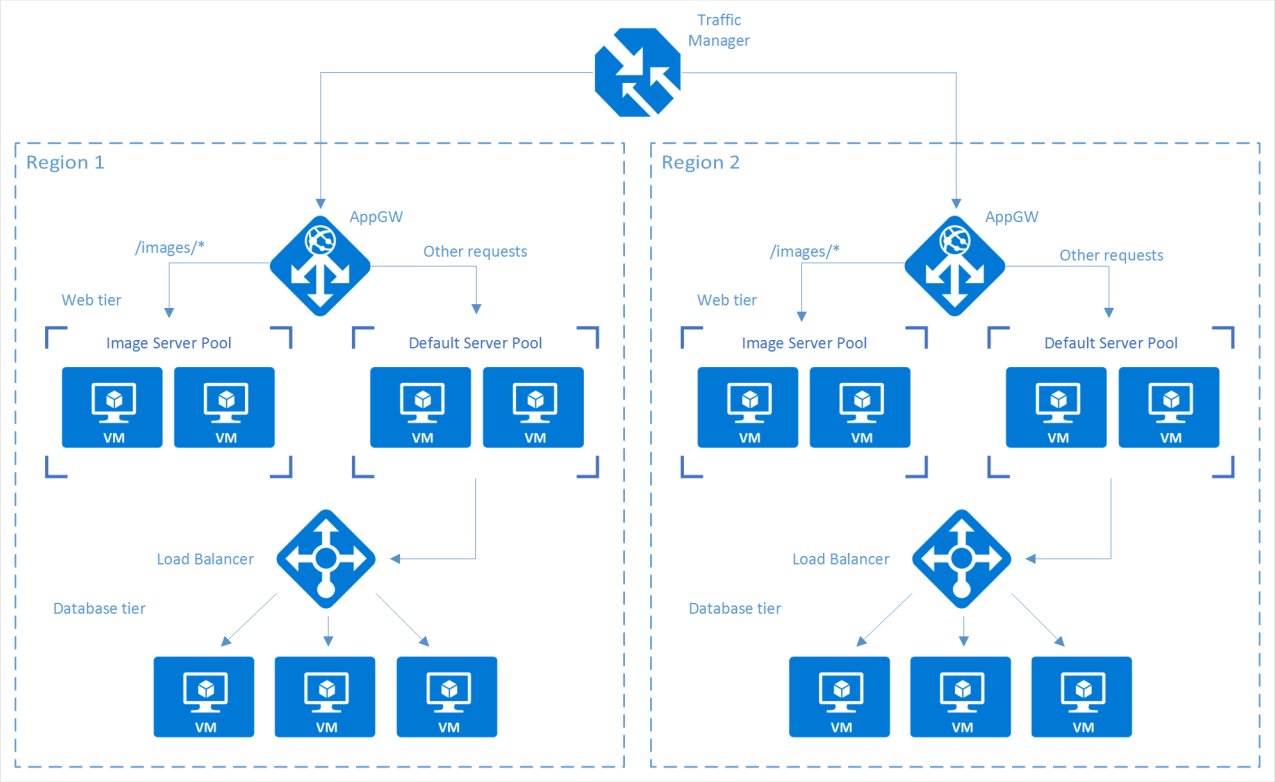
rendered webpages. The website must be geographically redundant, and it should serve its users from the closest (lowest latency) location to them. The application developer has decided that any URLs that match the pattern /images/* are served from a dedicated pool of VMs that are different from the rest of the web farm.

Additionally, the default VM pool serving the dynamic content needs to talk to a back-end database that is hosted on a high-availability cluster. The entire deployment is set up through Azure Resource Manager.

Using Traffic Manager, Application Gateway, and Load Balancer allows this website to achieve these design goals:

- **Multi-geo redundancy**: If one region goes down, Traffic Manager routes traffic seamlessly to the closest region without any intervention from the application owner.
- **Reduced latency**: Because Traffic Manager automatically directs the customer to the closest region, the customer experiences lower latency when requesting the webpage contents.
- **Independent scalability**: Because the web application workload is separated by type of content, the application owner can scale the request workloads independent of each other. Application Gateway ensures that the traffic is routed to the right pools based on the specified rules and the health of the application.
- **Internal load balancing**: Because Load Balancer is in front of the high-availability cluster, only the active and healthy endpoint for a database is exposed to the application. Additionally, a database administrator can optimize the workload by distributing active and passive replicas across the cluster independent of the front-end application. Load Balancer delivers connections to the high-availability cluster and ensures that only healthy databases receive connection requests.

The following diagram shows the architecture of this scenario:



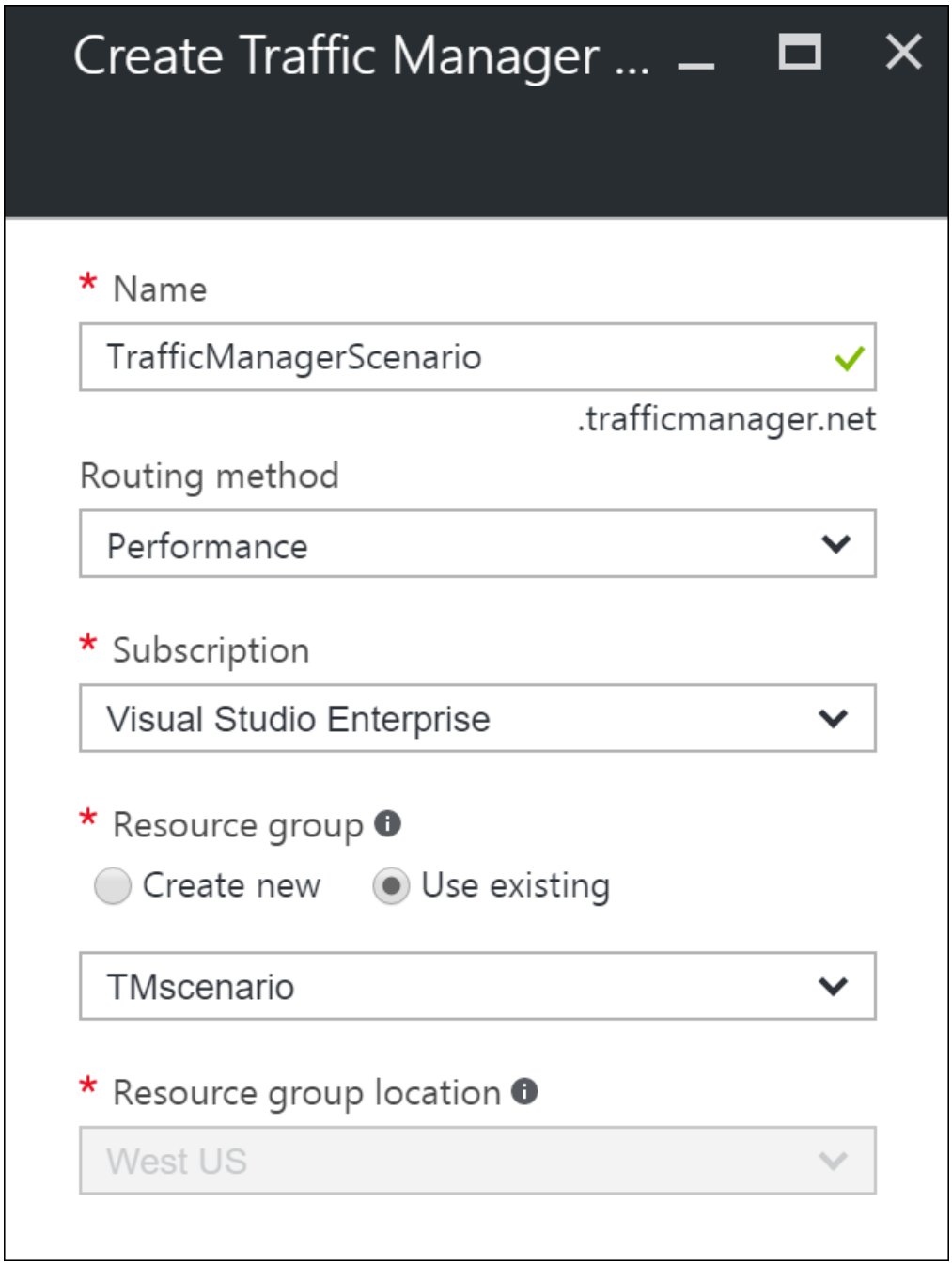
NOTE

This example is only one of many possible configurations of the load-balancing services that Azure offers. Traffic Manager, Application Gateway, and Load Balancer can be mixed and matched to best suit your load-balancing needs. For example, if SSL offload or Layer 7 processing is not necessary, Load Balancer can be used in place of Application Gateway.

Setting up the load-balancing stack

Step 1: Create a Traffic Manager profile

1. In the Azure portal, click **Create a resource** > **Networking** > **Traffic Manager profile** > **Create**.
2. Enter the following basic information:
   - **Name**: Give your Traffic Manager profile a DNS prefix name.
   - **Routing method**: Select the traffic-routing method policy. For more information about the methods, see About Traffic Manager traffic routing methods.
   - **Subscription**: Select the subscription that contains the profile.
   - **Resource group**: Select the resource group that contains the profile. It can be a new or existing resource group.
   - **Resource group location**: Traffic Manager service is global and not bound to a location. However, you must specify a region for the group where the metadata associated with the Traffic Manager profile resides. This location has no impact on the runtime availability of the profile.
3. Click **Create** to generate the Traffic Manager profile.



Step 2: Create the application gateways

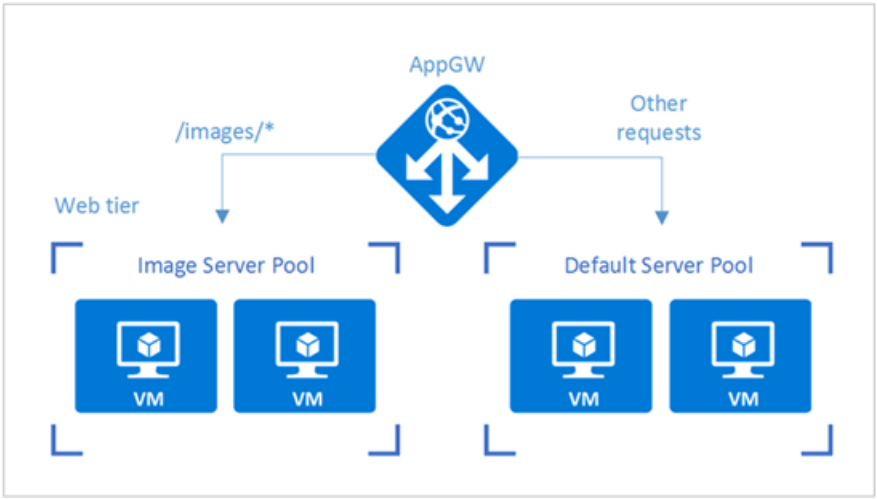
1. In the Azure portal, in the left pane, click **Create a resource** > **Networking** > **Application Gateway**.
2. Enter the following basic information about the application gateway:

- **Name**: The name of the application gateway.
- **SKU size**: The size of the application gateway, available as Small, Medium, or Large.
- **Instance count**: The number of instances, a value from 2 through 10.
- **Resource group**: The resource group that holds the application gateway. It can be an existing resource group or a new one.
- **Location**: The region for the application gateway, which is the same location as the resource group. The location is important, because the virtual network and public IP must be in the same location as the gateway.

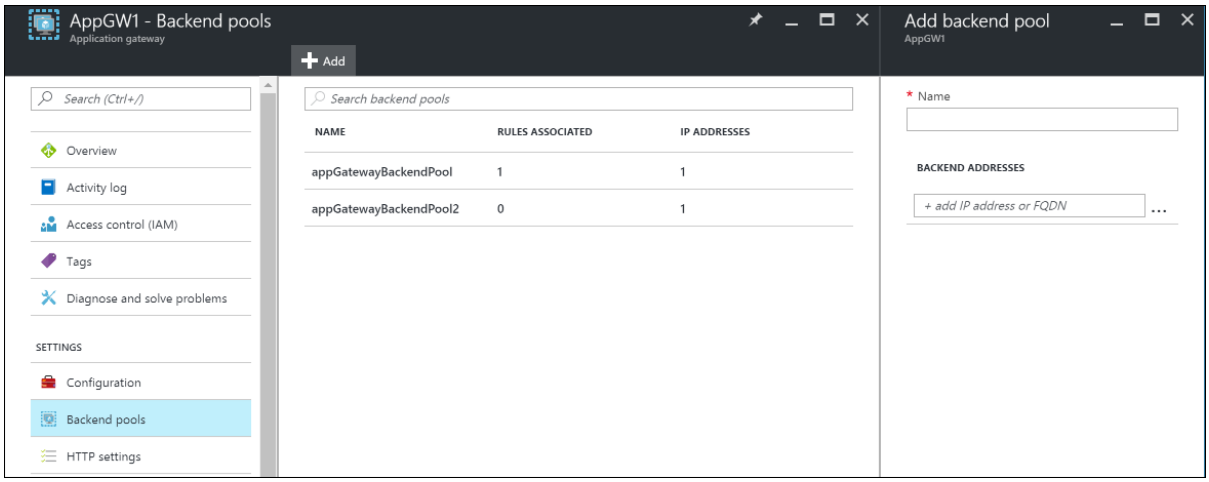
3. Click **OK**.
4. Define the virtual network, subnet, front-end IP, and listener configurations for the application gateway. In this scenario, the front-end IP address is **Public**, which allows it to be added as an endpoint to the Traffic Manager profile later on.
5. Configure the listener with one of the following options:
   - If you use HTTP, there is nothing to configure. Click **OK**.
   - If you use HTTPS, further configuration is required. Refer to Create an application gateway, starting at step 9. When you have completed the configuration, click **OK**.

Configure URL routing for application gateways

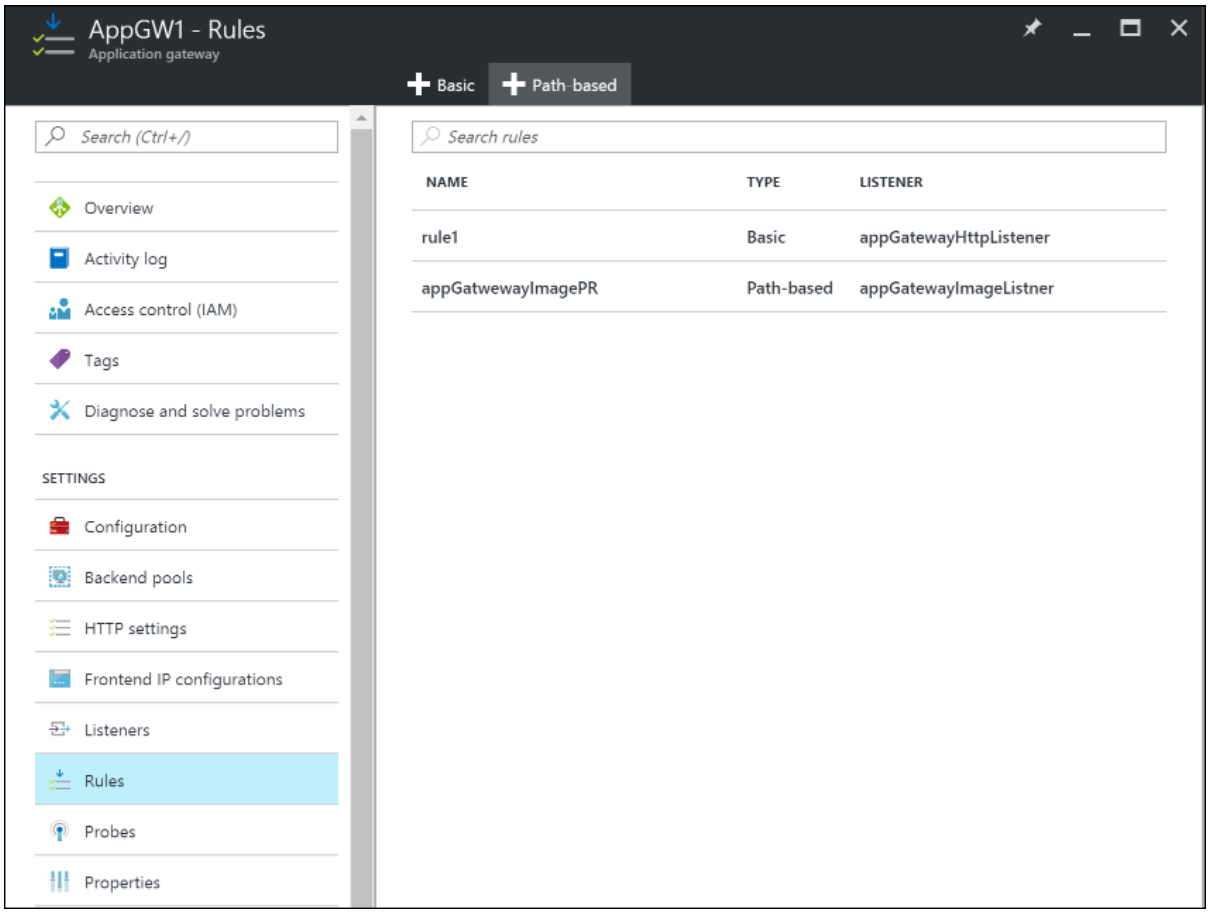
When you choose a back-end pool, an application gateway that's configured with a path-based rule takes a path pattern of the request URL in addition to round-robin distribution. In this scenario, we are adding a path-based rule to direct any URL with "/images/*" to the image server pool. For more information about configuring URL path-based routing for an application gateway, refer to Create a path-based rule for an application gateway.



1. From your resource group, go to the instance of the application gateway that you created in the preceding section.
2. Under **Settings**, select **Backend pools**, and then select **Add** to add the VMs that you want to associate with the web-tier back-end pools.
3. Enter the name of the back-end pool and all the IP addresses of the machines that reside in the pool. In this scenario, we are connecting two back-end server pools of virtual machines.



4. Under **Settings** of the application gateway, select **Rules**, and then click the **Path based** button to add a rule.



5. Configure the rule by providing the following information.

   Basic settings:

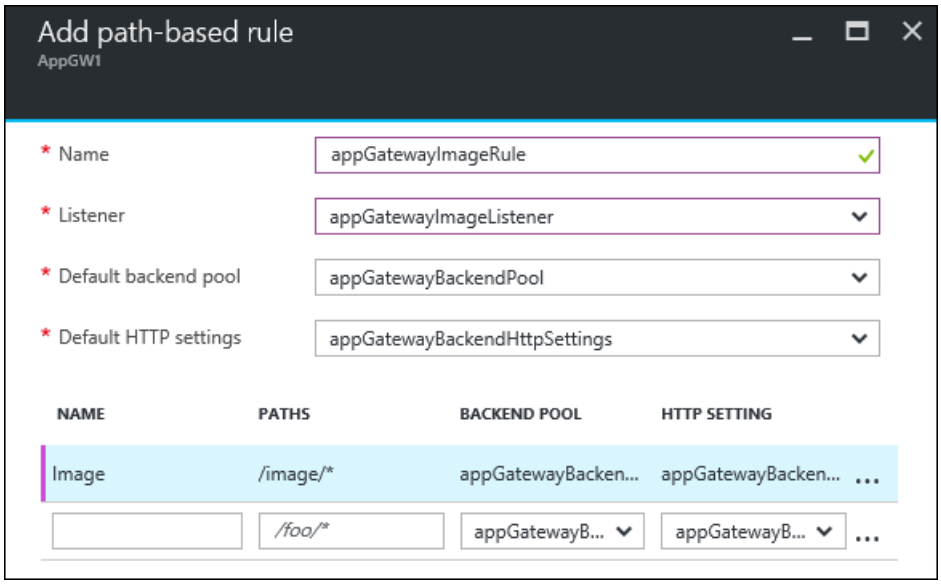
   - **Name**: The friendly name of the rule that is accessible in the portal.
   - **Listener**: The listener that is used for the rule.
   - **Default backend pool**: The back-end pool to be used with the default rule.
   - **Default HTTP settings**: The HTTP settings to be used with the default rule.

   Path-based rules:

   - **Name**: The friendly name of the path-based rule.
   - **Paths**: The path rule that is used for forwarding traffic.
   - **Backend Pool**: The back-end pool to be used with this rule.
   - **HTTP Setting**: The HTTP settings to be used with this rule.

IMPORTANT

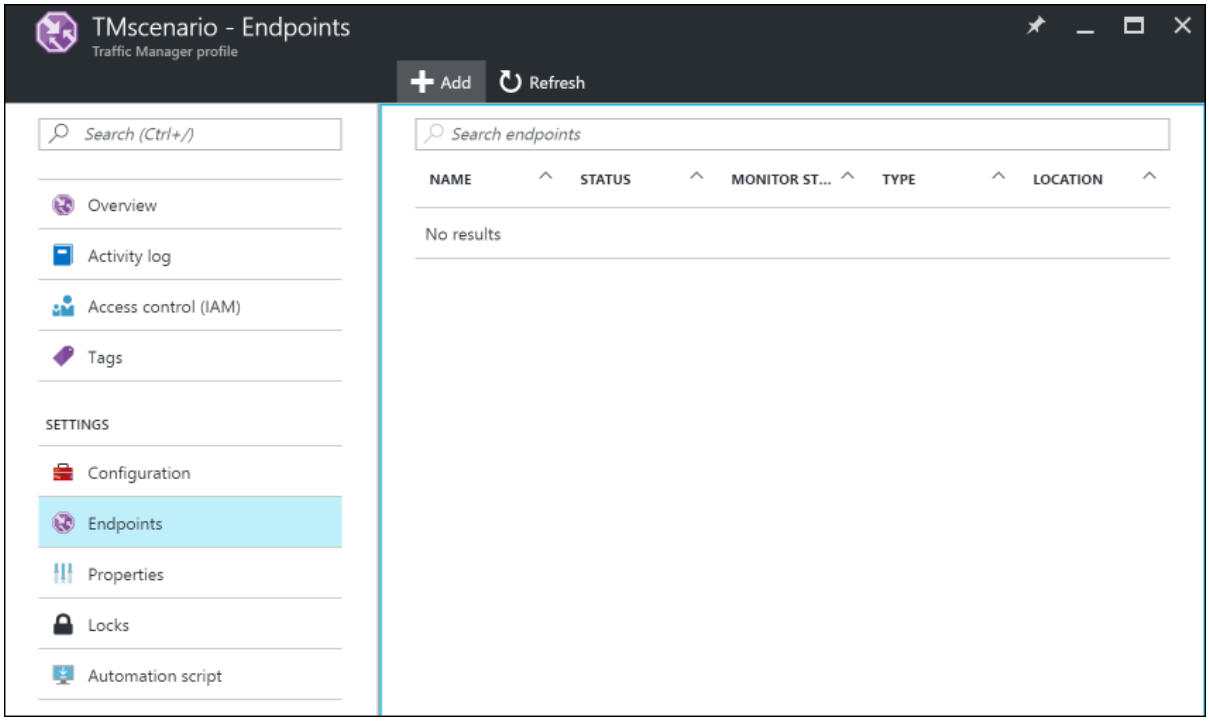
Paths: Valid paths must start with "/". The wildcard "*" is allowed only at the end. Valid examples are /xyz, /xyz*, or /xyz/*.



Step 3: Add application gateways to the Traffic Manager endpoints

In this scenario, Traffic Manager is connected to application gateways (as configured in the preceding steps) that reside in different regions. Now that the application gateways are configured, the next step is to connect them to your Traffic Manager profile.

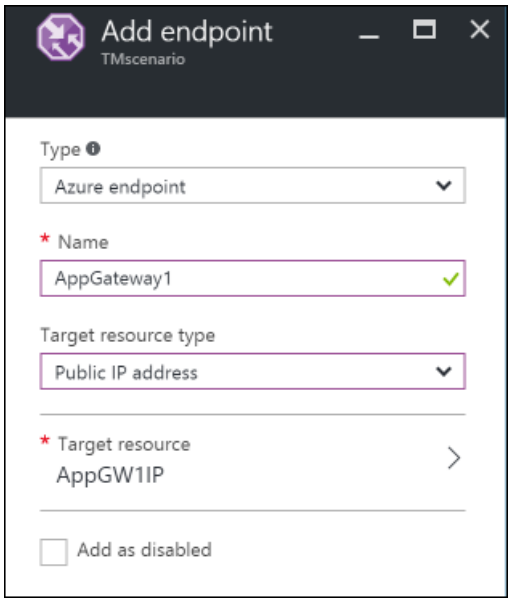
1. Open your Traffic Manager profile. To do so, look in your resource group or search for the name of the Traffic Manager profile from **All Resources**.
2. In the left pane, select **Endpoints**, and then click **Add** to add an endpoint.



3. Create an endpoint by entering the following information:

- **Type**: Select the type of endpoint to load-balance. In this scenario, select **Azure endpoint** because we are connecting it to the application gateway instances that were configured previously.

- **Name**: Enter the name of the endpoint.
- **Target resource type**: Select **Public IP address** and then, under **Target resource**, select the public IP of the application gateway that was configured previously.



4. Now you can test your setup by accessing it with the DNS of your Traffic Manager profile (in this example: TrafficManagerScenario.trafficmanager.net). You can resend requests, bring up or bring down VMs and web servers that were created in different regions, and change the Traffic Manager profile settings to test your setup.

Step 4: Create a load balancer

In this scenario, Load Balancer distributes connections from the web tier to the databases within a high-availability cluster.

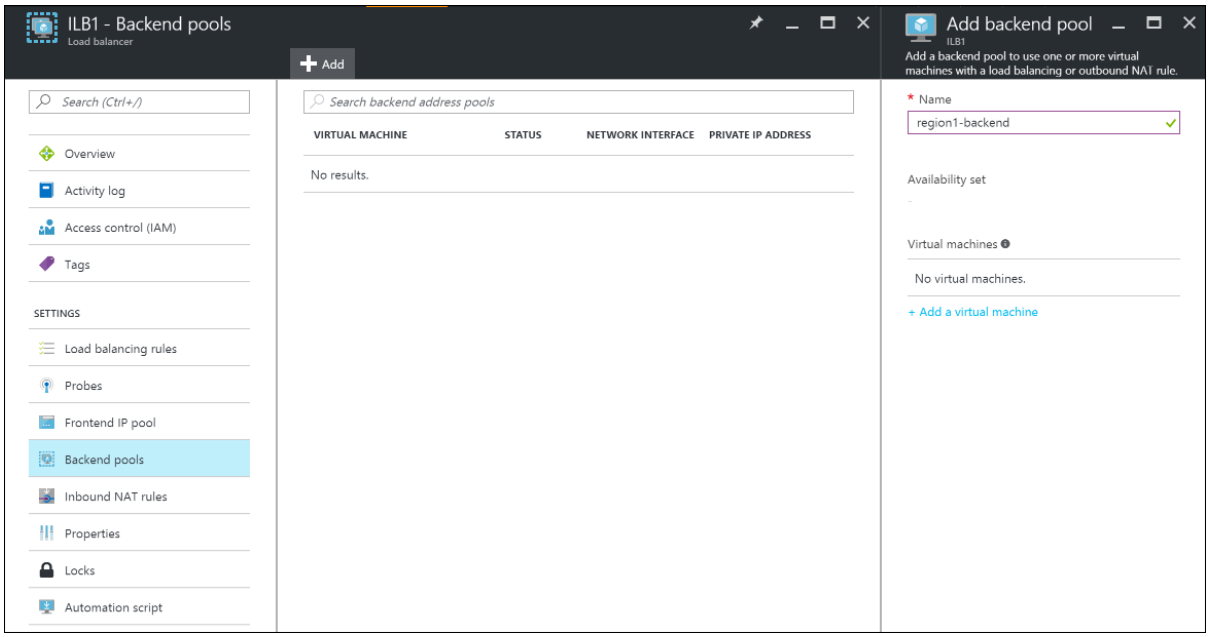
If your high-availability database cluster is using SQL Server AlwaysOn, refer to Configure one or more Always On Availability Group Listeners for step-by-step instructions.

For more information about configuring an internal load balancer, see Create an Internal load balancer in the Azure portal.

1. In the Azure portal, in the left pane, click **Create a resource** > **Networking** > **Load balancer**.
2. Choose a name for your load balancer.
3. Set the **Type** to **Internal**, and choose the appropriate virtual network and subnet for the load balancer to reside in.
4. Under **IP address assignment**, select either **Dynamic** or **Static**.
5. Under **Resource group**, choose the resource group for the load balancer.
6. Under **Location**, choose the appropriate region for the load balancer.
7. Click **Create** to generate the load balancer.

Connect a back-end database tier to the load balancer

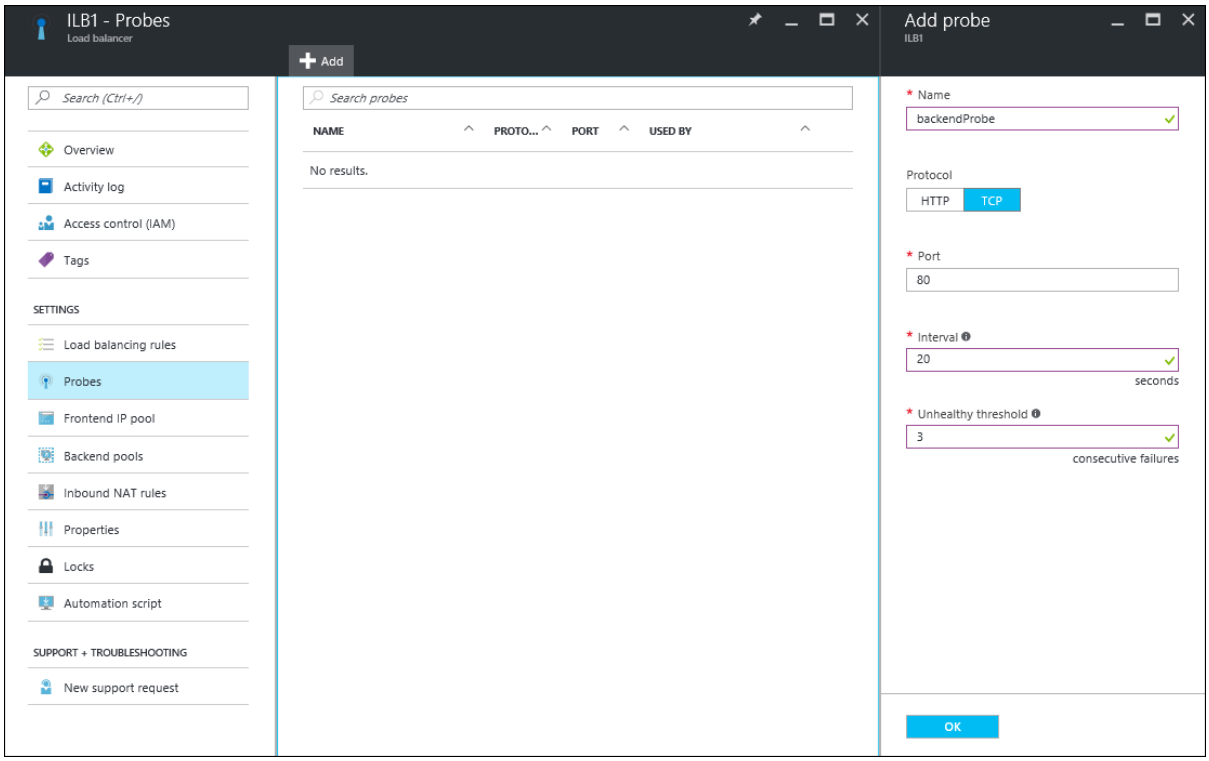
1. From your resource group, find the load balancer that was created in the previous steps.
2. Under **Settings**, click **Backend pools**, and then click **Add** to add a back-end pool.



3. Enter the name of the back-end pool.
4. Add either individual machines or an availability set to the back-end pool.

Configure a probe

1. In your load balancer, under **Settings**, select **Probes**, and then click **Add** to add a probe.



2. Enter the name for the probe.
3. Select the **Protocol** for the probe. For a database, you might want a TCP probe rather than an HTTP probe. To learn more about load-balancer probes, refer to Understand load balancer probes.
4. Enter the **Port** of your database to be used for accessing the probe.
5. Under **Interval**, specify how frequently to probe the application.
6. Under **Unhealthy threshold**, specify the number of continuous probe failures that must occur for the back-end VM to be considered unhealthy.

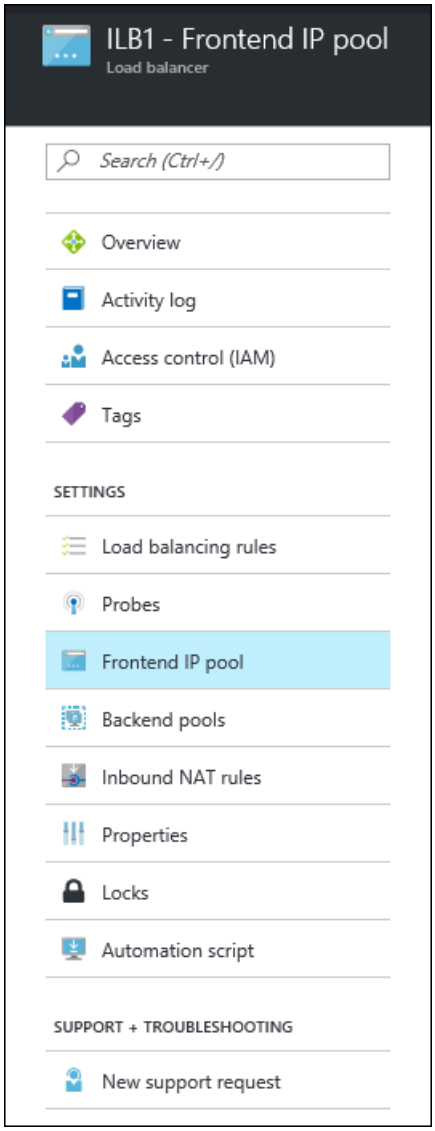
7. Click **OK** to create the probe.

Configure the load-balancing rules

1. Under **Settings** of your load balancer, select **Load balancing rules**, and then click **Add** to create a rule.
2. Enter the **Name** for the load-balancing rule.
3. Choose the **Frontend IP Address** of the load balancer, **Protocol**, and **Port**.
4. Under **Backend port**, specify the port to be used in the back-end pool.
5. Select the **Backend pool** and the **Probe** that were created in the previous steps to apply the rule to.
6. Under **Session persistence**, choose how you want the sessions to persist.
7. Under **Idle timeouts**, specify the number of minutes before an idle timeout.
8. Under **Floating IP**, select either **Disabled** or **Enabled**.
9. Click **OK** to create the rule.

Step 5: Connect web-tier VMs to the load balancer

Now we configure the IP address and load-balancer front-end port in the applications that are running on your web-tier VMs for any database connections. This configuration is specific to the applications that run on these VMs. To configure the destination IP address and port, refer to the application documentation. To find the IP address of the front end, in the Azure portal, go to the front-end IP pool on the **Load balancer settings**.



Next steps

- Overview of Traffic Manager
- Application Gateway overview

- Azure Load Balancer overview

Performance considerations for Traffic Manager

2/1/2020 • 3 minutes to read • Edit Online

This page explains performance considerations using Traffic Manager. Consider the following scenario:

You have instances of your website in the WestUS and EastAsia regions. One of the instances is failing the health check for the traffic manager probe. Application traffic is directed to the healthy region. This failover is expected but performance can be a problem based on the latency of the traffic now traveling to a distant region.

Performance considerations for Traffic Manager

The only performance impact that Traffic Manager can have on your website is the initial DNS lookup. A DNS request for the name of your Traffic Manager profile is handled by the Microsoft DNS root server that hosts the trafficmanager.net zone. Traffic Manager populates, and regularly updates, the Microsoft's DNS root servers based on the Traffic Manager policy and the probe results. So even during the initial DNS lookup, no DNS queries are sent to Traffic Manager.

Traffic Manager is made up of several components: DNS name servers, an API service, the storage layer, and an endpoint monitoring service. If a Traffic Manager service component fails, there is no effect on the DNS name associated with your Traffic Manager profile. The records in the Microsoft DNS servers remain unchanged. However, endpoint monitoring and DNS updating do not happen. Therefore, Traffic Manager is not able to update DNS to point to your failover site when your primary site goes down.

DNS name resolution is fast and results are cached. The speed of the initial DNS lookup depends on the DNS servers the client uses for name resolution. Typically, a client can complete a DNS lookup within ~50 ms. The results of the lookup are cached for the duration of the DNS Time-to-live (TTL). The default TTL for Traffic Manager is 300 seconds.

Traffic does NOT flow through Traffic Manager. Once the DNS lookup completes, the client has an IP address for an instance of your web site. The client connects directly to that address and does not pass through Traffic Manager. The Traffic Manager policy you choose has no influence on the DNS performance. However, a Performance routing-method can negatively impact the application experience. For example, if your policy redirects traffic from North America to an instance hosted in Asia, the network latency for those sessions may be a performance issue.

Measuring Traffic Manager Performance

There are several websites you can use to understand the performance and behavior of a Traffic Manager profile. Many of these sites are free but may have limitations. Some sites offer enhanced monitoring and reporting for a fee.

The tools on these sites measure DNS latencies and display the resolved IP addresses for client locations around the world. Most of these tools do not cache the DNS results. Therefore, the tools show the full DNS lookup each time a test is run. When you test from your own client, you only experience the full DNS lookup performance once during the TTL duration.

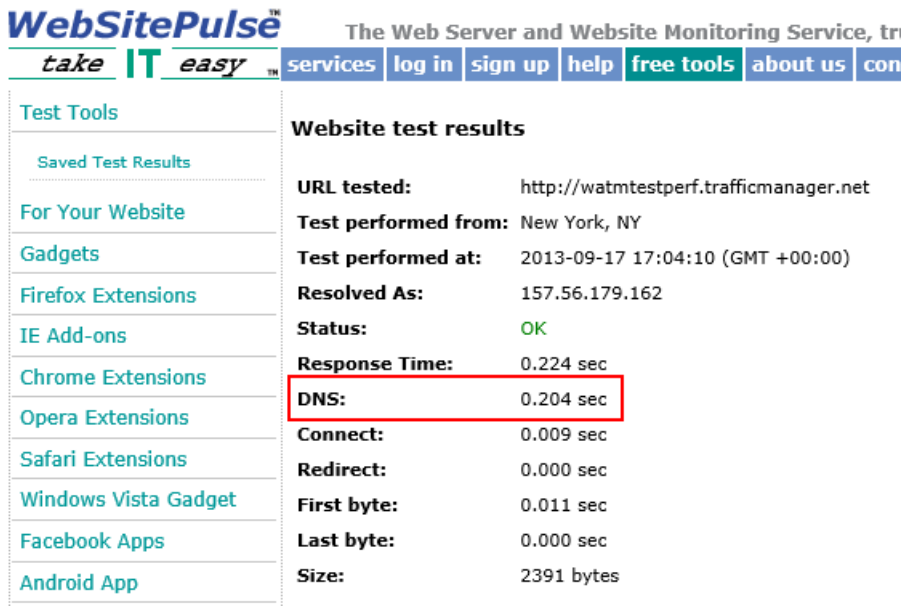
Sample tools to measure DNS performance

- SolveDNS

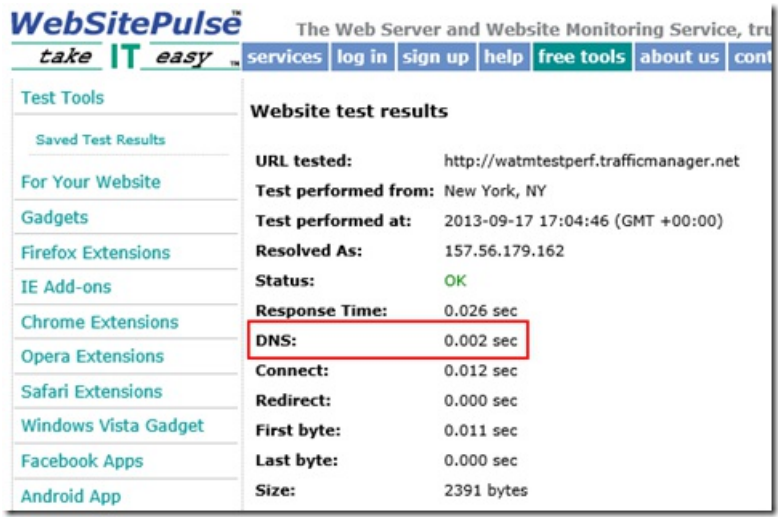
  SolveDNS offers many performance tools. The DNS Comparison tool can show you how long it takes to resolve your DNS name and how that compares to other DNS service providers.

- WebSitePulse

  One of the simplest tools is WebSitePulse. Enter the URL to see DNS resolution time, First Byte, Last Byte, and other performance statistics. You can choose from three different test locations. In this example, you see that the first execution shows that DNS lookup takes 0.204 sec.

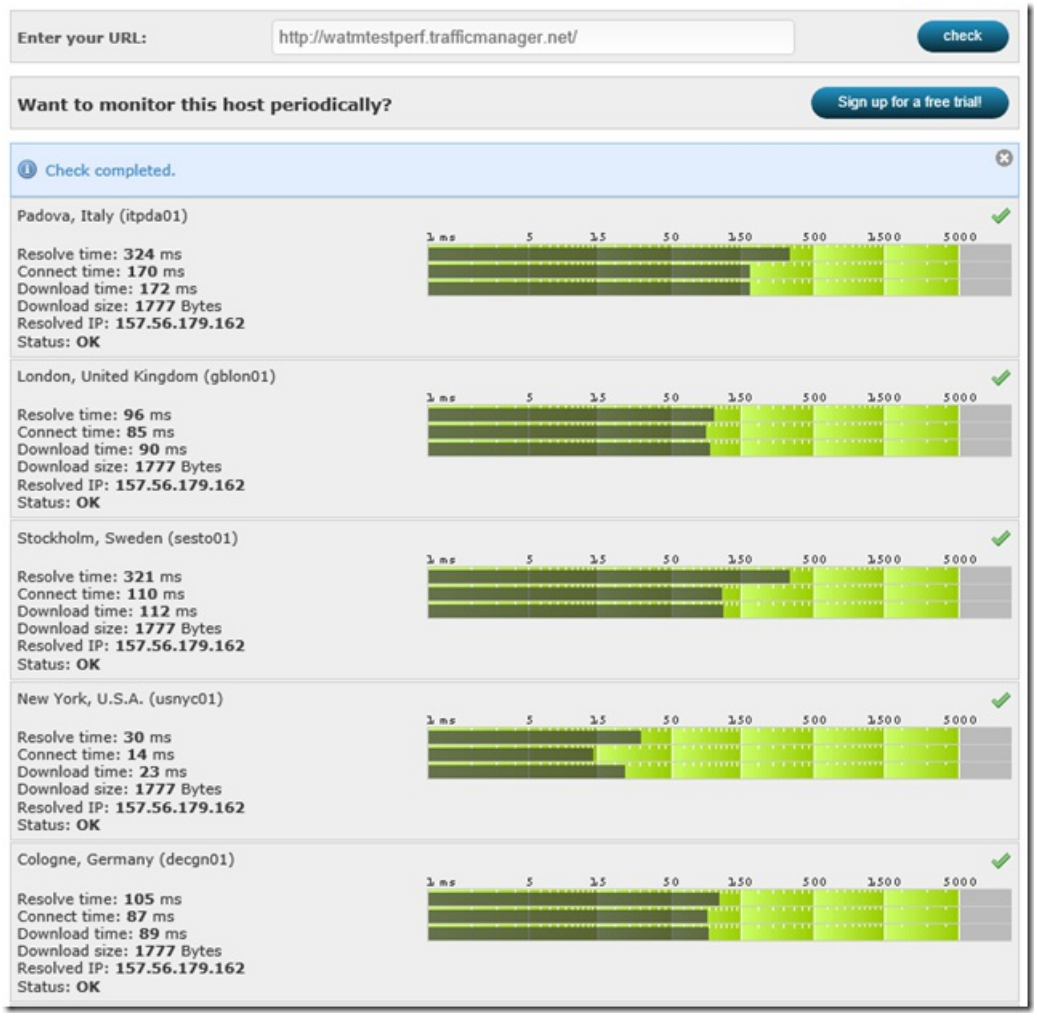


  Because the results are cached, the second test for the same Traffic Manager endpoint the DNS lookup takes 0.002 sec.



- CA App Synthetic Monitor

  Formerly known as the Watch-mouse Check Website tool, this site show you the DNS resolution time from multiple geographic regions simultaneously. Enter the URL to see DNS resolution time, connection time, and speed from several geographic locations. Use this test to see which hosted service is returned for different locations around the world.



- Pingdom

  This tool provides performance statistics for each element of a web page. The Page Analysis tab shows the percentage of time spent on DNS lookup.

- What's My DNS?

  This site does a DNS lookup from 20 different locations and displays the results on a map.

- Dig Web Interface

  This site shows more detailed DNS information including CNAMEs and A records. Make sure you check the 'Colorize output' and 'Stats' under options, and select 'All' under Nameservers.

Next Steps

About Traffic Manager traffic routing methods

Test your Traffic Manager settings

Operations on Traffic Manager (REST API Reference)

Azure Traffic Manager Cmdlets

Enable diagnostic logging in Azure Traffic Manager

2/1/2020 • 2 minutes to read • Edit Online

This article describes how to enable diagnostic logging and access log data for a Traffic Manager profile.

Azure Traffic Manager diagnostic logs can provide insight into the behavior of the Traffic Manager profile resource. For example, you can use the profile's log data to determine why individual probes have timed out against an endpoint.

Enable diagnostic logging

> **NOTE**
>
> This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

You can run the commands that follow in the Azure Cloud Shell, or by running PowerShell from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use with your account. If you run PowerShell from your computer, you need the Azure PowerShell module, 1.0.0 or later. You can run `Get-Module -ListAvailable Az` to find the installed version. If you need to install or upgrade, see Install Azure PowerShell module. If you are running PowerShell locally, you also need to run `Login-AzAccount` to sign in to Azure.

1. **Retrieve the Traffic Manager profile:**

   To enable diagnostic logging, you need the ID of a Traffic Manager profile. Retrieve the Traffic Manager profile that you want to enable diagnostic logging for with Get-AzTrafficManagerProfile. The output includes the Traffic Manager profile's ID information.

   ```
   Get-AzTrafficManagerProfile -Name <TrafficManagerprofilename> -ResourceGroupName <resourcegroupname>
   ```

2. **Enable diagnostic logging for the Traffic Manager profile:**

   Enable diagnostic logging for the Traffic Manager profile using the ID obtained in the previous step with Set-AzDiagnosticSetting. The following command stores verbose logs for the Traffic Manager profile to a specified Azure Storage account.

   ```
   Set-AzDiagnosticSetting -ResourceId <TrafficManagerprofileResourceId> -StorageAccountId
   <storageAccountId> -Enabled $true
   ```

3. **Verify diagnostic settings:**

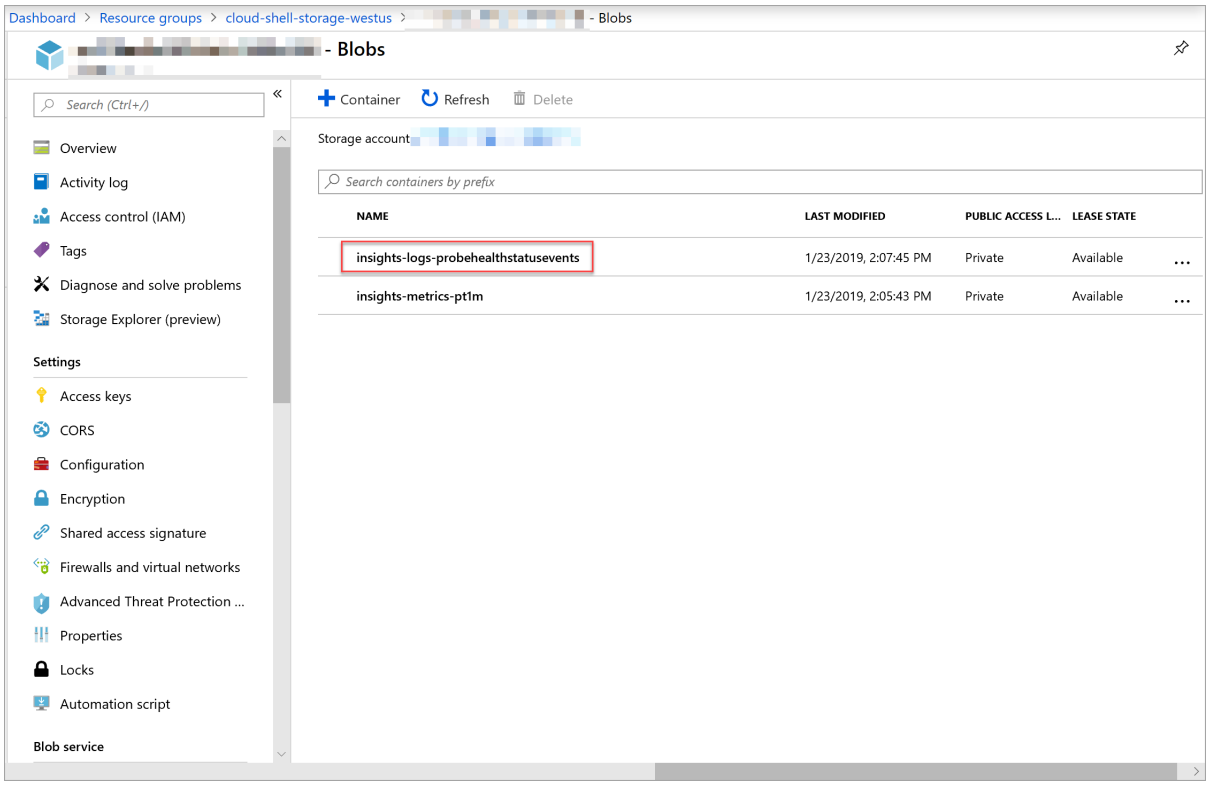
   Verify diagnostic settings for the Traffic Manager profile using Get-AzDiagnosticSetting. The following command displays the categories that are logged for a resource.

   ```
   Get-AzDiagnosticSetting -ResourceId <TrafficManagerprofileResourceId>
   ```

Ensure that all log categories associated with the Traffic Manager profile resource display as enabled. Also, verify that the storage account is correctly set.

Access log files

1. Sign in to the Azure portal.
2. Navigate to your Azure Storage account in the portal.
3. On the **Overview** page of your Azure storage account, under **Services** select **Blobs**.
4. For **Containers**, select **insights-logs-probehealthstatusevents**, and navigate down to the PT1H.json file and click **Download** to download and save a copy of this log file.



Traffic Manager log schema

All diagnostic logs available through Azure Monitor share a common top-level schema, with flexibility for each service to emit unique properties for their own events. For top-level diagnostic logs schema, see Supported services, schemas, and categories for Azure Diagnostic Logs.

The following table includes logs schema specific to the Azure Traffic Manager profile resource.

Field Name	Field Type	Definition	Example
EndpointName	String	The name of the Traffic Manager endpoint whose health status is being recorded.	*myPrimaryEndpoint*
Status	String	The health status of the Traffic Manager endpoint that was probed. The status can either be **Up** or **Down**.	**Up**

Next steps

- Learn more about Traffic Manager Monitoring

Using PowerShell to manage Traffic Manager

2/1/2020 • 11 minutes to read • Edit Online

Azure Resource Manager is the preferred management interface for services in Azure. Azure Traffic Manager profiles can be managed using Azure Resource Manager-based APIs and tools.

> **NOTE**
> This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

Resource model

Azure Traffic Manager is configured using a collection of settings called a Traffic Manager profile. This profile contains DNS settings, traffic routing settings, endpoint monitoring settings, and a list of service endpoints to which traffic is routed.

Each Traffic Manager profile is represented by a resource of type 'TrafficManagerProfiles'. At the REST API level, the URI for each profile is as follows:

```
https://management.azure.com/subscriptions/{subscription-id}/resourceGroups/{resource-group-name}/providers/Microsoft.Network/trafficManagerProfiles/{profile-name}?api-version={api-version}
```

Setting up Azure PowerShell

> **NOTE**
> This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

These instructions use Microsoft Azure PowerShell. The following article explains how to install and configure Azure PowerShell.

- How to install and configure Azure PowerShell

The examples in this article assume that you have an existing resource group. You can create a resource group using the following command:

```
New-AzResourceGroup -Name MyRG -Location "West US"
```

NOTE

Azure Resource Manager requires that all resource groups have a location. This location is used as the default for resources created in that resource group. However, since Traffic Manager profile resources are global, not regional, the choice of resource group location has no impact on Azure Traffic Manager.

Create a Traffic Manager Profile

To create a Traffic Manager profile, use the `New-AzTrafficManagerProfile` cmdlet:

```
$TmProfile = New-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG -TrafficRoutingMethod
Performance -RelativeDnsName contoso -Ttl 30 -MonitorProtocol HTTP -MonitorPort 80 -MonitorPath "/"
```

The following table describes the parameters:

PARAMETER	DESCRIPTION
Name	The resource name for the Traffic Manager profile resource. Profiles in the same resource group must have unique names. This name is separate from the DNS name used for DNS queries.
ResourceGroupName	The name of the resource group containing the profile resource.
TrafficRoutingMethod	Specifies the traffic-routing method used to determine which endpoint is returned in response a DNS query. Possible values are 'Performance', 'Weighted' or 'Priority'.
RelativeDnsName	Specifies the hostname portion of the DNS name provided by this Traffic Manager profile. This value is combined with the DNS domain name used by Azure Traffic Manager to form the fully qualified domain name (FQDN) of the profile. For example, setting the value of 'contoso' becomes 'contoso.trafficmanager.net.'
TTL	Specifies the DNS Time-to-Live (TTL), in seconds. This TTL informs the Local DNS resolvers and DNS clients how long to cache DNS responses for this Traffic Manager profile.
MonitorProtocol	Specifies the protocol to use to monitor endpoint health. Possible values are 'HTTP' and 'HTTPS'.
MonitorPort	Specifies the TCP port used to monitor endpoint health.
MonitorPath	Specifies the path relative to the endpoint domain name used to probe for endpoint health.

The cmdlet creates a Traffic Manager profile in Azure and returns a corresponding profile object to PowerShell. At this point, the profile does not contain any endpoints. For more information about adding endpoints to a Traffic Manager profile, see Adding Traffic Manager Endpoints.

Get a Traffic Manager Profile

To retrieve an existing Traffic Manager profile object, use the `Get-AzTrafficManagerProfile` cmdlet:

```
$TmProfile = Get-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG
```

This cmdlet returns a Traffic Manager profile object.

Update a Traffic Manager Profile

Modifying Traffic Manager profiles follows a 3-step process:

1. Retrieve the profile using `Get-AzTrafficManagerProfile` or use the profile returned by `New-AzTrafficManagerProfile`.
2. Modify the profile. You can add and remove endpoints or change endpoint or profile parameters. These changes are off-line operations. You are only changing the local object in memory that represents the profile.
3. Commit your changes using the `Set-AzTrafficManagerProfile` cmdlet.

All profile properties can be changed except the profile's RelativeDnsName. To change the RelativeDnsName, you must delete profile and a new profile with a new name.

The following example demonstrates how to change the profile's TTL:

```
$TmProfile = Get-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG
$TmProfile.Ttl = 300
Set-AzTrafficManagerProfile -TrafficManagerProfile $TmProfile
```

There are three types of Traffic Manager endpoints:

1. **Azure endpoints** are services hosted in Azure
2. **External endpoints** are services hosted outside of Azure
3. **Nested endpoints** are used to construct nested hierarchies of Traffic Manager profiles. Nested endpoints enable advanced traffic-routing configurations for complex applications.

In all three cases, endpoints can be added in two ways:

1. Using a 3-step process described previously. The advantage of this method is that several endpoint changes can be made in a single update.
2. Using the New-AzTrafficManagerEndpoint cmdlet. This cmdlet adds an endpoint to an existing Traffic Manager profile in a single operation.

Adding Azure Endpoints

Azure endpoints reference services hosted in Azure. Two types of Azure endpoints are supported:

1. Azure App Service
2. Azure PublicIpAddress resources (which can be attached to a load-balancer or a virtual machine NIC). The PublicIpAddress must have a DNS name assigned to be used in Traffic Manager.

In each case:

- The service is specified using the 'targetResourceId' parameter of `Add-AzTrafficManagerEndpointConfig` or `New-AzTrafficManagerEndpoint`.
- The 'Target' and 'EndpointLocation' are implied by the TargetResourceId.
- Specifying the 'Weight' is optional. Weights are only used if the profile is configured to use the 'Weighted' traffic-routing method. Otherwise, they are ignored. If specified, the value must be a number between 1 and 1000. The default value is '1'.
- Specifying the 'Priority' is optional. Priorities are only used if the profile is configured to use the 'Priority' traffic-

routing method. Otherwise, they are ignored. Valid values are from 1 to 1000 with lower values indicating a higher priority. If specified for one endpoint, they must be specified for all endpoints. If omitted, default values starting from '1' are applied in the order that the endpoints are listed.

Example 1: Adding App Service endpoints using `Add-AzTrafficManagerEndpointConfig`

In this example, we create a Traffic Manager profile and add two App Service endpoints using the `Add-AzTrafficManagerEndpointConfig` cmdlet.

```
$TmProfile = New-AzTrafficManagerProfile -Name myprofile -ResourceGroupName MyRG -TrafficRoutingMethod
Performance -RelativeDnsName myapp -Ttl 30 -MonitorProtocol HTTP -MonitorPort 80 -MonitorPath "/"
$webapp1 = Get-AzWebApp -Name webapp1
Add-AzTrafficManagerEndpointConfig -EndpointName webapp1ep -TrafficManagerProfile $TmProfile -Type
AzureEndpoints -TargetResourceId $webapp1.Id -EndpointStatus Enabled
$webapp2 = Get-AzWebApp -Name webapp2
Add-AzTrafficManagerEndpointConfig -EndpointName webapp2ep -TrafficManagerProfile $TmProfile -Type
AzureEndpoints -TargetResourceId $webapp2.Id -EndpointStatus Enabled
Set-AzTrafficManagerProfile -TrafficManagerProfile $TmProfile
```

Example 2: Adding a publicIpAddress endpoint using `New-AzTrafficManagerEndpoint`

In this example, a public IP address resource is added to the Traffic Manager profile. The public IP address must have a DNS name configured, and can be bound either to the NIC of a VM or to a load balancer.

```
$ip = Get-AzPublicIpAddress -Name MyPublicIP -ResourceGroupName MyRG
New-AzTrafficManagerEndpoint -Name MyIpEndpoint -ProfileName MyProfile -ResourceGroupName MyRG -Type
AzureEndpoints -TargetResourceId $ip.Id -EndpointStatus Enabled
```

Adding External Endpoints

Traffic Manager uses external endpoints to direct traffic to services hosted outside of Azure. As with Azure endpoints, external endpoints can be added either using `Add-AzTrafficManagerEndpointConfig` followed by `Set-AzTrafficManagerProfile`, or `New-AzTrafficManagerEndpoint`.

When specifying external endpoints:

- The endpoint domain name must be specified using the 'Target' parameter
- If the 'Performance' traffic-routing method is used, the 'EndpointLocation' is required. Otherwise it is optional. The value must be a valid Azure region name.
- The 'Weight' and 'Priority' are optional.

Example 1: Adding external endpoints using `Add-AzTrafficManagerEndpointConfig` **and** `Set-AzTrafficManagerProfile`

In this example, we create a Traffic Manager profile, add two external endpoints, and commit the changes.

```
$TmProfile = New-AzTrafficManagerProfile -Name myprofile -ResourceGroupName MyRG -TrafficRoutingMethod
Performance -RelativeDnsName myapp -Ttl 30 -MonitorProtocol HTTP -MonitorPort 80 -MonitorPath "/"
Add-AzTrafficManagerEndpointConfig -EndpointName eu-endpoint -TrafficManagerProfile $TmProfile -Type
ExternalEndpoints -Target app-eu.contoso.com -EndpointLocation "North Europe" -EndpointStatus Enabled
Add-AzTrafficManagerEndpointConfig -EndpointName us-endpoint -TrafficManagerProfile $TmProfile -Type
ExternalEndpoints -Target app-us.contoso.com -EndpointLocation "Central US" -EndpointStatus Enabled
Set-AzTrafficManagerProfile -TrafficManagerProfile $TmProfile
```

Example 2: Adding external endpoints using `New-AzTrafficManagerEndpoint`

In this example, we add an external endpoint to an existing profile. The profile is specified using the profile and resource group names.

```
New-AzTrafficManagerEndpoint -Name eu-endpoint -ProfileName MyProfile -ResourceGroupName MyRG -Type
ExternalEndpoints -Target app-eu.contoso.com -EndpointStatus Enabled
```

Adding 'Nested' endpoints

Each Traffic Manager profile specifies a single traffic-routing method. However, there are scenarios that require more sophisticated traffic routing than the routing provided by a single Traffic Manager profile. You can nest Traffic Manager profiles to combine the benefits of more than one traffic-routing method. Nested profiles allow you to override the default Traffic Manager behavior to support larger and more complex application deployments. For more detailed examples, see Nested Traffic Manager profiles.

Nested endpoints are configured at the parent profile, using a specific endpoint type, 'NestedEndpoints'. When specifying nested endpoints:

- The endpoint must be specified using the 'targetResourceId' parameter
- If the 'Performance' traffic-routing method is used, the 'EndpointLocation' is required. Otherwise it is optional. The value must be a valid Azure region name.
- The 'Weight' and 'Priority' are optional, as for Azure endpoints.
- The 'MinChildEndpoints' parameter is optional. The default value is '1'. If the number of available endpoints falls below this threshold, the parent profile considers the child profile 'degraded' and diverts traffic to the other endpoints in the parent profile.

Example 1: Adding nested endpoints using `Add-AzTrafficManagerEndpointConfig` and `Set-AzTrafficManagerProfile`

In this example, we create new Traffic Manager child and parent profiles, add the child as a nested endpoint to the parent, and commit the changes.

```
$child = New-AzTrafficManagerProfile -Name child -ResourceGroupName MyRG -TrafficRoutingMethod Priority -
RelativeDnsName child -Ttl 30 -MonitorProtocol HTTP -MonitorPort 80 -MonitorPath "/"
$parent = New-AzTrafficManagerProfile -Name parent -ResourceGroupName MyRG -TrafficRoutingMethod Performance -
RelativeDnsName parent -Ttl 30 -MonitorProtocol HTTP -MonitorPort 80 -MonitorPath "/"
Add-AzTrafficManagerEndpointConfig -EndpointName child-endpoint -TrafficManagerProfile $parent -Type
NestedEndpoints -TargetResourceId $child.Id -EndpointStatus Enabled -EndpointLocation "North Europe" -
MinChildEndpoints 2
Set-AzTrafficManagerProfile -TrafficManagerProfile $parent
```

For brevity in this example, we did not add any other endpoints to the child or parent profiles.

Example 2: Adding nested endpoints using `New-AzTrafficManagerEndpoint`

In this example, we add an existing child profile as a nested endpoint to an existing parent profile. The profile is specified using the profile and resource group names.

```
$child = Get-AzTrafficManagerEndpoint -Name child -ResourceGroupName MyRG
New-AzTrafficManagerEndpoint -Name child-endpoint -ProfileName parent -ResourceGroupName MyRG -Type
NestedEndpoints -TargetResourceId $child.Id -EndpointStatus Enabled -EndpointLocation "North Europe" -
MinChildEndpoints 2
```

Adding endpoints from another subscription

Traffic Manager can work with endpoints from different subscriptions. You need to switch to the subscription with the endpoint you want to add to retrieve the needed input to Traffic Manager. Then you need to switch to the subscriptions with the Traffic Manager profile, and add the endpoint to it. The below example shows how to do this with a public IP address.

```
Set-AzContext -SubscriptionId $EndpointSubscription
$ip = Get-AzPublicIpAddress -Name $IpAddressName -ResourceGroupName $EndpointRG

Set-AzContext -SubscriptionId $trafficmanagerSubscription
New-AzTrafficManagerEndpoint -Name $EndpointName -ProfileName $ProfileName -ResourceGroupName $TrafficManagerRG
-Type AzureEndpoints -TargetResourceId $ip.Id -EndpointStatus Enabled
```

Update a Traffic Manager Endpoint

There are two ways to update an existing Traffic Manager endpoint:

1. Get the Traffic Manager profile using `Get-AzTrafficManagerProfile`, update the endpoint properties within the profile, and commit the changes using `Set-AzTrafficManagerProfile`. This method has the advantage of being able to update more than one endpoint in a single operation.
2. Get the Traffic Manager endpoint using `Get-AzTrafficManagerEndpoint`, update the endpoint properties, and commit the changes using `Set-AzTrafficManagerEndpoint`. This method is simpler, since it does not require indexing into the Endpoints array in the profile.

Example 1: Updating endpoints using `Get-AzTrafficManagerProfile` and `Set-AzTrafficManagerProfile`

In this example, we modify the priority on two endpoints within an existing profile.

```
$TmProfile = Get-AzTrafficManagerProfile -Name myprofile -ResourceGroupName MyRG
$TmProfile.Endpoints[0].Priority = 2
$TmProfile.Endpoints[1].Priority = 1
Set-AzTrafficManagerProfile -TrafficManagerProfile $TmProfile
```

Example 2: Updating an endpoint using `Get-AzTrafficManagerEndpoint` and `Set-AzTrafficManagerEndpoint`

In this example, we modify the weight of a single endpoint in an existing profile.

```
$endpoint = Get-AzTrafficManagerEndpoint -Name myendpoint -ProfileName myprofile -ResourceGroupName MyRG -Type
ExternalEndpoints
$endpoint.Weight = 20
Set-AzTrafficManagerEndpoint -TrafficManagerEndpoint $endpoint
```

Enabling and Disabling Endpoints and Profiles

Traffic Manager allows individual endpoints to be enabled and disabled, as well as allowing enabling and disabling of entire profiles. These changes can be made by getting/updating/setting the endpoint or profile resources. To streamline these common operations, they are also supported via dedicated cmdlets.

Example 1: Enabling and disabling a Traffic Manager profile

To enable a Traffic Manager profile, use `Enable-AzTrafficManagerProfile`. The profile can be specified using a profile object. The profile object can be passed via the pipeline or by using the '-TrafficManagerProfile' parameter. In this example, we specify the profile by the profile and resource group name.

```
Enable-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyResourceGroup
```

To disable a Traffic Manager profile:

```
Disable-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyResourceGroup
```

The Disable-AzTrafficManagerProfile cmdlet prompts for confirmation. This prompt can be suppressed using the '-Force' parameter.

Example 2: Enabling and disabling a Traffic Manager endpoint

To enable a Traffic Manager endpoint, use `Enable-AzTrafficManagerEndpoint`. There are two ways to specify the endpoint

1. Using a TrafficManagerEndpoint object passed via the pipeline or using the '-TrafficManagerEndpoint' parameter
2. Using the endpoint name, endpoint type, profile name, and resource group name:

```
Enable-AzTrafficManagerEndpoint -Name MyEndpoint -Type AzureEndpoints -ProfileName MyProfile -ResourceGroupName
MyRG
```

Similarly, to disable a Traffic Manager endpoint:

```
Disable-AzTrafficManagerEndpoint -Name MyEndpoint -Type AzureEndpoints -ProfileName MyProfile -
ResourceGroupName MyRG -Force
```

As with `Disable-AzTrafficManagerProfile`, the `Disable-AzTrafficManagerEndpoint` cmdlet prompts for confirmation. This prompt can be suppressed using the '-Force' parameter.

Delete a Traffic Manager Endpoint

To remove individual endpoints, use the `Remove-AzTrafficManagerEndpoint` cmdlet:

```
Remove-AzTrafficManagerEndpoint -Name MyEndpoint -Type AzureEndpoints -ProfileName MyProfile -ResourceGroupName
MyRG
```

This cmdlet prompts for confirmation. This prompt can be suppressed using the '-Force' parameter.

Delete a Traffic Manager Profile

To delete a Traffic Manager profile, use the `Remove-AzTrafficManagerProfile` cmdlet, specifying the profile and resource group names:

```
Remove-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG [-Force]
```

This cmdlet prompts for confirmation. This prompt can be suppressed using the '-Force' parameter.

The profile to be deleted can also be specified using a profile object:

```
$TmProfile = Get-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG
Remove-AzTrafficManagerProfile -TrafficManagerProfile $TmProfile [-Force]
```

This sequence can also be piped:

```
Get-AzTrafficManagerProfile -Name MyProfile -ResourceGroupName MyRG | Remove-AzTrafficManagerProfile [-Force]
```

Next steps

Traffic Manager monitoring

Traffic Manager performance considerations

Point a company Internet domain to an Azure Traffic Manager domain

2/1/2020 • 2 minutes to read • Edit Online

When you create a Traffic Manager profile, Azure automatically assigns a DNS name for that profile. To use a name from your DNS zone, create a CNAME DNS record that maps to the domain name of your Traffic Manager profile. You can find the Traffic Manager domain name in the **General** section on the Configuration page of the Traffic Manager profile.

For example, to point name `www.contoso.com` to the Traffic Manager DNS name `contoso.trafficmanager.net`, you create the following DNS resource record:

```
www.contoso.com IN CNAME contoso.trafficmanager.net
```

All traffic requests to *www.contoso.com* get directed to *contoso.trafficmanager.net*.

> **IMPORTANT**
> You cannot point a second-level domain, such as *contoso.com*, to the Traffic Manager domain. DNS protocol standards do not allow CNAME records for second-level domain names.

Next steps

- Traffic Manager routing methods
- Traffic Manager - Disable, enable or delete a profile
- Traffic Manager - Disable or enable an endpoint

Traffic Manager subnet override using Azure CLI

2/1/2020 • 3 minutes to read • Edit Online

Traffic Manager subnet override allows you to alter the routing method of a profile. The addition of an override will direct traffic based upon the end user's IP address with a predefined IP range to endpoint mapping.

How subnet override works

When subnet overrides are added to a traffic manager profile, Traffic Manager will first check if there's a subnet override for the end user's IP address. If one is found, the user's DNS query will be directed to the corresponding endpoint. If a mapping is not found, Traffic Manager will fall back to the profile's original routing method.

The IP address ranges can be specified as either CIDR ranges (for example, 1.2.3.0/24) or as address ranges (for example, 1.2.3.4-5.6.7.8). The IP ranges associated with each endpoint must be unique to that endpoint. Any overlap of IP ranges among different endpoints will cause the profile to be rejected by Traffic Manager.

There are two types of routing profiles that support subnet overrides:

- **Geographic** - If Traffic Manager finds a subnet override for the DNS query's IP address, it will route the query to the endpoint whatever the health of the endpoint is.
- **Performance** - If Traffic Manager finds a subnet override for the DNS query's IP address, it will only route the traffic to the endpoint if it's healthy. Traffic Manager will fall back to the performance routing heuristic if the subnet override endpoint isn't healthy.

Create a Traffic Manager subnet override

To create a Traffic Manager subnet override, you can use Azure CLI to add the subnets for the override to the Traffic Manager endpoint.

Azure CLI

Use Azure Cloud Shell

Azure hosts Azure Cloud Shell, an interactive shell environment that you can use through your browser. You can use either Bash or PowerShell with Cloud Shell to work with Azure services. You can use the Cloud Shell preinstalled commands to run the code in this article without having to install anything on your local environment.

To start Azure Cloud Shell:

OPTION	EXAMPLE/LINK
Select **Try It** in the upper-right corner of a code block. Selecting **Try It** doesn't automatically copy the code to Cloud Shell.	
Go to https://shell.azure.com, or select the **Launch Cloud Shell** button to open Cloud Shell in your browser.	
Select the **Cloud Shell** button on the menu bar at the upper right in the Azure portal.	

To run the code in this article in Azure Cloud Shell:

1. Start Cloud Shell.
2. Select the **Copy** button on a code block to copy the code.
3. Paste the code into the Cloud Shell session by selecting **Ctrl**+**Shift**+**V** on Windows and Linux or by selecting **Cmd**+**Shift**+**V** on macOS.
4. Select **Enter** to run the code.

If you choose to install and use the CLI locally, this tutorial requires that you are running a version of the Azure CLI version 2.0.28 or later. To find the version, run `az --version`. If you need to install or upgrade, see Install Azure CLI.

Update the Traffic Manager endpoint with subnet override.

Use Azure CLI to update your endpoint with az network traffic-manager endpoint update.

```
### Add a range of IPs ###
az network traffic-manager endpoint update \
    --name MyEndpoint \
    --profile-name MyTmProfile \
    --resource-group MyResourceGroup \
    --subnets 1.2.3.4-5.6.7.8 \
    --type AzureEndpoints

### Add a subnet ###
az network traffic-manager endpoint update \
    --name MyEndpoint \
    --profile-name MyTmProfile \
    --resource-group MyResourceGroup \
    --subnets 9.10.11.0:24 \
    --type AzureEndpoints
```

You can remove the IP address ranges by running the az network traffic-manager endpoint update with the **--remove** option.

```
az network traffic-manager endpoint update \
    --name MyEndpoint \
    --profile-name MyTmProfile \
    --resource-group MyResourceGroup \
    --remove subnets \
    --type AzureEndpoints
```

Next Steps

Learn more about Traffic Manager traffic routing methods.

Learn about the Subnet traffic-routing method

Traffic Manager subnet override using Azure Powershell

2/1/2020 • 3 minutes to read • Edit Online

Traffic Manager subnet override allows you to alter the routing method of a profile. The addition of an override will direct traffic based upon the end user's IP address with a predefined IP range to endpoint mapping.

How subnet override works

When subnet overrides are added to a traffic manager profile, Traffic Manager will first check if there's a subnet override for the end user's IP address. If one is found, the user's DNS query will be directed to the corresponding endpoint. If a mapping isn't found, Traffic Manager will fall back to the profile's original routing method.

The IP address ranges can be specified as either CIDR ranges (for example, 1.2.3.0/24) or as address ranges (for example, 1.2.3.4-5.6.7.8). The IP ranges associated with each endpoint must be unique to that endpoint. Any overlap of IP ranges among different endpoints will cause the profile to be rejected by Traffic Manager.

There are two types of routing profiles that support subnet overrides:

- **Geographic** - If Traffic Manager finds a subnet override for the DNS query's IP address, it will route the query to the endpoint whatever the health of the endpoint is.
- **Performance** - If Traffic Manager finds a subnet override for the DNS query's IP address, it will only route the traffic to the endpoint if it's healthy. Traffic Manager will fall back to the performance routing heuristic if the subnet override endpoint isn't healthy.

Create a Traffic Manager subnet override

To create a Traffic Manager subnet override, you can use Azure PowerShell to add the subnets for the override to the Traffic Manager endpoint.

Azure PowerShell

> **NOTE**
> This article has been updated to use the new Azure PowerShell Az module. You can still use the AzureRM module, which will continue to receive bug fixes until at least December 2020. To learn more about the new Az module and AzureRM compatibility, see Introducing the new Azure PowerShell Az module. For Az module installation instructions, see Install Azure PowerShell.

You can run the commands that follow in the Azure Cloud Shell, or by running PowerShell from your computer. The Azure Cloud Shell is a free interactive shell. It has common Azure tools preinstalled and configured to use with your account. If you run PowerShell from your computer, you need the Azure PowerShell module, 1.0.0 or later. You can run `Get-Module -ListAvailable Az` to find the installed version. If you need to install or upgrade, see Install Azure PowerShell module. If you are running PowerShell locally, you also need to run `Login-AzAccount` to sign in to Azure.

1. **Retrieve the Traffic Manager endpoint:**

   To enable the subnet override, retrieve the endpoint you wish to add the override to and store it in a variable using Get-AzTrafficManagerEndpoint.

Replace the Name, ProfileName, and ResourceGroupName with the values of the endpoint that you're changing.

```
$TrafficManagerEndpoint = Get-AzTrafficManagerEndpoint -Name "contoso" -ProfileName "ContosoProfile" -
ResourceGroupName "ResourceGroup" -Type AzureEndpoints
```

2. **Add the IP address range to the endpoint:**

   To add the IP address range to the endpoint, you'll use Add-AzTrafficManagerIpAddressRange to add the range.

```
### Add a range of IPs ###
Add-AzTrafficManagerIPAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First "1.2.3.4" -
Last "5.6.7.8"

### Add a subnet ###
Add-AzTrafficManagerIPAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First "9.10.11.0" -
Scope 24

### Add a range of IPs with a subnet ###
Add-AzTrafficManagerIPAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First "12.13.14.0" -
Last "12.13.14.31" -Scope 27
```

   Once the ranges are added, use Set-AzTrafficManagerEndpoint to update the endpoint.

```
Set-AzTrafficManagerEndpoint -TrafficManagerEndpoint $TrafficManagerEndpoint
```

Removal of the IP address range can be completed by using Remove-AzTrafficManagerIpAddressRange.

1. **Retrieve the Traffic Manager endpoint:**

   To enable the subnet override, retrieve the endpoint you wish to add the override to and store it in a variable using Get-AzTrafficManagerEndpoint.

   Replace the Name, ProfileName, and ResourceGroupName with the values of the endpoint that you're changing.

```
$TrafficManagerEndpoint = Get-AzTrafficManagerEndpoint -Name "contoso" -ProfileName "ContosoProfile" -
ResourceGroupName "ResourceGroup" -Type AzureEndpoints
```

2. **Remove the IP address range from the endpoint:**

```
### Remove a range of IPs ###
Remove-AzTrafficManagerIpAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First "1.2.3.4" -
Last "5.6.7.8"

### Remove a subnet ###
Remove-AzTrafficManagerIpAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First "9.10.11.0"
-Scope 24

### Remove a range of IPs with a subnet ###
Remove-AzTrafficManagerIpAddressRange -TrafficManagerEndpoint $TrafficManagerEndpoint -First
"12.13.14.0" -Last "12.13.14.31" -Scope 27
```

Once the ranges are removed, use Set-AzTrafficManagerEndpoint to update the endpoint.

```
Set-AzTrafficManagerEndpoint -TrafficManagerEndpoint $TrafficManagerEndpoint
```

Next steps

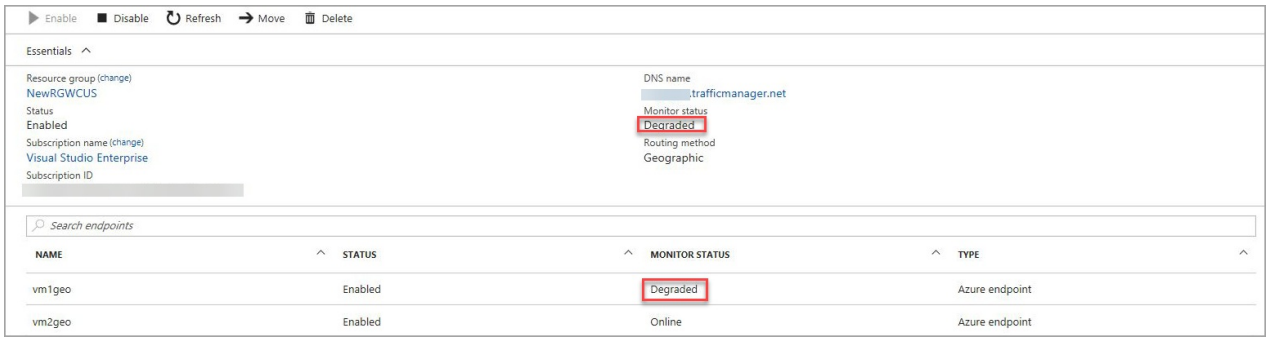
Learn more about Traffic Manager traffic routing methods.

Learn about the Subnet traffic-routing method

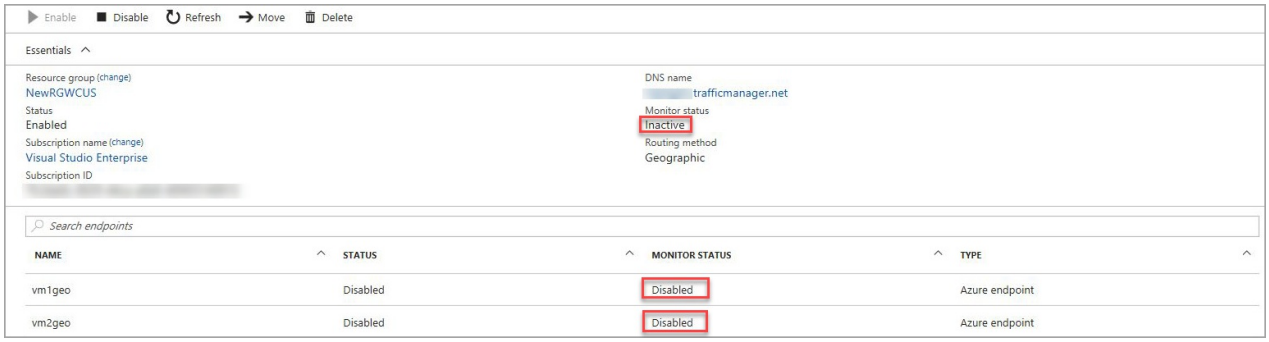
Troubleshooting degraded state on Azure Traffic Manager

2/1/2020 • 3 minutes to read • Edit Online

This article describes how to troubleshoot an Azure Traffic Manager profile that is showing a degraded status. As a first step in troubleshooting a Azure Traffic Manager degraded state is to enable diagnostic logging. Refer to Enable diagnostic logs for more information. For this scenario, consider that you have configured a Traffic Manager profile pointing to some of your cloudapp.net hosted services. If the health of your Traffic Manager displays a **Degraded** status, then the status of one or more endpoints may be **Degraded**:



If the health of your Traffic Manager displays an **Inactive** status, then both end points may be **Disabled**:



Understanding Traffic Manager probes

- Traffic Manager considers an endpoint to be ONLINE only when the probe receives an HTTP 200 response back from the probe path. If you application returns any other HTTP response code you should add that response code to Expected status code ranges of your Traffic Manager profile.
- A 30x redirect response is treated as failure unless you have specified this as a valid response code in Expected status code ranges of your Traffic Manager profile. Traffic Manager does not probe the redirection target.
- For HTTPs probes, certificate errors are ignored.
- The actual content of the probe path doesn't matter, as long as a 200 is returned. Probing a URL to some static content like "/favicon.ico" is a common technique. Dynamic content, like the ASP pages, may not always return 200, even when the application is healthy.
- A best practice is to set the probe path to something that has enough logic to determine that the site is up or down. In the previous example, by setting the path to "/favicon.ico", you are only testing that w3wp.exe is responding. This probe may not indicate that your web application is healthy. A better option would be to set a path to a something such as "/Probe.aspx" that has logic to determine the health of the site. For example, you could use performance counters to CPU utilization or measure the number of failed requests. Or you could attempt to access database resources or session state to make sure that the web application is working.

- If all endpoints in a profile are degraded, then Traffic Manager treats all endpoints as healthy and routes traffic to all endpoints. This behavior ensures that problems with the probing mechanism do not result in a complete outage of your service.

Troubleshooting

To troubleshoot a probe failure, you need a tool that shows the HTTP status code return from the probe URL. There are many tools available that show you the raw HTTP response.

- Fiddler
- curl
- wget

Also, you can use the Network tab of the F12 Debugging Tools in Internet Explorer to view the HTTP responses.

For this example we want to see the response from our probe URL: http://watestsdp2008r2.cloudapp.net:80/Probe. The following PowerShell example illustrates the problem.

```
Invoke-WebRequest 'http://watestsdp2008r2.cloudapp.net/Probe' -MaximumRedirection 0 -ErrorAction
SilentlyContinue | Select-Object StatusCode,StatusDescription
```

Example output:

```
StatusCode StatusDescription
---------- -----------------
       301 Moved Permanently
```

Notice that we received a redirect response. As stated previously, any StatusCode other than 200 is considered a failure. Traffic Manager changes the endpoint status to Offline. To resolve the problem, check the website configuration to ensure that the proper StatusCode can be returned from the probe path. Reconfigure the Traffic Manager probe to point to a path that returns a 200.

If your probe is using the HTTPS protocol, you may need to disable certificate checking to avoid SSL/TLS errors during your test. The following PowerShell statements disable certificate validation for the current PowerShell session:

```
add-type @"
using System.Net;
using System.Security.Cryptography.X509Certificates;
public class TrustAllCertsPolicy : ICertificatePolicy {
    public bool CheckValidationResult(
    ServicePoint srvPoint, X509Certificate certificate,
    WebRequest request, int certificateProblem) {
    return true;
    }
}
"@
[System.Net.ServicePointManager]::CertificatePolicy = New-Object TrustAllCertsPolicy
```

Next Steps

About Traffic Manager traffic routing methods

What is Traffic Manager

Cloud Services

Azure App Service

Operations on Traffic Manager (REST API Reference)

Azure Traffic Manager Cmdlets

Azure subscription and service limits, quotas, and constraints

2/25/2020 • 85 minutes to read • Edit Online

This document lists some of the most common Microsoft Azure limits, which are also sometimes called quotas.

To learn more about Azure pricing, see Azure pricing overview. There, you can estimate your costs by using the pricing calculator. You also can go to the pricing details page for a particular service, for example, Windows VMs. For tips to help manage your costs, see Prevent unexpected costs with Azure billing and cost management.

Managing limits

If you want to raise the limit or quota above the default limit, open an online customer support request at no charge. The limits can't be raised above the maximum limit value shown in the following tables. If there's no maximum limit column, the resource doesn't have adjustable limits.

Free Trial subscriptions aren't eligible for limit or quota increases. If you have a Free Trial subscription, you can upgrade to a Pay-As-You-Go subscription. For more information, see Upgrade your Azure Free Trial subscription to a Pay-As-You-Go subscription and the Free Trial subscription FAQ.

Some limits are managed at a regional level.

Let's use vCPU quotas as an example. To request a quota increase with support for vCPUs, you must decide how many vCPUs you want to use in which regions. You then make a specific request for Azure resource group vCPU quotas for the amounts and regions that you want. If you need to use 30 vCPUs in West Europe to run your application there, you specifically request 30 vCPUs in West Europe. Your vCPU quota isn't increased in any other region--only West Europe has the 30-vCPU quota.

As a result, decide what your Azure resource group quotas must be for your workload in any one region. Then request that amount in each region into which you want to deploy. For help in how to determine your current quotas for specific regions, see Resolve errors for resource quotas.

General limits

For limits on resource names, see Naming rules and restrictions for Azure resources.

For information about Resource Manager API read and write limits, see Throttling Resource Manager requests.

Subscription limits

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Subscriptions per Azure Active Directory tenant	Unlimited.	Unlimited.
Coadministrators per subscription	Unlimited.	Unlimited.
Resource groups per subscription	980	980

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Resource Manager API request size	4,194,304 bytes.	4,194,304 bytes.
Tags per subscription[1]	Unlimited.	Unlimited.
Unique tag calculations per subscription[1]	10,000	10,000
Subscription-level deployments per location	800[2]	800

[1]You can apply an unlimited number of tags per subscription. The number of tags per resource or resource group is limited to 50. Resource Manager returns a list of unique tag name and values in the subscription only when the number of tags is 10,000 or less. You still can find a resource by tag when the number exceeds 10,000.

[2]If you reach the limit of 800 deployments, delete deployments from the history that are no longer needed. To delete subscription level deployments, use Remove-AzDeployment or az deployment delete.

Resource group limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Resources per resource group	N/A	Resources aren't limited by resource group. Instead, they're limited by resource type in a resource group. See next row.
Resources per resource group, per resource type	800	Some resource types can exceed the 800 limit. See Resources not limited to 800 instances per resource group.
Deployments per resource group in the deployment history	800[1]	800
Resources per deployment	800	800
Management locks per unique scope	20	20
Number of tags per resource or resource group	50	50
Tag key length	512	512
Tag value length	256	256

[1]If you reach the limit of 800 deployments per resource group, delete deployments from the history that are no longer needed. Deleting an entry from the deployment history doesn't affect the deployed resources. For more information, see Resolve error when deployment count exceeds 800.

Template limits

VALUE	DEFAULT LIMIT	MAXIMUM LIMIT
Parameters	256	256

VALUE	DEFAULT LIMIT	MAXIMUM LIMIT
Variables	256	256
Resources (including copy count)	800	800
Outputs	64	64
Template expression	24,576 chars	24,576 chars
Resources in exported templates	200	200
Template size	4 MB	4 MB
Parameter file size	64 KB	64 KB

You can exceed some template limits by using a nested template. For more information, see Use linked templates when you deploy Azure resources. To reduce the number of parameters, variables, or outputs, you can combine several values into an object. For more information, see Objects as parameters.

Active Directory limits

Here are the usage constraints and other service limits for the Azure Active Directory (Azure AD) service.

CATEGORY	LIMITS
Directories	A single user can belong to a maximum of 500 Azure AD directories as a member or a guest. A single user can create a maximum of 20 directories.
Domains	You can add no more than 900 managed domain names. If you set up all of your domains for federation with on-premises Active Directory, you can add no more than 450 domain names in each directory.
Resources	• A maximum of 50,000 Azure AD resources can be created in a single directory by users of the Free edition of Azure Active Directory by default. If you have at least one verified domain, the default directory service quota in Azure AD is extended to 300,000 Azure AD resources. • A non-admin user can create no more than 250 Azure AD resources. Both active resources and deleted resources that are available to restore count toward this quota. Only deleted Azure AD resources that were deleted fewer than 30 days ago are available to restore. Deleted Azure AD resources that are no longer available to restore count toward this quota at a value of one-quarter for 30 days. If you have developers who are likely to repeatedly exceed this quota in the course of their regular duties, you can create and assign a custom role with permission to create a limitless number of app registrations.

CATEGORY	LIMITS
Schema extensions	• String-type extensions can have a maximum of 256 characters. • Binary-type extensions are limited to 256 bytes. • Only 100 extension values, across *all* types and *all* applications, can be written to any single Azure AD resource. • Only User, Group, TenantDetail, Device, Application, and ServicePrincipal entities can be extended with string-type or binary-type single-valued attributes. • Schema extensions are available only in the Graph API version 1.21 preview. The application must be granted write access to register an extension.
Applications	A maximum of 100 users can be owners of a single application.
Application Manifest	A maximum of 1200 entries can be added in the Application Manifest.

CATEGORY	LIMITS
Groups	• A user can create a maximum of 250 groups in an Azure AD organization. • An Azure AD organization can have a maximum of 5000 dynamic groups. • A maximum of 100 users can be owners of a single group. • Any number of Azure AD resources can be members of a single group. • A user can be a member of any number of groups. • The number of members in a group that you can synchronize from your on-premises Active Directory to Azure Active Directory by using Azure AD Connect is limited to 50,000 members. • Nested Groups in Azure AD are not supported within all scenarios At this time the following are the supported scenarios with nested groups. • One group can be added as a member of another group and you can achieve group nesting. • Group membership claims (when an app is configured to receive group membership claims in the token, nested groups the signed-in user is a member of are included) • Conditional access (when scoping a conditional access policy to a group) • Restricting access to self-serve password reset • Restricting which users can do Azure AD Join and device registration The following scenarios DO NOT supported nested groups: • App role assignment (assigning groups to an app is supported, but groups nested within the directly assigned group will not have access), both for access and for provisioning • Group-based licensing (assigning a license automatically to all members of a group) • Office 365 Groups.
Application Proxy	• A maximum of 500 transactions per second per App Proxy application • A maximum of 750 transactions per second for the Azure AD organization A transaction is defined as a single http request and response for a unique resource. When throttled, clients will receive a 429 response (too many requests).

CATEGORY	LIMITS
Access Panel	• There's no limit to the number of applications that can be seen in the Access Panel per user. This applies to users assigned licenses for Azure AD Premium or the Enterprise Mobility Suite. • A maximum of 10 app tiles can be seen in the Access Panel for each user. This limit applies to users who are assigned licenses for Azure AD Free license plan. Examples of app tiles include Box, Salesforce, or Dropbox. This limit doesn't apply to administrator accounts.
Reports	A maximum of 1,000 rows can be viewed or downloaded in any report. Any additional data is truncated.
Administrative units	An Azure AD resource can be a member of no more than 30 administrative units.
Admin roles and permissions	• A group cannot be added as an owner. • A group cannot be assigned to a role. • Users' ability to read other users' directory information cannot be restricted outside of the Azure AD organization-wide switch to disable all non-admin users' access to all directory information (not recommended). More information on default permissions here. • It may take up to 15 minutes or signing out/signing in before admin role membership additions and revocations take effect.

API Management limits

RESOURCE	LIMIT
Maximum number of scale units	10 per region[1]
Cache size	5 GiB per unit[2]
Concurrent back-end connections[3] per HTTP authority	2,048 per unit[4]
Maximum cached response size	2 MiB
Maximum policy document size	256 KiB[5]
Maximum custom gateway domains per service instance[6]	20
Maximum number of CA certificates per service instance	10
Maximum number of service instances per subscription[7]	20
Maximum number of subscriptions per service instance[7]	500
Maximum number of client certificates per service instance[7]	50

RESOURCE	LIMIT
Maximum number of APIs per service instance[7]	50
Maximum number of API operations per service instance[7]	1,000
Maximum total request duration[7]	30 seconds
Maximum buffered payload size[7]	2 MiB
Maximum request URL size[8]	4096 bytes

[1]Scaling limits depend on the pricing tier. To see the pricing tiers and their scaling limits, see API Management pricing.
[2]Per unit cache size depends on the pricing tier. To see the pricing tiers and their scaling limits, see API Management pricing.
[3]Connections are pooled and reused unless explicitly closed by the back end.
[4]This limit is per unit of the Basic, Standard, and Premium tiers. The Developer tier is limited to 1,024. This limit doesn't apply to the Consumption tier.
[5]This limit applies to the Basic, Standard, and Premium tiers. In the Consumption tier, policy document size is limited to 4 KiB.
[6]This resource is available in the Premium tier only.
[7]This resource applies to the Consumption tier only.
[8]Applies to the Consumption tier only. Includes an up to 2048 bytes long query string.

App Service limits

The following App Service limits include limits for Web Apps, Mobile Apps, and API Apps.

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
Web, mobile, or API apps per Azure App Service plan[1]	10	100	Unlimited[2]	Unlimited[2]	Unlimited[2]	Unlimited[2]
App Service plan	10 per region	10 per resource group	100 per resource group	100 per resource group	100 per resource group	100 per resource group
Compute instance type	Shared	Shared	Dedicated[3]	Dedicated[3]	Dedicated[3]	Dedicated[3]
Scale out (maximum instances)	1 shared	1 shared	3 dedicated[3]	10 dedicated[3]	30 dedicated[3]	100 dedicated[4]
Storage[5]	1 GB[5]	1 GB[5]	10 GB[5]	50 GB[5]	250 GB[5]	1 TB[5]
CPU time (5 minutes)[6]	3 minutes	3 minutes	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
CPU time (day)[6]	60 minutes	240 minutes	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates	Unlimited, pay at standard rates
Memory (1 hour)	1,024 MB per App Service plan	1,024 MB per app	N/A	N/A	N/A	N/A
Bandwidth	165 MB	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply	Unlimited, data transfer rates apply
Application architecture	32-bit	32-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit	32-bit/64-bit
Web sockets per instance[7]	5	35	350	Unlimited	Unlimited	Unlimited
IP connections	600	600	Depends on instance size[8]	Depends on instance size[8]	Depends on instance size[8]	16,000
Concurrent debugger connections per application	1	1	1	5	5	5
App Service Certificates per subscription[9]	Not supported	Not supported	10	10	10	10
Custom domains per app	0 (azurewebsites.net subdomain only)	500	500	500	500	500
Custom domain SSL support	Not supported, wildcard certificate for *.azurewebsites.net available by default	Not supported, wildcard certificate for *.azurewebsites.net available by default	Unlimited SNI SSL connections	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included	Unlimited SNI SSL and 1 IP SSL connections included
Hybrid connections per plan			5	25	200	200
Integrated load balancer		X	X	X	X	X[10]
Always On			X	X	X	X

RESOURCE	FREE	SHARED	BASIC	STANDARD	PREMIUM (V2)	ISOLATED
Scheduled backups				Scheduled backups every 2 hours, a maximum of 12 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)	Scheduled backups every hour, a maximum of 50 backups per day (manual + scheduled)
Autoscale				X	X	X
WebJobs[11]	X	X	X	X	X	X
Endpoint monitoring			X	X	X	X
Staging slots				5	20	20
SLA			99.95%	99.95%	99.95%	99.95%

[1]Apps and storage quotas are per App Service plan unless noted otherwise.
[2]The actual number of apps that you can host on these machines depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.
[3]Dedicated instances can be of different sizes. For more information, see App Service pricing.
[4]More are allowed upon request.
[5]The storage limit is the total content size across all apps in the same App service plan. The total content size of all apps across all App service plans in a single resource group and region cannot exceed 500GB.
[6]These resources are constrained by physical resources on the dedicated instances (the instance size and the number of instances).
[7]If you scale an app in the Basic tier to two instances, you have 350 concurrent connections for each of the two instances. For Standard tier and above, there are no theoretical limits to web sockets, but other factors can limit the number of web sockets. For example, maximum concurrent requests allowed (defined by `maxConcurrentRequestsPerCpu`) are: 7,500 per small VM, 15,000 per medium VM (7,500 x 2 cores), and 75,000 per large VM (18,750 x 4 cores).
[8]The maximum IP connections are per instance and depend on the instance size: 1,920 per B1/S1/P1V2 instance, 3,968 per B2/S2/P2V2 instance, 8,064 per B3/S3/P3V2 instance.
[9]The App Service Certificate quota limit per subscription can be increased via a support request to a maximum limit of 200.
[10]App Service Isolated SKUs can be internally load balanced (ILB) with Azure Load Balancer, so there's no public connectivity from the internet. As a result, some features of an ILB Isolated App Service must be used from machines that have direct access to the ILB network endpoint.
[11]Run custom executables and/or scripts on demand, on a schedule, or continuously as a background task within your App Service instance. Always On is required for continuous WebJobs execution. There's no predefined limit on the number of WebJobs that can run in an App Service instance. There are practical limits that depend on what the application code is trying to do.

Automation limits

Process automation

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum number of new jobs that can be submitted every 30 seconds per Azure Automation account (nonscheduled jobs)	100	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum number of concurrent running jobs at the same instance of time per Automation account (nonscheduled jobs)	200	When this limit is reached, the subsequent requests to create a job fail. The client receives an error response.
Maximum storage size of job metadata for a 30-day rolling period	10 GB (approximately 4 million jobs)	When this limit is reached, the subsequent requests to create a job fail.
Maximum job stream limit	1MB	A single stream cannot be larger than 1 MB.
Maximum number of modules that can be imported every 30 seconds per Automation account	5	
Maximum size of a module	100 MB	
Job run time, Free tier	500 minutes per subscription per calendar month	
Maximum amount of disk space allowed per sandbox[1]	1 GB	Applies to Azure sandboxes only.
Maximum amount of memory given to a sandbox[1]	400 MB	Applies to Azure sandboxes only.
Maximum number of network sockets allowed per sandbox[1]	1,000	Applies to Azure sandboxes only.
Maximum runtime allowed per runbook[1]	3 hours	Applies to Azure sandboxes only.
Maximum number of Automation accounts in a subscription	No limit	
Maximum number of Hybrid Worker Groups per Automation Account	4,000	
Maximum number of concurrent jobs that can be run on a single Hybrid Runbook Worker	50	
Maximum runbook job parameter size	512 kilobits	
Maximum runbook parameters	50	If you reach the 50-parameter limit, you can pass a JSON or XML string to a parameter and parse it with the runbook.

RESOURCE	MAXIMUM LIMIT	NOTES
Maximum webhook payload size	512 kilobits	
Maximum days that job data is retained	30 days	
Maximum PowerShell workflow state size	5 MB	Applies to PowerShell workflow runbooks when checkpointing workflow.

[1]A sandbox is a shared environment that can be used by multiple jobs. Jobs that use the same sandbox are bound by the resource limitations of the sandbox.

Change Tracking and Inventory

The following table shows the tracked item limits per machine for change tracking.

RESOURCE	LIMIT	NOTES
File	500	
Registry	250	
Windows software	250	Doesn't include software updates.
Linux packages	1,250	
Services	250	
Daemon	250	

Update Management

The following table shows the limits for Update Management.

RESOURCE	LIMIT	NOTES
Number of machines per update deployment	1000	

Azure Cache for Redis limits

RESOURCE	LIMIT
Cache size	1.2 TB
Databases	64
Maximum connected clients	40,000
Azure Cache for Redis replicas, for high availability	1
Shards in a premium cache with clustering	10

Azure Cache for Redis limits and sizes are different for each pricing tier. To see the pricing tiers and their associated sizes, see Azure Cache for Redis pricing.

For more information on Azure Cache for Redis configuration limits, see Default Redis server configuration.

Because configuration and management of Azure Cache for Redis instances is done by Microsoft, not all Redis commands are supported in Azure Cache for Redis. For more information, see Redis commands not supported in Azure Cache for Redis.

Azure Cloud Services limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Web or worker roles per deployment[1]	25	25
Instance input endpoints per deployment	25	25
Input endpoints per deployment	25	25
Internal endpoints per deployment	25	25
Hosted service certificates per deployment	199	199

[1]Each Azure Cloud Service with web or worker roles can have two deployments, one for production and one for staging. This limit refers to the number of distinct roles, that is, configuration. This limit doesn't refer to the number of instances per role, that is, scaling.

Azure Cognitive Search limits

Pricing tiers determine the capacity and limits of your search service. Tiers include:

- **Free** multi-tenant service, shared with other Azure subscribers, is intended for evaluation and small development projects.
- **Basic** provides dedicated computing resources for production workloads at a smaller scale, with up to three replicas for highly available query workloads.
- **Standard**, which includes S1, S2, S3, and S3 High Density, is for larger production workloads. Multiple levels exist within the Standard tier so that you can choose a resource configuration that best matches your workload profile.

Limits per subscription

You can create multiple services within a subscription. Each one can be provisioned at a specific tier. You're limited only by the number of services allowed at each tier. For example, you could create up to 12 services at the Basic tier and another 12 services at the S1 tier within the same subscription. For more information about tiers, see Choose an SKU or tier for Azure Cognitive Search.

Maximum service limits can be raised upon request. If you need more services within the same subscription, contact Azure Support.

RESOURCE	FREE[1]	BASIC	S1	S2	S3	S3 HD	L1	L2
Maximum services	1	16	16	8	6	6	6	6

RESOURCE	FREE	BASIC	S1	S2	S3	S3 HD	L1	L2
Maximum scale in search units (SU)[2]	N/A	3 SU	36 SU	36 SU	36 SU	36 SU	36 SU	36 SU

[1] Free is based on shared, not dedicated, resources. Scale-up is not supported on shared resources.

[2] Search units are billing units, allocated as either a *replica* or a *partition*. You need both resources for storage, indexing, and query operations. To learn more about SU computations, see Scale resource levels for query and index workloads.

Limits per search service

Storage is constrained by disk space or by a hard limit on the *maximum number* of indexes, document, or other high-level resources, whichever comes first. The following table documents storage limits. For maximum limits on indexes, documents, and other objects, see Limits by resource.

RESOURCE	FREE	BASIC[1]	S1	S2	S3	S3 HD[2]	L1	L2
Service level agreement (SLA)[3]	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Storage per partition	50 MB	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Partitions per service	N/A	1	12	12	12	3	12	12
Partition size	N/A	2 GB	25 GB	100 GB	200 GB	200 GB	1 TB	2 TB
Replicas	N/A	3	12	12	12	12	12	12

[1] Basic has one fixed partition. At this tier, additional search units are used for allocating more replicas for increased query workloads.

[2] S3 HD has a hard limit of three partitions, which is lower than the partition limit for S3. The lower partition limit is imposed because the index count for S3 HD is substantially higher. Given that service limits exist for both computing resources (storage and processing) and content (indexes and documents), the content limit is reached first.

[3] Service level agreements are offered for billable services on dedicated resources. Free services and preview features have no SLA. For billable services, SLAs take effect when you provision sufficient redundancy for your service. Two or more replicas are required for query (read) SLAs. Three or more replicas are required for query and indexing (read-write) SLAs. The number of partitions isn't an SLA consideration.

To learn more about limits on a more granular level, such as document size, queries per second, keys, requests, and responses, see Service limits in Azure Cognitive Search.

Azure Cognitive Services limits

The following limits are for the number of Cognitive Services resources per Azure subscription. Each of the Cognitive Services may have additional limitations, for more information see Azure Cognitive Services.

TYPE	LIMIT	EXAMPLE
A mixture of Cognitive Services resources	Maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, 50 Speech Service resources in West US, and 50 Text Analytics resources in East US.
A single type of Cognitive Services resources.	Maximum of 100 resources per region, with a maximum of 200 total Cognitive Services resources.	100 Computer Vision resources in West US 2, and 100 Computer Vision resources in East US.

Azure Cosmos DB limits

For Azure Cosmos DB limits, see Limits in Azure Cosmos DB.

Azure Data Explorer limits

The following table describes the maximum limits for Azure Data Explorer clusters.

RESOURCE	LIMIT
Clusters per region per subscription	20
Instances per cluster	1000
Number of databases in a cluster	10,000
Number of attached database configurations in a cluster	70

The following table describes the limits on management operations performed on Azure Data Explorer clusters.

SCOPE	OPERATION	LIMIT
Cluster	read (for example, get a cluster)	500 per 5 minutes
Cluster	write (for example, create a database)	1000 per hour

Azure Database for MySQL

For Azure Database for MySQL limits, see Limitations in Azure Database for MySQL.

Azure Database for PostgreSQL

For Azure Database for PostgreSQL limits, see Limitations in Azure Database for PostgreSQL.

Azure Functions limits

RESOURCE	CONSUMPTION PLAN	PREMIUM PLAN	APP SERVICE PLAN[1]
Scale out	Event driven	Event driven	Manual/autoscale

RESOURCE	CONSUMPTION PLAN	PREMIUM PLAN	APP SERVICE PLAN
Max instances	200	100	10-20
Default timeout duration (min)	5	30	30[2]
Max timeout duration (min)	10	unbounded[8]	unbounded[3]
Max outbound connections (per instance)	600 active (1200 total)	unbounded	unbounded
Max request size (MB)[4]	100	100	100
Max query string length[4]	4096	4096	4096
Max request URL length[4]	8192	8192	8192
ACU per instance	100	210-840	100-840
Max memory (GB per instance)	1.5	3.5-14	1.75-14
Function apps per plan	100	100	unbounded[5]
App Service plans	100 per region	100 per resource group	100 per resource group
Storage[6]	1 GB	250 GB	50-1000 GB
Custom domains per app	500[7]	500	500
Custom domain SSL support	unbounded SNI SSL connection included	unbounded SNI SSL and 1 IP SSL connections included	unbounded SNI SSL and 1 IP SSL connections included

[1] For specific limits for the various App Service plan options, see the App Service plan limits.
[2] By default, the timeout for the Functions 1.x runtime in an App Service plan is unbounded.
[3] Requires the App Service plan be set to Always On. Pay at standard rates.
[4] These limits are set in the host.
[5] The actual number of function apps that you can host depends on the activity of the apps, the size of the machine instances, and the corresponding resource utilization.
[6] The storage limit is the total content size in temporary storage across all apps in the same App Service plan. Consumption plan uses Azure Files for temporary storage.
[7] When your function app is hosted in a Consumption plan, only the CNAME option is supported. For function apps in a Premium plan or an App Service plan, you can map a custom domain using either a CNAME or an A record.
[8] Guaranteed for up to 60 minutes.

Azure Kubernetes Service limits

RESOURCE	DEFAULT LIMIT
Maximum clusters per subscription	100

RESOURCE	DEFAULT LIMIT
Maximum nodes per cluster with Virtual Machine Availability Sets and Basic Load Balancer SKU	100
Maximum nodes per cluster with Virtual Machine Scale Sets and Standard Load Balancer SKU	1000 (100 nodes per node pool)
Maximum pods per node: Basic networking with Kubenet	110
Maximum pods per node: Advanced networking with Azure Container Networking Interface	Azure CLI deployment: 30[1] Azure Resource Manager template: 30[1] Portal deployment: 30

[1]When you deploy an Azure Kubernetes Service (AKS) cluster with the Azure CLI or a Resource Manager template, this value is configurable up to 250 pods per node. You can't configure maximum pods per node after you've already deployed an AKS cluster, or if you deploy a cluster by using the Azure portal.

Azure Machine Learning limits

The latest values for Azure Machine Learning Compute quotas can be found in the Azure Machine Learning quota page

Azure Maps limits

The following table shows the usage limit for the Azure Maps S0 pricing tier. Usage limit depends on the pricing tier.

RESOURCE	S0 PRICING TIER LIMIT
Maximum request rate per subscription	50 requests per second

The following table shows the data size limit for Azure Maps. The Azure Maps data service is available only at the S1 pricing tier.

RESOURCE	LIMIT
Maximum size of data	50 MB

For more information on the Azure Maps pricing tiers, see Azure Maps pricing.

Azure Monitor limits

Alerts

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Metric alerts (classic)	100 active alert rules per subscription.	Call support.
Metric alerts	1000 active alert rules per subscription in Azure public, Azure China 21Vianet and Azure Government clouds.	Call support.
Activity log alerts	100 active alert rules per subscription.	Same as default.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Log alerts	512	Call support.
Action groups	2,000 action groups per subscription.	Call support.
Autoscale settings	100 per region per subscription.	Same as default.

Action groups

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure app push	10 Azure app actions per action group.	Call support.
Email	1,000 email actions in an action group. No more than 100 emails in an hour. Also see the rate limiting information.	Call support.
ITSM	10 ITSM actions in an action group.	Call support.
Logic app	10 logic app actions in an action group.	Call support.
Runbook	10 runbook actions in an action group.	Call support.
SMS	10 SMS actions in an action group. No more than 1 SMS message every 5 minutes. Also see the rate limiting information.	Call support.
Voice	10 voice actions in an action group. No more than 1 voice call every 5 minutes. Also see the rate limiting information.	Call support.
Webhook	10 webhook actions in an action group. Maximum number of webhook calls is 1500 per minute per subscription. Other limits are available at action-specific information.	Call support.

Log queries and language

LIMIT	DESCRIPTION
Query language	Azure Monitor uses the same Kusto query language as Azure Data Explorer. See Azure Monitor log query language differences for KQL language elements not supported in Azure Monitor.
Azure regions	Log queries can experience excessive overhead when data spans Log Analytics workspaces in multiple Azure regions. See Query limits for details.

LIMIT	DESCRIPTION
Cross resource queries	Maximum number of Application Insights resources and Log Analytics workspaces in a single query limited to 100. Cross-resource query is not supported in View Designer. Cross-resource query in log alerts is supported in the new scheduledQueryRules API. See Cross-resource query limits for details.
Query throttling	A user is limited to 200 queries per 30 seconds on any number of workspaces. This limit applies to programmatic queries or to queries initiated by visualization parts such as Azure dashboards and the Log Analytics workspace summary page.

Log Analytics workspaces

Data collection volume and retention

TIER	LIMIT PER DAY	DATA RETENTION	COMMENT
Current Per GB pricing tier (introduced April 2018)	No limit	30 - 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Free tiers (introduced April 2016)	500 MB	7 days	When your workspace reaches the 500 MB per day limit, data ingestion stops and resumes at the start of the next day. A day is based on UTC. Note that data collected by Azure Security Center is not included in this 500 MB per day limit and will continue to be collected above this limit.
Legacy Standalone Per GB tier (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Per Node (OMS) (introduced April 2016)	No limit	30 to 730 days	Data retention beyond 31 days is available for additional charges. Learn more about Azure Monitor pricing.
Legacy Standard tier	No limit	30 days	Retention can't be adjusted
Legacy Premium tier	No limit	365 days	Retention can't be adjusted

Number of workspaces per subscription.

PRICING TIER	WORKSPACE LIMIT	COMMENTS
Free tier	10	This limit can't be increased.
All other tiers	No limit	You're limited by the number of resources within a resource group and the number of resource groups per subscription.

Azure portal

CATEGORY	LIMITS	COMMENTS
Maximum records returned by a log query	10,000	Reduce results using query scope, time range, and filters in the query.

Data Collector API

CATEGORY	LIMITS	COMMENTS
Maximum size for a single post	30 MB	Split larger volumes into multiple posts.
Maximum size for field values	32 KB	Fields longer than 32 KB are truncated.

Search API

CATEGORY	LIMITS	COMMENTS
Maximum records returned in a single query	500,000	
Maximum size of data returned	64,000,000 bytes (~61 MiB)	
Maximum query running time	10 minutes	See Timeouts for details.
Maximum request rate	200 requests per 30 seconds per AAD user or client IP address	See Rate limits for details.

General workspace limits

CATEGORY	LIMITS	COMMENTS
Maximum columns in a table	500	
Maximum characters for column name	500	
Data export	Not currently available	Use Azure Function or Logic App to aggregate and export data.

Data ingestion volume rate

Azure Monitor is a high scale data service that serves thousands of customers sending terabytes of data each month at a growing pace. The default ingestion volume rate limit for data sent from Azure resources using diagnostic settings is approximately **6 GB/min** per workspace. This is an approximate value since the actual size can vary between data types depending on the log length and its compression ratio. This limit does not apply to

data that is sent from agents or Data Collector API.

If you send data at a higher rate to a single workspace, some data is dropped, and an event is sent to the *Operation* table in your workspace every 6 hours while the threshold continues to be exceeded. If your ingestion volume continues to exceed the rate limit or you are expecting to reach it sometime soon, you can request an increase to your workspace by opening a support request.

To be notified on such an event in your workspace, create a log alert rule using the following query with alert logic base on number of results grater than zero.

```
Operation
|where OperationCategory == "Ingestion"
|where Detail startswith "The rate of data crossed the threshold"
```

> **NOTE**
> Depending on how long you've been using Log Analytics, you might have access to legacy pricing tiers. Learn more about Log Analytics legacy pricing tiers.

Application Insights

There are some limits on the number of metrics and events per application, that is, per instrumentation key. Limits depend on the pricing plan that you choose.

RESOURCE	DEFAULT LIMIT	NOTE
Total data per day	100 GB	You can reduce data by setting a cap. If you need more data, you can increase the limit in the portal, up to 1,000 GB. For capacities greater than 1,000 GB, send email to AIDataCap@microsoft.com.
Throttling	32,000 events/second	The limit is measured over a minute.
Data retention	90 days	This resource is for Search, Analytics, and Metrics Explorer.
Availability multi-step test detailed results retention	90 days	This resource provides detailed results of each step.
Maximum event size	64,000,000 bytes	
Property and metric name length	150	See type schemas.
Property value string length	8,192	See type schemas.
Trace and exception message length	32,768	See type schemas.
Availability tests count per app	100	
Profiler data retention	5 days	
Profiler data sent per day	10 GB	

For more information, see About pricing and quotas in Application Insights.

Azure Policy limits

There's a maximum count for each object type for Azure Policy. An entry of *Scope* means either the subscription or the management group.

WHERE	WHAT	MAXIMUM COUNT
Scope	Policy definitions	500
Scope	Initiative definitions	100
Tenant	Initiative definitions	1,000
Scope	Policy or initiative assignments	100
Policy definition	Parameters	20
Initiative definition	Policies	100
Initiative definition	Parameters	100
Policy or initiative assignments	Exclusions (notScopes)	400
Policy rule	Nested conditionals	512

Azure SignalR Service limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure SignalR Service units per instance for Free tier	1	1
Azure SignalR Service units per instance for Standard tier	100	100
Azure SignalR Service units per subscription per region for Free tier	5	5
Total Azure SignalR Service unit counts per subscription per region	150	Unlimited
Connections per unit per day for Free tier	20	20
Connections per unit per day for Standard tier	1,000	1,000
Included messages per unit per day for Free tier	20,000	20,000

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Included messages per unit per day for Standard tier	1,000,000	1,000,000

To request an update to your subscription's default limits, open a support ticket.

Backup limits

For a summary of Azure Backup support settings and limitations, see Azure Backup Support Matrices.

Batch limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Azure Batch accounts per region per subscription	1-3	50
Dedicated cores per Batch account	90-900	Contact support
Low-priority cores per Batch account	10-100	Contact support
Active jobs and job schedules per Batch account (**completed** jobs have no limit)	100-300	1,000[1]
Pools per Batch account	20-100	500[1]

> **NOTE**
> Default limits vary depending on the type of subscription you use to create a Batch account. Cores quotas shown are for Batch accounts in Batch service mode. View the quotas in your Batch account.

[1]To request an increase beyond this limit, contact Azure Support.

Classic deployment model limits

If you use classic deployment model instead of the Azure Resource Manager deployment model, the following limits apply.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
vCPUs per subscription[1]	20	10,000
Coadministrators per subscription	200	200
Storage accounts per subscription[2]	100	100
Cloud services per subscription	20	200
Local networks per subscription	10	500
DNS servers per subscription	9	100

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Reserved IPs per subscription	20	100
Affinity groups per subscription	256	256
Subscription name length (characters)	64	64

[1]Extra small instances count as one vCPU toward the vCPU limit despite using a partial CPU core.

[2]The storage account limit includes both Standard and Premium storage accounts.

Container Instances limits

RESOURCE	DEFAULT LIMIT
Standard sku container groups per region per subscription	100[1]
Dedicated sku container groups per region per subscription	0[1]
Number of containers per container group	60
Number of volumes per container group	20
Ports per IP	5
Container instance log size - running instance	4 MB
Container instance log size - stopped instance	16 KB or 1,000 lines
Container creates per hour	300[1]
Container creates per 5 minutes	100[1]
Container deletes per hour	300[1]
Container deletes per 5 minutes	100[1]

[1]To request a limit increase, create an Azure Support request.

Container Registry limits

The following table details the features and limits of the Basic, Standard, and Premium service tiers.

RESOURCE	BASIC	STANDARD	PREMIUM
Storage[1]	10 GiB	100 GiB	500 GiB
Maximum image layer size	200 GiB	200 GiB	200 GiB
ReadOps per minute[2, 3]	1,000	3,000	10,000
WriteOps per minute[2, 4]	100	500	2,000

RESOURCE	BASIC	STANDARD	PREMIUM
Download bandwidth MBps[2]	30	60	100
Upload bandwidth MBps[2]	10	20	50
Webhooks	2	10	500
Geo-replication	N/A	N/A	Supported
Content trust	N/A	N/A	Supported
Virtual network access	N/A	N/A	Preview
Repository-scoped permissions	N/A	N/A	Preview
• Tokens	N/A	N/A	20,000
• Scope maps	N/A	N/A	20,000
• Repositories per scope map	N/A	N/A	500

[1]The specified storage limits are the amount of *included* storage for each tier. You're charged an additional daily rate per GiB for image storage above these limits. For rate information, see Azure Container Registry pricing.

[2]*ReadOps*, *WriteOps*, and *Bandwidth* are minimum estimates. Azure Container Registry strives to improve performance as usage requires.

[3]A docker pull translates to multiple read operations based on the number of layers in the image, plus the manifest retrieval.

[4]A docker push translates to multiple write operations, based on the number of layers that must be pushed. A `docker push` includes *ReadOps* to retrieve a manifest for an existing image.

Content Delivery Network limits

RESOURCE	DEFAULT LIMIT
Azure Content Delivery Network profiles	25
Content Delivery Network endpoints per profile	25
Custom domains per endpoint	25

A Content Delivery Network subscription can contain one or more Content Delivery Network profiles. A Content Delivery Network profile can contain one or more Content Delivery Network endpoints. You might want to use multiple profiles to organize your Content Delivery Network endpoints by internet domain, web application, or some other criteria.

Data Factory limits

Azure Data Factory is a multitenant service that has the following default limits in place to make sure customer subscriptions are protected from each other's workloads. To raise the limits up to the maximum for your

subscription, contact support.

Version 2

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Data factories in an Azure subscription	50	Contact support.
Total number of entities, such as pipelines, data sets, triggers, linked services, and integration runtimes, within a data factory	5,000	Contact support.
Total CPU cores for Azure-SSIS Integration Runtimes under one subscription	256	Contact support.
Concurrent pipeline runs per data factory that's shared among all pipelines in the factory	10,000	Contact support.
Concurrent External activity runs per subscription per Azure Integration Runtime region External activities are managed on integration runtime but execute on linked services, including Databricks, stored procedure, HDInsights, Web, and others.	3000	Contact support.
Concurrent Pipeline activity runs per subscription per Azure Integration Runtime region Pipeline activities execute on integration runtime, including Lookup, GetMetadata, and Delete.	1000	Contact support.
Concurrent authoring operations per subscription per Azure Integration Runtime region Including test connection, browse folder list and table list, preview data.	200	Contact support.
Concurrent Data Integration Units[1] consumption per subscription per Azure Integration Runtime region	Region group 1[2]: 6000 Region group 2[2]: 3000 Region group 3[2]: 1500	Contact support.
Maximum activities per pipeline, which includes inner activities for containers	40	40
Maximum number of linked integration runtimes that can be created against a single self-hosted integration runtime	100	Contact support.
Maximum parameters per pipeline	50	50
ForEach items	100,000	100,000
ForEach parallelism	20	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum queued runs per pipeline	100	100
Characters per expression	8,192	8,192
Minimum tumbling window trigger interval	15 min	15 min
Maximum timeout for pipeline activity runs	7 days	7 days
Bytes per object for pipeline objects[3]	200 KB	200 KB
Bytes per object for dataset and linked service objects[3]	100 KB	2,000 KB
Data Integration Units[1] per copy activity run	256	Contact support.
Write API calls	1,200/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support.
Read API calls	12,500/h This limit is imposed by Azure Resource Manager, not Azure Data Factory.	Contact support.
Monitoring queries per minute	1,000	Contact support.
Entity CRUD operations per minute	50	Contact support.
Maximum time of data flow debug session	8 hrs	8 hrs
Concurrent number of data flows per factory	50	Contact support.
Concurrent number of data flow debug sessions per user per factory	3	3
Data Flow Azure IR TTL limit	4 hrs	Contact support.

[1] The data integration unit (DIU) is used in a cloud-to-cloud copy operation, learn more from Data integration units (version 2). For information on billing, see Azure Data Factory pricing.

[2] Azure Integration Runtime is globally available to ensure data compliance, efficiency, and reduced network egress costs.

REGION GROUP	REGIONS
Region group 1	Central US, East US, East US2, North Europe, West Europe, West US, West US 2

REGION GROUP	REGIONS
Region group 2	Australia East, Australia Southeast, Brazil South, Central India, Japan East, Northcentral US, Southcentral US, Southeast Asia, West Central US
Region group 3	Canada Central, East Asia, France Central, Korea Central, UK South

[3] Pipeline, data set, and linked service objects represent a logical grouping of your workload. Limits for these objects don't relate to the amount of data you can move and process with Azure Data Factory. Data Factory is designed to scale to handle petabytes of data.

Version 1

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Pipelines within a data factory	2,500	Contact support.
Data sets within a data factory	5,000	Contact support.
Concurrent slices per data set	10	10
Bytes per object for pipeline objects[1]	200 KB	200 KB
Bytes per object for data set and linked service objects[1]	100 KB	2,000 KB
Azure HDInsight on-demand cluster cores within a subscription[2]	60	Contact support.
Cloud data movement units per copy activity run[3]	32	Contact support.
Retry count for pipeline activity runs	1,000	MaxInt (32 bit)

[1] Pipeline, data set, and linked service objects represent a logical grouping of your workload. Limits for these objects don't relate to the amount of data you can move and process with Azure Data Factory. Data Factory is designed to scale to handle petabytes of data.

[2] On-demand HDInsight cores are allocated out of the subscription that contains the data factory. As a result, the previous limit is the Data Factory-enforced core limit for on-demand HDInsight cores. It's different from the core limit that's associated with your Azure subscription.

[3] The cloud data movement unit (DMU) for version 1 is used in a cloud-to-cloud copy operation, learn more from Cloud data movement units (version 1). For information on billing, see Azure Data Factory pricing.

RESOURCE	DEFAULT LOWER LIMIT	MINIMUM LIMIT
Scheduling interval	15 minutes	15 minutes
Interval between retry attempts	1 second	1 second
Retry timeout value	1 second	1 second

Web service call limits

Azure Resource Manager has limits for API calls. You can make API calls at a rate within the Azure Resource Manager API limits.

Data Lake Analytics limits

Azure Data Lake Analytics makes the complex task of managing distributed infrastructure and complex code easy. It dynamically provisions resources, and you can use it to do analytics on exabytes of data. When the job completes, it winds down resources automatically. You pay only for the processing power that was used. As you increase or decrease the size of data stored or the amount of compute used, you don't have to rewrite code. To raise the default limits for your subscription, contact support.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of concurrent jobs	20	
Maximum number of analytics units (AUs) per account	250	Use any combination of up to a maximum of 250 AUs across 20 jobs. To increase this limit, contact Microsoft Support.
Maximum script size for job submission	3 MB	
Maximum number of Data Lake Analytics accounts per region per subscription	5	To increase this limit, contact Microsoft Support.

Data Lake Store limits

Azure Data Lake Storage Gen1 is an enterprise-wide hyper-scale repository for big data analytic workloads. You can use Data Lake Storage Gen1 to capture data of any size, type, and ingestion speed in one single place for operational and exploratory analytics. There's no limit to the amount of data you can store in a Data Lake Storage Gen1 account.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of Data Lake Storage Gen1 accounts, per subscription, per region	10	To request an increase for this limit, contact support.
Maximum number of access ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.
Maximum number of default ACLs, per file or folder	32	This is a hard limit. Use groups to manage access with fewer entries.

Data Share limits

Azure Data Share enables organizations to simply and securely share data with their customers and partners.

RESOURCE	LIMIT
Maximum number of Data Share resources per Azure subscription	50

RESOURCE	LIMIT
Maximum number of sent shares per Data Share resource	100
Maximum number of received shares per Data Share resource	100
Maximum number of invitations per sent share	100
Maximum number of share subscriptions per sent share	100
Maximum number of datasets per share	100
Maximum number of snapshot schedules per share	1

Database Migration Service Limits

Azure Database Migration Service is a fully managed service designed to enable seamless migrations from multiple database sources to Azure data platforms with minimal downtime.

RESOURCE	DEFAULT LIMIT	COMMENTS
Maximum number of services per subscription, per region	2	To request an increase for this limit, contact support.

Event Grid limits

The following limits apply to Azure Event Grid system topics and custom topics, *not* event domains.

RESOURCE	LIMIT
Custom topics per Azure subscription	100
Event subscriptions per topic	500
Publish rate for a custom topic (ingress)	5,000 events per second per topic
Publish requests	250 per second
Event size	1 MB (charged in as multiple 64-KB events)

The following limits apply to event domains only.

RESOURCE	LIMIT
Topics per event domain	100,000
Event subscriptions per topic within a domain	500
Domain scope event subscriptions	50
Publish rate for an event domain (ingress)	5,000 events per second

RESOURCE	LIMIT
Publish requests	250 per second
Event Domains per Azure Subscription	100

Event Hubs limits

The following tables provide quotas and limits specific to Azure Event Hubs. For information about Event Hubs pricing, see Event Hubs pricing.

The following limits are common across basic, standard, and dedicated tiers.

LIMIT	SCOPE	NOTES	VALUE
Number of Event Hubs namespaces per subscription	Subscription	-	100
Number of event hubs per namespace	Namespace	Subsequent requests for creation of a new event hub are rejected.	10
Number of partitions per event hub	Entity	-	32
Maximum size of an event hub name	Entity	-	50 characters
Number of non-epoch receivers per consumer group	Entity	-	5
Maximum throughput units	Namespace	Exceeding the throughput unit limit causes your data to be throttled and generates a server busy exception. To request a larger number of throughput units for a Standard tier, file a support request. Additional throughput units are available in blocks of 20 on a committed purchase basis.	20
Number of authorization rules per namespace	Namespace	Subsequent requests for authorization rule creation are rejected.	12
Number of calls to the GetRuntimeInformation method	Entity	-	50 per second
Number of virtual network (VNet) and IP Config rules	Entity	-	128

Event Hubs Basic and Standard - quotas and limits

LIMIT	SCOPE	NOTES	BASIC	STANDARD
Maximum size of Event Hubs event	Entity		256 KB	1 MB
Number of consumer groups per event hub	Entity		1	20
Number of AMQP connections per namespace	Namespace	Subsequent requests for additional connections are rejected, and an exception is received by the calling code.	100	5,000
Maximum retention period of event data	Entity		1 day	1-7 days
Apache Kafka enabled namespace	Namespace	Event Hubs namespace streams applications using Kafka protocol	No	Yes
Capture	Entity	When enabled, micro-batches on the same stream	No	Yes

Event Hubs Dedicated - quotas and limits

The Event Hubs Dedicated offering is billed at a fixed monthly price, with a minimum of 4 hours of usage. The Dedicated tier offers all the features of the Standard plan, but with enterprise scale capacity and limits for customers with demanding workloads.

FEATURE	LIMITS
Bandwidth	20 CUs
Namespaces	50 per CU
Event Hubs	1000 per namespace
Ingress events	Included
Message Size	1 MB
Partitions	2000 per CU
Consumer groups	No limit per CU, 1000 per event hub
Brokered connections	100 K included
Message Retention	90 days, 10 TB included per CU
Capture	Included

Identity Manager limits

CATEGORY	LIMIT
User-assigned managed identities	• When you create user-assigned managed identities, only alphanumeric characters (0-9, a-z, and A-Z) and the hyphen (-) are supported. For the assignment to a virtual machine or virtual machine scale set to work properly, the name is limited to 24 characters. • If you use the managed identity virtual machine extension, the supported limit is 32 user-assigned managed identities. Without the managed identity virtual machine extension, the supported limit is 512 user-assigned identities.

IoT Central limits

IoT Central limits the number of applications you can deploy in a subscription to 10. If you need to increase this limit, contact Microsoft support.

IoT Hub limits

The following table lists the limits associated with the different service tiers S1, S2, S3, and F1. For information about the cost of each *unit* in each tier, see Azure IoT Hub pricing.

RESOURCE	S1 STANDARD	S2 STANDARD	S3 STANDARD	F1 FREE
Messages/day	400,000	6,000,000	300,000,000	8,000
Maximum units	200	200	10	1

> **NOTE**
> If you anticipate using more than 200 units with an S1 or S2 tier hub or 10 units with an S3 tier hub, contact Microsoft Support.

The following table lists the limits that apply to IoT Hub resources.

RESOURCE	LIMIT
Maximum paid IoT hubs per Azure subscription	100
Maximum free IoT hubs per Azure subscription	1
Maximum number of characters in a device ID	128
Maximum number of device identities returned in a single call	1,000
IoT Hub message maximum retention for device-to-cloud messages	7 days
Maximum size of device-to-cloud message	256 KB

RESOURCE	LIMIT
Maximum size of device-to-cloud batch	AMQP and HTTP: 256 KB for the entire batch MQTT: 256 KB for each message
Maximum messages in device-to-cloud batch	500
Maximum size of cloud-to-device message	64 KB
Maximum TTL for cloud-to-device messages	2 days
Maximum delivery count for cloud-to-device messages	100
Maximum cloud-to-device queue depth per device	50
Maximum delivery count for feedback messages in response to a cloud-to-device message	100
Maximum TTL for feedback messages in response to a cloud-to-device message	2 days
Maximum size of device twin	8 KB for tags section, and 32 KB for desired and reported properties sections each
Maximum length of device twin string key	1 KB
Maximum length of device twin string value	4 KB
Maximum depth of object in device twin	10
Maximum size of direct method payload	128 KB
Job history maximum retention	30 days
Maximum concurrent jobs	10 (for S3), 5 for (S2), 1 (for S1)
Maximum additional endpoints	10 (for S1, S2, and S3)
Maximum message routing rules	100 (for S1, S2, and S3)
Maximum number of concurrently connected device streams	50 (for S1, S2, S3, and F1 only)
Maximum device stream data transfer	300 MB per day (for S1, S2, S3, and F1 only)

NOTE

If you need more than 100 paid IoT hubs in an Azure subscription, contact Microsoft Support.

NOTE

Currently, the total number of devices plus modules that can be registered to a single IoT hub is capped at 1,000,000. If you want to increase this limit, contact Microsoft Support.

IoT Hub throttles requests when the following quotas are exceeded.

THROTTLE	PER-HUB VALUE
Identity registry operations (create, retrieve, list, update, and delete), individual or bulk import/export	83.33/sec/unit (5,000/min/unit) (for S3). 1.67/sec/unit (100/min/unit) (for S1 and S2).
Device connections	6,000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Device-to-cloud sends	6,000/sec/unit (for S3), 120/sec/unit (for S2), 12/sec/unit (for S1). Minimum of 100/sec.
Cloud-to-device sends	83.33/sec/unit (5,000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S1 and S2).
Cloud-to-device receives	833.33/sec/unit (50,000/min/unit) (for S3), 16.67/sec/unit (1,000/min/unit) (for S1 and S2).
File upload operations	83.33 file upload initiations/sec/unit (5,000/min/unit) (for S3), 1.67 file upload initiations/sec/unit (100/min/unit) (for S1 and S2). 10,000 SAS URIs can be out for an Azure Storage account at one time. 10 SAS URIs/device can be out at one time.
Direct methods	24 MB/sec/unit (for S3), 480 KB/sec/unit (for S2), 160 KB/sec/unit (for S1). Based on 8-KB throttling meter size.
Device twin reads	500/sec/unit (for S3), Maximum of 100/sec or 10/sec/unit (for S2), 100/sec (for S1)
Device twin updates	250/sec/unit (for S3), Maximum of 50/sec or 5/sec/unit (for S2), 50/sec (for S1)
Jobs operations (create, update, list, and delete)	83.33/sec/unit (5,000/min/unit) (for S3), 1.67/sec/unit (100/min/unit) (for S2), 1.67/sec/unit (100/min/unit) (for S1).
Jobs per-device operation throughput	50/sec/unit (for S3), maximum of 10/sec or 1/sec/unit (for S2), 10/sec (for S1).
Device stream initiation rate	5 new streams/sec (for S1, S2, S3, and F1 only).

IoT Hub Device Provisioning Service limits

The following table lists the limits that apply to Azure IoT Hub Device Provisioning Service resources.

RESOURCE	LIMIT
Maximum device provisioning services per Azure subscription	10
Maximum number of enrollments	1,000,000
Maximum number of registrations	1,000,000
Maximum number of enrollment groups	100
Maximum number of CAs	25
Maximum number of linked IoT hubs	50
Maximum size of message	96 KB

NOTE

To increase the number of enrollments and registrations on your provisioning service, contact Microsoft Support.

NOTE

Increasing the maximum number of CAs is not supported.

The Device Provisioning Service throttles requests when the following quotas are exceeded.

THROTTLE	PER-UNIT VALUE
Operations	200/min/service
Device registrations	200/min/service
Device polling operation	5/10 sec/device

Key Vault limits

Key transactions (maximum transactions allowed in 10 seconds, per vault per region[1]):

KEY TYPE	HSM KEY CREATE KEY	HSM KEY ALL OTHER TRANSACTIONS	SOFTWARE KEY CREATE KEY	SOFTWARE KEY ALL OTHER TRANSACTIONS
RSA 2,048-bit	5	1,000	10	2,000
RSA 3,072-bit	5	250	10	500
RSA 4,096-bit	5	125	10	250
ECC P-256	5	1,000	10	2,000
ECC P-384	5	1,000	10	2,000

KEY TYPE	HSM KEY CREATE KEY	HSM KEY ALL OTHER TRANSACTIONS	SOFTWARE KEY CREATE KEY	SOFTWARE KEY ALL OTHER TRANSACTIONS
ECC P-521	5	1,000	10	2,000
ECC SECP256K1	5	1,000	10	2,000

NOTE

In the previous table, we see that for RSA 2,048-bit software keys, 2,000 GET transactions per 10 seconds are allowed. For RSA 2,048-bit HSM-keys, 1,000 GET transactions per 10 seconds are allowed.

The throttling thresholds are weighted, and enforcement is on their sum. For example, as shown in the previous table, when you perform GET operations on RSA HSM-keys, it's eight times more expensive to use 4,096-bit keys compared to 2,048-bit keys. That's because 1,000/125 = 8.

In a given 10-second interval, an Azure Key Vault client can do *only one* of the following operations before it encounters a `429` throttling HTTP status code:

- 2,000 RSA 2,048-bit software-key GET transactions
- 1,000 RSA 2,048-bit HSM-key GET transactions
- 125 RSA 4,096-bit HSM-key GET transactions
- 124 RSA 4,096-bit HSM-key GET transactions and 8 RSA 2,048-bit HSM-key GET transactions

Secrets, managed storage account keys, and vault transactions:

TRANSACTIONS TYPE	MAXIMUM TRANSACTIONS ALLOWED IN 10 SECONDS, PER VAULT PER REGION[1]
All transactions	2,000

For information on how to handle throttling when these limits are exceeded, see Azure Key Vault throttling guidance.

[1] A subscription-wide limit for all transaction types is five times per key vault limit. For example, HSM-other transactions per subscription are limited to 5,000 transactions in 10 seconds per subscription.

Media Services limits

NOTE

For resources that aren't fixed, open a support ticket to ask for an increase in the quotas. Don't create additional Azure Media Services accounts in an attempt to obtain higher limits.

RESOURCE	DEFAULT LIMIT
Azure Media Services accounts in a single subscription	25 (fixed)
Media reserved units per Media Services account	25 (S1) 10 (S2, S3)[1]
Jobs per Media Services account	50,000[2]
Chained tasks per job	30 (fixed)

RESOURCE	DEFAULT LIMIT
Assets per Media Services account	1,000,000
Assets per task	50
Assets per job	100
Unique locators associated with an asset at one time	5[4]
Live channels per Media Services account	5
Programs in stopped state per channel	50
Programs in running state per channel	3
Streaming endpoints that are stopped or running per Media Services account	2
Streaming units per streaming endpoint	10
Storage accounts	1,000[5] (fixed)
Policies	1,000,000[6]
File size	In some scenarios, there's a limit on the maximum file size supported for processing in Media Services.[7]

[1]If you change the type, for example, from S2 to S1, the maximum reserved unit limits are reset.

[2]This number includes queued, finished, active, and canceled jobs. It doesn't include deleted jobs. You can delete old jobs by using **IJob.Delete** or the **DELETE** HTTP request.

As of April 1, 2017, any job record in your account older than 90 days is automatically deleted, along with its associated task records. Automatic deletion occurs even if the total number of records is below the maximum quota. To archive the job and task information, use the code described in Manage assets with the Media Services .NET SDK.

[3]When you make a request to list job entities, a maximum of 1,000 jobs is returned per request. To keep track of all submitted jobs, use the top or skip queries as described in OData system query options.

[4]Locators aren't designed for managing per-user access control. To give different access rights to individual users, use digital rights management (DRM) solutions. For more information, see Protect your content with Azure Media Services.

[5]The storage accounts must be from the same Azure subscription.

[6]There's a limit of 1,000,000 policies for different Media Services policies. An example is for the Locator policy or ContentKeyAuthorizationPolicy.

> **NOTE**
> If you always use the same days and access permissions, use the same policy ID. For information and an example, see Manage assets with the Media Services .NET SDK.

[7]The maximum size supported for a single blob is currently up to 5 TB in Azure Blob Storage. Additional limits apply in Media Services based on the VM sizes that are used by the service. The size limit applies to the files that you upload and also the files that get generated as a result of Media Services processing (encoding or analyzing). If your source file is larger than 260-GB, your Job will likely fail.

The following table shows the limits on the media reserved units S1, S2, and S3. If your source file is larger than the limits defined in the table, your encoding job fails. If you encode 4K resolution sources of long duration, you're required to use S3 media reserved units to achieve the performance needed. If you have 4K content that's larger than the 260-GB limit on the S3 media reserved units, open a support ticket.

MEDIA RESERVED UNIT TYPE	MAXIMUM INPUT SIZE (GB)
S1	26
S2	60
S3	260

Mobile Services limits

TIER	FREE	BASIC	STANDARD
API calls	500,000	1.5 million per unit	15 million per unit
Active devices	500	Unlimited	Unlimited
Scale	N/A	Up to 6 units	Unlimited units
Push notifications	Azure Notification Hubs Free tier included, up to 1 million pushes	Notification Hubs Basic tier included, up to 10 million pushes	Notification Hubs Standard tier included, up to 10 million pushes
Real-time messaging/ Web Sockets	Limited	350 per mobile service	Unlimited
Offline synchronizations	Limited	Included	Included
Scheduled jobs	Limited	Included	Included
Azure SQL Database (required) Standard rates apply for additional capacity	20 MB included	20 MB included	20 MB included
CPU capacity	60 minutes per day	Unlimited	Unlimited
Outbound data transfer	165 MB per day (daily rollover)	Included	Included

For more information on limits and pricing, see Azure Mobile Services pricing.

Multi-Factor Authentication limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of trusted IP addresses or ranges per subscription	0	50
Remember my devices, number of days	14	60
Maximum number of app passwords	0	No limit
Allow **X** attempts during MFA call	1	99
Two-way text message timeout seconds	60	600
Default one-time bypass seconds	300	1,800
Lock user account after **X** consecutive MFA denials	Not set	99
Reset account lockout counter after **X** minutes	Not set	9,999
Unlock account after **X** minutes	Not set	9,999

Networking limits

Networking limits - Azure Resource Manager The following limits apply only for networking resources managed through **Azure Resource Manager** per region per subscription. Learn how to view your current resource usage against your subscription limits.

> **NOTE**
> We recently increased all default limits to their maximum limits. If there's no maximum limit column, the resource doesn't have adjustable limits. If you had these limits increased by support in the past and don't see updated limits in the following tables, open an online customer support request at no charge

RESOURCE	DEFAULT/MAXIMUM LIMIT
Virtual networks	1,000
Subnets per virtual network	3,000
Virtual network peerings per virtual network	500
Virtual network gateways (VPN gateways) per virtual network	1
Virtual network gateways (ExpressRoute gateways) per virtual network	1
DNS servers per virtual network	20
Private IP addresses per virtual network	65,536

RESOURCE	DEFAULT/MAXIMUM LIMIT
Private IP addresses per network interface	256
Private IP addresses per virtual machine	256
Public IP addresses per network interface	256
Public IP addresses per virtual machine	256
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000
Network interface cards	65,536
Network Security Groups	5,000
NSG rules per NSG	1,000
IP addresses and ranges specified for source or destination in a security group	4,000
Application security groups	3,000
Application security groups per IP configuration, per NIC	20
IP configurations per application security group	4,000
Application security groups that can be specified within all security rules of a network security group	100
User-defined route tables	200
User-defined routes per route table	400
Point-to-site root certificates per Azure VPN Gateway	20
Virtual network TAPs	100
Network interface TAP configurations per virtual network TAP	100

Public IP address limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP addresses[1]	10 for Basic.	Contact support.
Static Public IP addresses[1]	10 for Basic.	Contact support.
Standard Public IP addresses[1]	10	Contact support.
Public IP Prefixes	limited by number of Standard Public IPs in a subscription	Contact support.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Public IP prefix length	/28	Contact support.

[1]Default limits for Public IP addresses vary by offer category type, such as Free Trial, Pay-As-You-Go, CSP. For example, the default for Enterprise Agreement subscriptions is 1000.

Load balancer limits

The following limits apply only for networking resources managed through Azure Resource Manager per region per subscription. Learn how to view your current resource usage against your subscription limits.

Standard Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	1,500
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	600
Backend pool size	1,000 IP configurations, single virtual network
High-availability ports	1 per internal frontend
Outbound rules per Load Balancer	20

Basic Load Balancer

RESOURCE	DEFAULT/MAXIMUM LIMIT
Load balancers	1,000
Rules per resource	250
Rules per NIC (across all IPs on a NIC)	300
Frontend IP configurations	200
Backend pool size	300 IP configurations, single availability set
Availability sets per Load Balancer	150

The following limits apply only for networking resources managed through the classic deployment model per subscription. Learn how to view your current resource usage against your subscription limits.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual networks	100	100
Local network sites	20	50

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
DNS servers per virtual network	20	20
Private IP addresses per virtual network	4,096	4,096
Concurrent TCP or UDP flows per NIC of a virtual machine or role instance	500,000, up to 1,000,000 for two or more NICs.	500,000, up to 1,000,000 for two or more NICs.
Network Security Groups (NSGs)	200	200
NSG rules per NSG	1,000	1,000
User-defined route tables	200	200
User-defined routes per route table	400	400
Public IP addresses (dynamic)	500	500
Reserved public IP addresses	500	500
Public VIP per deployment	5	Contact support
Private VIP (internal load balancing) per deployment	1	1
Endpoint access control lists (ACLs)	50	50

ExpressRoute limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
ExpressRoute circuits per subscription	10
ExpressRoute circuits per region per subscription, with Azure Resource Manager	10
Maximum number of routes advertised to Azure private peering with ExpressRoute Standard	4,000
Maximum number of routes advertised to Azure private peering with ExpressRoute Premium add-on	10,000
Maximum number of routes advertised from Azure private peering from the VNet address space for an ExpressRoute connection	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Standard	200
Maximum number of routes advertised to Microsoft peering with ExpressRoute Premium add-on	200

RESOURCE	DEFAULT/MAXIMUM LIMIT
Maximum number of ExpressRoute circuits linked to the same virtual network in the same peering location	4
Maximum number of ExpressRoute circuits linked to the same virtual network in different peering locations	4
Number of virtual network links allowed per ExpressRoute circuit	See the Number of virtual networks per ExpressRoute circuit table.

Number of virtual networks per ExpressRoute circuit

CIRCUIT SIZE	NUMBER OF VIRTUAL NETWORK LINKS FOR STANDARD	NUMBER OF VIRTUAL NETWORK LINKS WITH PREMIUM ADD-ON
50 Mbps	10	20
100 Mbps	10	25
200 Mbps	10	25
500 Mbps	10	40
1 Gbps	10	50
2 Gbps	10	60
5 Gbps	10	75
10 Gbps	10	100
40 Gbps*	10	100
100 Gbps*	10	100

**100 Gbps ExpressRoute Direct Only*

> **NOTE**
> Global Reach connections count against the limit of virtual network connections per ExpressRoute Circuit. For example, a 10 Gbps Premium Circuit would allow for 5 Global Reach connections and 95 connections to the ExpressRoute Gateways or 95 Global Reach connections and 5 connections to the ExpressRoute Gateways or any other combination up to the limit of 100 connections for the circuit.

Virtual WAN limits

RESOURCE	LIMIT
Virtual WAN hubs per region	1
Virtual WAN hubs per virtual wan	Azure regions
VPN (branch) connections per hub	1,000

RESOURCE	LIMIT
VNet connections per hub	500
Point-to-Site users per hub	10,000
Aggregate throughput per Virtual WAN VPN gateway	20 Gbps
Throughput per Virtual WAN VPN connection (2 tunnels)	2 Gbps with 1 Gbps/IPsec tunnel
Aggregate throughput per Virtual WAN ExpressRoute gateway	20 Gbps

Application Gateway limits

The following table applies to v1, v2, Standard, and WAF SKUs unless otherwise stated.

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Azure Application Gateway	1,000 per subscription	
Front-end IP configurations	2	1 public and 1 private
Front-end ports	100[1]	
Back-end address pools	100[1]	
Back-end servers per pool	1,200	
HTTP listeners	100[1]	
HTTP load-balancing rules	100[1]	
Back-end HTTP settings	100[1]	
Instances per gateway	V1 SKU - 32 V2 SKU - 125	
SSL certificates	100[1]	1 per HTTP listener
Maximum SSL certificate size	V1 SKU - 10 KB V2 SKU - 16 KB	
Authentication certificates	100	
Trusted root certificates	100	
Request timeout minimum	1 second	
Request timeout maximum	24 hours	
Number of sites	100[1]	1 per HTTP listener
URL maps per listener	1	

RESOURCE	DEFAULT/MAXIMUM LIMIT	NOTE
Maximum path-based rules per URL map	100	
Redirect configurations	100[1]	
Concurrent WebSocket connections	Medium gateways 20k Large gateways 50k	
Maximum URL length	32KB	
Maximum header size for HTTP/2	4KB	
Maximum file upload size, Standard	2 GB	
Maximum file upload size WAF	V1 Medium WAF gateways, 100 MB V1 Large WAF gateways, 500 MB V2 WAF, 750 MB	
WAF body size limit, without files	128 KB	
Maximum WAF custom rules	100	
Maximum WAF exclusions	100	

[1] In case of WAF-enabled SKUs, we recommend that you limit the number of resources to 40 for optimal performance.

Network Watcher limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT	NOTE
Azure Network Watcher	1 per region	1 per region	Network Watcher is created to enable access to the service. Only one instance of Network Watcher is required per subscription per region.
Packet capture sessions	10,000 per region	10,000	Number of sessions only, not saved captures.

Private Link limits

The following limits apply to Azure private link:

RESOURCE	LIMIT
Number of private endpoints per virtual network	1000
Number of private endpoints per subscription	64000
Number of private link service per subscription	800
Number of IP Configurations on a private link service	8 (This number is for the NAT IP addresses used per PLS)

RESOURCE	LIMIT
Number of private endpoints on the same private link service	1000

Traffic Manager limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Profiles per subscription	200
Endpoints per profile	200

Azure Bastion limits

RESOURCE	DEFAULT LIMIT
Concurrent RDP connections	25*
Concurrent SSH connections	More than 50**

*May vary due to other on-going RDP sessions or other on-going SSH sessions.
**May vary if there are existing RDP connections or usage from other on-going SSH sessions.

Azure DNS limits

Public DNS zones

RESOURCE	DEFAULT LIMIT
Public DNS Zones per subscription	250 [1]
Record sets per public DNS zone	10,000 [1]
Records per record set in public DNS zone	20
Number of Alias records for a single Azure resource	20
Private DNS zones per subscription	1000
Record sets per private DNS zone	25000
Records per record set for private DNS zones	20
Virtual Network Links per private DNS zone	1000
Virtual Networks Links per private DNS zones with auto-registration enabled	100
Number of private DNS zones a virtual network can get linked to with auto-registration enabled	1
Number of private DNS zones a virtual network can get linked	1000

RESOURCE	DEFAULT LIMIT
Number of DNS queries a virtual machine can send to Azure DNS resolver, per second	500 [2]
Maximum number of DNS queries queued (pending response) per virtual machine	200 [2]

[1]If you need to increase these limits, contact Azure Support.

[2]These limits are applied to every individual virtual machine and not at the virtual network level. DNS queries exceeding these limits are dropped.

Azure Firewall limits

RESOURCE	DEFAULT LIMIT
Data throughput	30 Gbps[1]
Rules	10,000. All rule types combined.
Maximum DNAT rules	299
Minimum AzureFirewallSubnet size	/26
Port range in network and application rules	0-64,000. Work is in progress to relax this limitation.
Public IP addresses	100 maximum (Currently, SNAT ports are added only for the first five public IP addresses.)
Route table	By default, AzureFirewallSubnet has a 0.0.0.0/0 route with the NextHopType value set to **Internet**. Azure Firewall must have direct Internet connectivity. If your AzureFirewallSubnet learns a default route to your on-premises network via BGP, you must override that with a 0.0.0.0/0 UDR with the **NextHopType** value set as **Internet** to maintain direct Internet connectivity. By default, Azure Firewall doesn't support forced tunneling to an on-premises network. However, if your configuration requires forced tunneling to an on-premises network, Microsoft will support it on a case by case basis. Contact Support so that we can review your case. If accepted, we'll allow your subscription and ensure the required firewall Internet connectivity is maintained.

[1]If you need to increase these limits, contact Azure Support.

Azure Front Door Service limits

RESOURCE	DEFAULT/MAXIMUM LIMIT
Azure Front Door Service resources per subscription	100
Front-end hosts, which includes custom domains per resource	100

RESOURCE	DEFAULT/MAXIMUM LIMIT
Routing rules per resource	100
Back-end pools per resource	50
Back ends per back-end pool	100
Path patterns to match for a routing rule	25
Custom web application firewall rules per policy	10
Web application firewall policy per subscription	100
Web application firewall match conditions per custom rule	10
Web application firewall IP address ranges per match condition	600
Web application firewall string match values per match condition	10
Web application firewall string match value length	256
Web application firewall POST body parameter name length	256
Web application firewall HTTP header name length	256
Web application firewall cookie name length	256
Web application firewall HTTP request body size inspected	128 KB
Web application firewall custom response body length	2 KB

Timeout values

Client to Front Door

- Front Door has an idle TCP connection timeout of 61 seconds.

Front Door to application back-end

- If the response is a chunked response, a 200 is returned if or when the first chunk is received.
- After the HTTP request is forwarded to the back end, Front Door waits for 30 seconds for the first packet from the back end. Then it returns a 503 error to the client.
- After the first packet is received from the back end, Front Door waits for 30 seconds in an idle timeout. Then it returns a 503 error to the client.
- Front Door to the back-end TCP session timeout is 30 minutes.

Upload and download data limit

	WITH CHUNKED TRANSFER ENCODING (CTE)	WITHOUT HTTP CHUNKING
Download	There's no limit on the download size.	There's no limit on the download size.

	WITH CHUNKED TRANSFER ENCODING (CTE)	WITHOUT HTTP CHUNKING
Upload	There's no limit as long as each CTE upload is less than 2 GB.	The size can't be larger than 2 GB.

Other limits

- Maximum URL size - 8,192 bytes - Specifies maximum length of the raw URL (scheme + hostname + port + path + query string of the URL)
- Maximum Query String size - 4,096 bytes - Specifies the maximum length of the query string, in bytes.

Notification Hubs limits

TIER	FREE	BASIC	STANDARD
Included pushes	1 million	10 million	10 million
Active devices	500	200,000	10 million
Tag quota per installation or registration	60	60	60

For more information on limits and pricing, see Notification Hubs pricing.

Role-based access control limits

RESOURCE	LIMIT
Role assignments for Azure resources per Azure subscription	2,000
Role assignments for Azure resources per management group	500
Custom roles for Azure resources per tenant	5,000
Custom roles for Azure resources per tenant (specialized clouds, such as Azure Government, Azure Germany, and Azure China 21Vianet)	2,000

Service Bus limits

The following table lists quota information specific to Azure Service Bus messaging. For information about pricing and other quotas for Service Bus, see Service Bus pricing.

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum number of Basic or Standard namespaces per Azure subscription	Namespace	Subsequent requests for additional Basic or Standard namespaces are rejected by the Azure portal.	100

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum number of Premium namespaces per Azure subscription	Namespace	Subsequent requests for additional Premium namespaces are rejected by the portal.	100
Queue or topic size	Entity	Defined upon creation of the queue or topic. Subsequent incoming messages are rejected, and an exception is received by the calling code.	1, 2, 3, 4 GB or 5 GB. In the Premium SKU, and the Standard SKU with partitioning enabled, the maximum queue or topic size is 80 GB.
Number of concurrent connections on a namespace	Namespace	Subsequent requests for additional connections are rejected, and an exception is received by the calling code. REST operations don't count toward concurrent TCP connections.	NetMessaging: 1,000. AMQP: 5,000.
Number of concurrent receive requests on a queue, topic, or subscription entity	Entity	Subsequent receive requests are rejected, and an exception is received by the calling code. This quota applies to the combined number of concurrent receive operations across all subscriptions on a topic.	5,000
Number of topics or queues per namespace	Namespace	Subsequent requests for creation of a new topic or queue on the namespace are rejected. As a result, if configured through the Azure portal, an error message is generated. If called from the management API, an exception is received by the calling code.	10,000 for the Basic or Standard tier. The total number of topics and queues in a namespace must be less than or equal to 10,000. For the Premium tier, 1,000 per messaging unit (MU). Maximum limit is 4,000.
Number of partitioned topics or queues per namespace	Namespace	Subsequent requests for creation of a new partitioned topic or queue on the namespace are rejected. As a result, if configured through the Azure portal, an error message is generated. If called from the management API, the exception **QuotaExceededException** is received by the calling code.	Basic and Standard tiers: 100. Partitioned entities aren't supported in the Premium tier. Each partitioned queue or topic counts toward the quota of 1,000 entities per namespace.
Maximum size of any messaging entity path: queue or topic	Entity	-	260 characters.

QUOTA NAME	SCOPE	NOTES	VALUE
Maximum size of any messaging entity name: namespace, subscription, or subscription rule	Entity	-	50 characters.
Maximum size of a message ID	Entity	-	128
Maximum size of a message session ID	Entity	-	128
Message size for a queue, topic, or subscription entity	Entity	Incoming messages that exceed these quotas are rejected, and an exception is received by the calling code.	Maximum message size: 256 KB for Standard tier, 1 MB for Premium tier. Due to system overhead, this limit is less than these values. Maximum header size: 64 KB. Maximum number of header properties in property bag: **byte/int.MaxValue**. Maximum size of property in property bag: No explicit limit. Limited by maximum header size.
Message property size for a queue, topic, or subscription entity	Entity	The exception **SerializationException** is generated.	Maximum message property size for each property is 32,000. Cumulative size of all properties can't exceed 64,000. This limit applies to the entire header of the BrokeredMessage, which has both user properties and system properties, such as SequenceNumber, Label, and MessageId.
Number of subscriptions per topic	Entity	Subsequent requests for creating additional subscriptions for the topic are rejected. As a result, if configured through the portal, an error message is shown. If called from the management API, an exception is received by the calling code.	2,000 per-topic for the Standard tier.
Number of SQL filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	2,000

QUOTA NAME	SCOPE	NOTES	VALUE
Number of correlation filters per topic	Entity	Subsequent requests for creation of additional filters on the topic are rejected, and an exception is received by the calling code.	100,000
Size of SQL filters or actions	Namespace	Subsequent requests for creation of additional filters are rejected, and an exception is received by the calling code.	Maximum length of filter condition string: 1,024 (1 K). Maximum length of rule action string: 1,024 (1 K). Maximum number of expressions per rule action: 32.
Number of SharedAccessAuthorizationRule rules per namespace, queue, or topic	Entity, namespace	Subsequent requests for creation of additional rules are rejected, and an exception is received by the calling code.	Maximum number of rules per entity type: 12. Rules that are configured on a Service Bus namespace apply to all types: queues, topics.
Number of messages per transaction	Transaction	Additional incoming messages are rejected, and an exception stating "Cannot send more than 100 messages in a single transaction" is received by the calling code.	100 For both **Send()** and **SendAsync()** operations.
Number of virtual network and IP filter rules	Namespace		128

Site Recovery limits

The following limits apply to Azure Site Recovery.

LIMIT IDENTIFIER	DEFAULT LIMIT
Number of vaults per subscription	500
Number of servers per Azure vault	250
Number of protection groups per Azure vault	No limit
Number of recovery plans per Azure vault	No limit
Number of servers per protection group	No limit
Number of servers per recovery plan	50

SQL Database limits

For SQL Database limits, see SQL Database resource limits for single databases, SQL Database resource limits for elastic pools and pooled databases, and SQL Database resource limits for managed instances.

SQL Data Warehouse limits

For SQL Data Warehouse limits, see SQL Data Warehouse resource limits.

Storage limits

The following table describes default limits for Azure general-purpose v1, v2, and Blob storage accounts. The *ingress* limit refers to all data from requests that are sent to a storage account. The *egress* limit refers to all data from responses that are received from a storage account.

RESOURCE	DEFAULT LIMIT
Number of storage accounts per region per subscription, including both standard and premium accounts	250
Maximum storage account capacity	2 PiB for US and Europe, and 500 TiB for all other regions (including the UK)[1]
Maximum number of blob containers, blobs, file shares, tables, queues, entities, or messages per storage account	No limit
Maximum request rate[1] per storage account	20,000 requests per second
Maximum ingress[1] per storage account (US, Europe regions)	25 Gbps
Maximum ingress[1] per storage account (regions other than US and Europe)	5 Gbps if RA-GRS/GRS is enabled, 10 Gbps for LRS/ZRS[2]
Maximum egress for general-purpose v2 and Blob storage accounts (all regions)	50 Gbps
Maximum egress for general-purpose v1 storage accounts (US regions)	20 Gbps if RA-GRS/GRS is enabled, 30 Gbps for LRS/ZRS[2]
Maximum egress for general-purpose v1 storage accounts (non-US regions)	10 Gbps if RA-GRS/GRS is enabled, 15 Gbps for LRS/ZRS[2]
Maximum number of virtual network rules per storage account	200
Maximum number of IP address rules per storage account	200

[1]Azure Storage standard accounts support higher capacity limits and higher limits for ingress by request. To request an increase in account limits for ingress, contact Azure Support. For more information, see Announcing larger, higher scale storage accounts.

[2] If your storage account has read-access enabled with geo-redundant storage (RA-GRS) or geo-zone-redundant storage (RA-GZRS), then the egress targets for the secondary location are identical to those of the primary location. Azure Storage replication options include:

- Locally redundant storage (LRS)
- Zone-redundant storage (ZRS)

- Geo-redundant storage (GRS)
- Read-access geo-redundant storage (RA-GRS)
- Geo-zone-redundant storage (GZRS)
- Read-access geo-zone-redundant storage (RA-GZRS)

> **NOTE**
> Microsoft recommends that you use a general-purpose v2 storage account for most scenarios. You can easily upgrade a general-purpose v1 or an Azure Blob storage account to a general-purpose v2 account with no downtime and without the need to copy data. For more information, see Upgrade to a general-purpose v2 storage account.

If the needs of your application exceed the scalability targets of a single storage account, you can build your application to use multiple storage accounts. You can then partition your data objects across those storage accounts. For information on volume pricing, see Azure Storage pricing.

All storage accounts run on a flat network topology and support the scalability and performance targets outlined in this article, regardless of when they were created. For more information on the Azure Storage flat network architecture and on scalability, see Microsoft Azure Storage: A Highly Available Cloud Storage Service with Strong Consistency.

For more information on limits for standard storage accounts, see Scalability targets for standard storage accounts.

Storage resource provider limits

The following limits apply only when you perform management operations by using Azure Resource Manager with Azure Storage.

RESOURCE	DEFAULT LIMIT
Storage account management operations (read)	800 per 5 minutes
Storage account management operations (write)	1200 per hour
Storage account management operations (list)	100 per 5 minutes

Azure Blob storage limits

RESOURCE	TARGET
Maximum size of single blob container	Same as maximum storage account capacity
Maximum number of blocks in a block blob or append blob	50,000 blocks
Maximum size of a block in a block blob	100 MiB
Maximum size of a block blob	50,000 X 100 MiB (approximately 4.75 TiB)
Maximum size of a block in an append blob	4 MiB
Maximum size of an append blob	50,000 x 4 MiB (approximately 195 GiB)
Maximum size of a page blob	8 TiB
Maximum number of stored access policies per blob container	5

RESOURCE	TARGET
Target request rate for a single blob	Up to 500 requests per second
Target throughput for a single page blob	Up to 60 MiB per second
Target throughput for a single block blob	Up to storage account ingress/egress limits[1]

[1] Throughput for a single blob depends on several factors, including, but not limited to: concurrency, request size, performance tier, speed of source for uploads, and destination for downloads. To take advantage of the performance enhancements of high-throughput block blobs, upload larger blobs or blocks. Specifically, call the Put Blob or Put Block operation with a blob or block size that is greater than 4 MiB for standard storage accounts. For premium block blob or for Data Lake Storage Gen2 storage accounts, use a block or blob size that is greater than 256 KiB.

Azure Files limits

For more information on Azure Files limits, see Azure Files scalability and performance targets.

RESOURCE	STANDARD FILE SHARES	PREMIUM FILE SHARES
Minimum size of a file share	No minimum; pay as you go	100 GiB; provisioned
Maximum size of a file share	100 TiB*, 5 TiB	100 TiB
Maximum size of a file in a file share	1 TiB	1 TiB
Maximum number of files in a file share	No limit	No limit
Maximum IOPS per share	10,000 IOPS*, 1,000 IOPS	100,000 IOPS
Maximum number of stored access policies per file share	5	5
Target throughput for a single file share	up to 300 MiB/sec*, Up to 60 MiB/sec ,	See premium file share ingress and egress values
Maximum egress for a single file share	See standard file share target throughput	Up to 6,204 MiB/s
Maximum ingress for a single file share	See standard file share target throughput	Up to 4,136 MiB/s
Maximum open handles per file	2,000 open handles	2,000 open handles
Maximum number of share snapshots	200 share snapshots	200 share snapshots
Maximum object (directories and files) name length	2,048 characters	2,048 characters
Maximum pathname component (in the path \A\B\C\D, each letter is a component)	255 characters	255 characters

* Available in most regions, see Regional availability for the details on available regions.

Azure File Sync limits

RESOURCE	TARGET	HARD LIMIT
Storage Sync Services per region	20 Storage Sync Services	Yes
Sync groups per Storage Sync Service	100 sync groups	Yes
Registered servers per Storage Sync Service	99 servers	Yes
Cloud endpoints per sync group	1 cloud endpoint	Yes
Server endpoints per sync group	50 server endpoints	No
Server endpoints per server	30 server endpoints	Yes
File system objects (directories and files) per sync group	100 million objects	No
Maximum number of file system objects (directories and files) in a directory	5 million objects	Yes
Maximum object (directories and files) security descriptor size	64 KiB	Yes
File size	100 GiB	No
Minimum file size for a file to be tiered	V9: Based on file system cluster size (double file system cluster size). For example, if the file system cluster size is 4kb, the minimum file size will be 8kb. V8 and older: 64 KiB	Yes

NOTE

An Azure File Sync endpoint can scale up to the size of an Azure file share. If the Azure file share size limit is reached, sync will not be able to operate.

Azure Queue storage limits

RESOURCE	TARGET
Maximum size of a single queue	500 TiB
Maximum size of a message in a queue	64 KiB
Maximum number of stored access policies per queue	5
Maximum request rate per storage account	20,000 messages per second, which assumes a 1-KiB message size
Target throughput for a single queue (1-KiB messages)	Up to 2,000 messages per second

Azure Table storage limits

RESOURCE	TARGET
Maximum size of a single table	500 TiB
Maximum size of a table entity	1 MiB
Maximum number of properties in a table entity	255, which includes three system properties: PartitionKey, RowKey, and Timestamp
Maximum total size of property values in an entity	1 MiB
Maximum total size of an individual property in an entity	Varies by property type. For more information, see **Property Types** in Understanding the Table Service Data Model.
Maximum number of stored access policies per table	5
Maximum request rate per storage account	20,000 transactions per second, which assumes a 1-KiB entity size
Target throughput for a single table partition (1 KiB-entities)	Up to 2,000 entities per second

Virtual machine disk limits

You can attach a number of data disks to an Azure virtual machine. Based on the scalability and performance targets for a VM's data disks, you can determine the number and type of disk that you need to meet your performance and capacity requirements.

> **IMPORTANT**
> For optimal performance, limit the number of highly utilized disks attached to the virtual machine to avoid possible throttling. If all attached disks aren't highly utilized at the same time, the virtual machine can support a larger number of disks.

For Azure managed disks:

The following table illustrates the default and maximum limits of the number of resources per region per subscription. There is no limit for the number of Managed Disks, snapshots and images per resource group.

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Standard managed disks	50,000	50,000
Standard SSD managed disks	50,000	50,000
Premium managed disks	50,000	50,000
Standard_LRS snapshots	50,000	50,000
Standard_ZRS snapshots	50,000	50,000
Managed image	50,000	50,000

- **For Standard storage accounts:** A Standard storage account has a maximum total request rate of 20,000 IOPS. The total IOPS across all of your virtual machine disks in a Standard storage account should not

exceed this limit.

You can roughly calculate the number of highly utilized disks supported by a single Standard storage account based on the request rate limit. For example, for a Basic tier VM, the maximum number of highly utilized disks is about 66, which is 20,000/300 IOPS per disk. The maximum number of highly utilized disks for a Standard tier VM is about 40, which is 20,000/500 IOPS per disk.

- **For Premium storage accounts:** A Premium storage account has a maximum total throughput rate of 50 Gbps. The total throughput across all of your VM disks should not exceed this limit.

For more information, see Virtual machine sizes.

Managed virtual machine disks

Standard HDD managed disks

STANDARD DISK TYPE	S4	S6	S10	S15	S20	S30	S40	S50	S60	S70	S80
Disk size in GiB	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 1,300	Up to 2,000	Up to 2,000
Throughput per disk	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 300 MiB/sec	Up to 500 MiB/sec	Up to 500 MiB/sec

Standard SSD managed disks

STANDARD SSD SIZES	E1*	E2*	E3*	E4	E6	E10	E15	E20	E30	E40	E50	E60	E70	E80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	Up to 120	Up to 120	Up to 120	Up to 120	Up to 240	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 500	Up to 2,000	Up to 4,000	Up to 6,000
Throughput per disk	Up to 25 MiB/sec	Up to 25 MiB/sec	Up to 25 MiB/sec	Up to 25 MiB/sec	Up to 50 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 60 MiB/sec	Up to 400 MiB/sec	Up to 600 MiB/sec	Up to 750 MiB/sec

*Denotes a disk size that is currently in preview, for regional availability information see New disk sizes: Managed and unmanaged.

Premium SSD managed disks: Per-disk limits

PREMIUM SSD SIZES	P1*	P2*	P3*	P4	P6	P10	P15	P20	P30	P40	P50	P60	P70	P80
Disk size in GiB	4	8	16	32	64	128	256	512	1,024	2,048	4,096	8,192	16,384	32,767
IOPS per disk	120	120	120	120	240	500	1,100	2,300	5,000	7,500	7,500	16,000	18,000	20,000
Throughput per disk	25 MiB/sec	25 MiB/sec	25 MiB/sec	25 MiB/sec	50 MiB/sec	100 MiB/sec	125 MiB/sec	150 MiB/sec	200 MiB/sec	250 MiB/sec	250 MiB/sec	500 MiB/sec	750 MiB/sec	900 MiB/sec
Max burst IOPS per disk**	3,500	3,500	3,500	3,500	3,500	3,500	3,500	3,500						
Max burst throughput per disk**	170 MiB/sec	170 MiB/sec	170 MiB/sec	170 MiB/sec	170 MiB/sec	170 MiB/sec	170 MiB/sec	170 MiB/sec						
Max burst duration**	30 min	30 min	30 min	30 min	30 min	30 min	30 min	30 min						
Eligible for reservation	No	No	No	No	No	No	No	No	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year	Yes, up to one year

*Denotes a disk size that is currently in preview, for regional availability information see New disk sizes: Managed and unmanaged.

**Denotes a feature that is currently in preview, see Disk bursting for more information.

Premium SSD managed disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS Per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/s with GS5 VM

Unmanaged virtual machine disks

Standard unmanaged virtual machine disks: Per-disk limits

VM TIER	BASIC TIER VM	STANDARD TIER VM
Disk size	4,095 GB	4,095 GB
Maximum 8-KB IOPS per persistent disk	300	500
Maximum number of disks that perform the maximum IOPS	66	40

Premium unmanaged virtual machine disks: Per-account limits

RESOURCE	DEFAULT LIMIT
Total disk capacity per account	35 TB
Total snapshot capacity per account	10 TB
Maximum bandwidth per account (ingress + egress)[1]	<=50 Gbps

[1]*Ingress* refers to all data from requests that are sent to a storage account. *Egress* refers to all data from responses that are received from a storage account.

Premium unmanaged virtual machine disks: Per-disk limits

PREMIUM STORAGE DISK TYPE	P10	P20	P30	P40	P50
Disk size	128 GiB	512 GiB	1,024 GiB (1 TB)	2,048 GiB (2 TB)	4,095 GiB (4 TB)
Maximum IOPS per disk	500	2,300	5,000	7,500	7,500
Maximum throughput per disk	100 MB/sec	150 MB/sec	200 MB/sec	250 MB/sec	250 MB/sec
Maximum number of disks per storage account	280	70	35	17	8

Premium unmanaged virtual machine disks: Per-VM limits

RESOURCE	DEFAULT LIMIT
Maximum IOPS per VM	80,000 IOPS with GS5 VM
Maximum throughput per VM	2,000 MB/sec with GS5 VM

StorSimple System limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of storage account credentials	64	
Maximum number of volume containers	64	
Maximum number of volumes	255	
Maximum number of schedules per bandwidth template	168	A schedule for every hour, every day of the week.
Maximum size of a tiered volume on physical devices	64 TB for StorSimple 8100 and StorSimple 8600	StorSimple 8100 and StorSimple 8600 are physical devices.
Maximum size of a tiered volume on virtual devices in Azure	30 TB for StorSimple 8010 64 TB for StorSimple 8020	StorSimple 8010 and StorSimple 8020 are virtual devices in Azure that use Standard storage and Premium storage, respectively.
Maximum size of a locally pinned volume on physical devices	9 TB for StorSimple 8100 24 TB for StorSimple 8600	StorSimple 8100 and StorSimple 8600 are physical devices.
Maximum number of iSCSI connections	512	
Maximum number of iSCSI connections from initiators	512	
Maximum number of access control records per device	64	
Maximum number of volumes per backup policy	24	
Maximum number of backups retained per backup policy	64	
Maximum number of schedules per backup policy	10	
Maximum number of snapshots of any type that can be retained per volume	256	This amount includes local snapshots and cloud snapshots.
Maximum number of snapshots that can be present in any device	10,000	

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of volumes that can be processed in parallel for backup, restore, or clone	16	• If there are more than 16 volumes, they're processed sequentially as processing slots become available. • New backups of a cloned or a restored tiered volume can't occur until the operation is finished. For a local volume, backups are allowed after the volume is online.
Restore and clone recover time for tiered volumes	<2 minutes	• The volume is made available within 2 minutes of a restore or clone operation, regardless of the volume size. • The volume performance might initially be slower than normal as most of the data and metadata still resides in the cloud. Performance might increase as data flows from the cloud to the StorSimple device. • The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate might be affected by Internet bandwidth to the cloud. • The restore or clone operation is complete when all the metadata is on the device. • Backup operations can't be performed until the restore or clone operation is fully complete.

LIMIT IDENTIFIER	LIMIT	COMMENTS
Restore recover time for locally pinned volumes	<2 minutes	• The volume is made available within 2 minutes of the restore operation, regardless of the volume size. • The volume performance might initially be slower than normal as most of the data and metadata still resides in the cloud. Performance might increase as data flows from the cloud to the StorSimple device. • The total time to download metadata depends on the allocated volume size. Metadata is automatically brought into the device in the background at the rate of 5 minutes per TB of allocated volume data. This rate might be affected by Internet bandwidth to the cloud. • Unlike tiered volumes, if there are locally pinned volumes, the volume data is also downloaded locally on the device. The restore operation is complete when all the volume data has been brought to the device. • The restore operations might be long and the total time to complete the restore will depend on the size of the provisioned local volume, your Internet bandwidth, and the existing data on the device. Backup operations on the locally pinned volume are allowed while the restore operation is in progress.
Thin-restore availability	Last failover	
Maximum client read/write throughput, when served from the SSD tier*	920/720 MB/sec with a single 10-gigabit Ethernet network interface	Up to two times with MPIO and two network interfaces.
Maximum client read/write throughput, when served from the HDD tier*	120/250 MB/sec	
Maximum client read/write throughput, when served from the cloud tier*	11/41 MB/sec	Read throughput depends on clients generating and maintaining sufficient I/O queue depth.

*Maximum throughput per I/O type was measured with 100 percent read and 100 percent write scenarios. Actual throughput might be lower and depends on I/O mix and network conditions.

Stream Analytics limits

LIMIT IDENTIFIER	LIMIT	COMMENTS
Maximum number of streaming units per subscription per region	500	To request an increase in streaming units for your subscription beyond 500, contact Microsoft Support.
Maximum number of inputs per job	60	There's a hard limit of 60 inputs per Azure Stream Analytics job.
Maximum number of outputs per job	60	There's a hard limit of 60 outputs per Stream Analytics job.
Maximum number of functions per job	60	There's a hard limit of 60 functions per Stream Analytics job.
Maximum number of streaming units per job	192	There's a hard limit of 192 streaming units per Stream Analytics job.
Maximum number of jobs per region	1,500	Each subscription can have up to 1,500 jobs per geographical region.
Reference data blob MB	300	Reference data blobs can't be larger than 300 MB each.

Virtual Machines limits

Virtual Machines limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Virtual machines per cloud service[1]	50	50
Input endpoints per cloud service[2]	150	150

[1]Virtual machines created by using the classic deployment model instead of Azure Resource Manager are automatically stored in a cloud service. You can add more virtual machines to that cloud service for load balancing and availability.

[2]Input endpoints allow communications to a virtual machine from outside the virtual machine's cloud service. Virtual machines in the same cloud service or virtual network can automatically communicate with each other. For more information, see How to set up endpoints to a virtual machine.

Virtual Machines limits - Azure Resource Manager

The following limits apply when you use Azure Resource Manager and Azure resource groups.

RESOURCE	DEFAULT LIMIT
VMs per subscription	25,000[1] per region.
VM total cores per subscription	20[1] per region. Contact support to increase limit.
Azure Spot VM total cores per subscription	20[1] per region. Contact support to increase limit.
VM per series, such as Dv2 and F, cores per subscription	20[1] per region. Contact support to increase limit.

RESOURCE	DEFAULT LIMIT
Availability sets per subscription	2,000 per region.
Virtual machines per availability set	200
Certificates per subscription	Unlimited[2]

[1]Default limits vary by offer category type, such as Free Trial and Pay-As-You-Go, and by series, such as Dv2, F, and G. For example, the default for Enterprise Agreement subscriptions is 350.

[2]With Azure Resource Manager, certificates are stored in the Azure Key Vault. The number of certificates is unlimited for a subscription. There's a 1-MB limit of certificates per deployment, which consists of either a single VM or an availability set.

> **NOTE**
> Virtual machine cores have a regional total limit. They also have a limit for regional per-size series, such as Dv2 and F. These limits are separately enforced. For example, consider a subscription with a US East total VM core limit of 30, an A series core limit of 30, and a D series core limit of 30. This subscription can deploy 30 A1 VMs, or 30 D1 VMs, or a combination of the two not to exceed a total of 30 cores. An example of a combination is 10 A1 VMs and 20 D1 VMs.

Shared Image Gallery limits

There are limits, per subscription, for deploying resources using Shared Image Galleries:

- 100 shared image galleries, per subscription, per region
- 1,000 image definitions, per subscription, per region
- 10,000 image versions, per subscription, per region

Virtual machine scale sets limits

RESOURCE	DEFAULT LIMIT	MAXIMUM LIMIT
Maximum number of VMs in a scale set	1,000	1,000
Maximum number of VMs based on a custom VM image in a scale set	600	600
Maximum number of scale sets in a region	2,000	2,000

See also

- Understand Azure limits and increases
- Virtual machine and cloud service sizes for Azure
- Sizes for Azure Cloud Services
- Naming rules and restrictions for Azure resources