Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers

Article • 09/20/2021 • 11 contributors

Feedback

In this article

Applies to: Windows Server 2022, Windows Server 2019, Windows Server 2016

You can use this topic to learn how to configure DNS Policy to allow primary DNS servers to respond to DNS client queries based on the geographical location of both the client and the resource to which the client is attempting to connect, providing the client with the IP address of the closest resource.

> **Important**
>
> This scenario illustrates how to deploy DNS policy for geo-location based traffic management when you are using only primary DNS servers. You can also accomplish geo-location based traffic management when you have both primary and secondary DNS servers. If you have a primary-secondary deployment, first complete the steps in this topic, and then complete the steps that are provided in the topic **Use DNS Policy for Geo-Location Based Traffic Management with Primary-Secondary Deployments**.

With new DNS policies, you can create a DNS policy that allows the DNS server to respond to a client query asking for the IP address of a Web server. Instances of the Web server might be located in different datacenters at different physical locations. DNS can assess the client and Web server locations, then respond to the client request by providing the client with a Web server IP address for a Web server that is physically located closer to the client.

You can use the following DNS policy parameters to control the DNS server responses to queries from DNS clients.

- **Client Subnet**. Name of a predefined client subnet. Used to verify the subnet from which the query was sent.
- **Transport Protocol**. Transport protocol used in the query. Possible entries are **UDP** and **TCP**.
- **Internet Protocol**. Network protocol used in the query. Possible entries are **IPv4** and **IPv6**.
- **Server Interface IP address**. IP address of the network interface of the DNS server which received the DNS request.
- **FQDN**. The Fully Qualified Domain Name (FQDN) of the record in the query, with the possibility of using a wild card.
- **Query Type**. Type of record being queried (A, SRV, TXT, etc.).
- **Time of Day**. Time of day the query is received.

You can combine the following criteria with a logical operator (AND/OR) to formulate policy expressions. When these expressions match, the policies are expected to perform one of the following actions.

- **Ignore**. The DNS server silently drops the query.
- **Deny**. The DNS server responds that query with a failure response.
- **Allow**. The DNS server responds back with traffic managed response.

Geo-Location Based Traffic Management Example

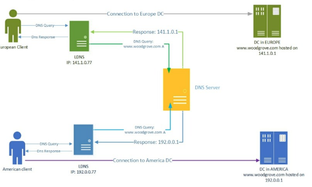
Following is an example of how you can use DNS policy to achieve traffic redirection on the basis of the physical location of the client that performs a DNS query.

This example uses two fictional companies - Contoso Cloud Services, which provides web and domain hosting solutions; and Woodgrove Food Services, which provides food delivery services in multiple cities across the globe, and which has a Web site named woodgrove.com.

Contoso Cloud Services has two datacenters, one in the U.S. and another in Europe. The European datacenter hosts a food ordering portal for woodgrove.com.

To ensure that woodgrove.com customers get a responsive experience from their website, Woodgrove wants European clients directed to the European datacenter and American clients directed to the U.S. datacenter. Customers located elsewhere in the world can be directed to either of the datacenters.

The following illustration depicts this scenario.



How the DNS name resolution process works

During the name resolution process, the user tries to connect to www.woodgrove.com. This results in a DNS name resolution request that is sent to the DNS server that is configured in the Network Connection properties on the user's computer. Typically, this is the DNS server provided by the local ISP acting as a caching resolver, and is referred as the LDNS.

If the DNS name is not present in the local cache of LDNS, the LDNS server forwards the query to the DNS server that is authoritative for woodgrove.com. The authoritative DNS server responds with the requested record (www.woodgrove.com) to the LDNS server, which in turn caches the record locally before sending it to the user's computer.

Because Contoso Cloud Services uses DNS Server policies, the authoritative DNS server that hosts contoso.com is configured to return geo-location based traffic managed responses. This results in the direction of European Clients to the European datacenter and the direction of American Clients to the U.S. datacenter, as depicted in the illustration.

In this scenario, the authoritative DNS server usually sees the name resolution request coming from the LDNS server and, very rarely, from the user's computer. Because of this, the source IP address in the name resolution request as seen by the authoritative DNS server is that of the LDNS server and not that of the user's computer. However, using the IP address of the LDNS server when you configure geo-location based query responses provides a fair estimate of the geo-location of the user, because the user is querying the DNS server of his local ISP.

> **Note**
>
> DNS policies utilize the sender IP in the UDP/TCP packet that contains the DNS query. If the query reaches the primary server through multiple resolver/LDNS hops, the policy will consider only the IP of the last resolver from which the DNS server receives the query.

How to configure DNS Policy for Geo-Location Based Query Responses

To configure DNS policy for geo-location based query responses, you must perform the following steps.

1. Create the DNS Client Subnets
2. Create the Scopes of the Zone
3. Add Records to the Zone Scopes
4. Create the Policies

> **Note**
>
> You must perform these steps on the DNS server that is authoritative for the zone you want to configure. Membership in **DnsAdmins**, or equivalent, is required to perform the following procedures.

The following sections provide detailed configuration instructions.

> **Important**
>
> The following sections include example Windows PowerShell commands that contain example values for many parameters. Ensure that you replace example values in these commands with values that are appropriate for your deployment before you run these commands.

Create the DNS Client Subnets

The first step is to identify the subnets or IP address space of the regions for which you want to redirect traffic. For example, if you want to redirect traffic for the U.S. and Europe, you need to identify the subnets or IP address spaces of these regions.

You can obtain this information from Geo-IP maps. Based on these Geo-IP distributions, you must create the "DNS Client Subnets." A DNS Client Subnet is a logical grouping of IPv4 or IPv6 subnets from which queries are sent to a DNS server.

You can use the following Windows PowerShell commands to create DNS Client Subnets.

```
PowerShell
Add-DnsServerClientSubnet -Name "USSubnet" -IPv4Subnet "192.0.0.0/24"
Add-DnsServerClientSubnet -Name "EuropeSubnet" -IPv4Subnet "141.1.0.0/24"
```

For more information, see Add-DnsServerClientSubnet.

Create Zone Scopes

After the client subnets are configured, you must partition the zone whose traffic you want to redirect into two different zone scopes, one scope for each of the DNS Client Subnets that you have configured.

For example, if you want to redirect traffic for the DNS name www.woodgrove.com, you must create two different zone scopes in the woodgrove.com zone, one for the U.S. and one for Europe.

A zone scope is a unique instance of the zone. A DNS zone can have multiple zone scopes, with each zone scope containing its own set of DNS records. The same record can be present in multiple scopes, with different IP addresses or the same IP addresses.

> **Note**
>
> By default, a zone scope exists on the DNS zones. This zone scope has the same name as the zone and legacy DNS operations work on this scope.

You can use the following Windows PowerShell commands to create zone scopes.

```
PowerShell
Add-DnsServerZoneScope -ZoneName "woodgrove.com" -Name "USZoneScope"
Add-DnsServerZoneScope -ZoneName "woodgrove.com" -Name "EuropeZoneScope"
```

For more information, see Add-DnsServerZoneScope.

Add Records to the Zone Scopes

Now you must add the records representing the web server host into the two zone scopes.

For example, **USZoneScope** and **EuropeZoneScope**. In USZoneScope, you can add the record www.woodgrove.com with the IP address 192.0.0.1, which is located in a U.S. datacenter; and in EuropeZoneScope you can add the same record (www.woodgrove.com) with the IP address 141.1.0.1 in the European datacenter.

You can use the following Windows PowerShell commands to add records to the zone scopes.

```
PowerShell
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "192.0.0.1" -ZoneSc
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "141.1.0.1" -ZoneSco
```

In this example, you must also use the following Windows PowerShell commands to add records into the default zone scope to ensure that the rest of the world can still access the woodgrove.com web server from either of the two datacenters.

```
PowerShell
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "192.0.0.1"
Add-DnsServerResourceRecord -ZoneName "woodgrove.com" -A -Name "www" -IPv4Address "141.1.0.1"
```

The **ZoneScope** parameter is not included when you add a record in the default scope. This is the same as adding records to a standard DNS zone.

For more information, see Add-DnsServerResourceRecord.

Create the Policies

After you have created the subnets, the partitions (zone scopes), and you have added records, you must create policies that connect the subnets and partitions, so that when a query comes from a source in one of the DNS client subnets, the query response is returned from the correct scope of the zone. No policies are required for mapping the default zone scope.

You can use the following Windows PowerShell commands to create a DNS policy that links the DNS Client Subnets and the zone scopes.

```
PowerShell
Add-DnsServerQueryResolutionPolicy -Name "USPolicy" -Action ALLOW -ClientSubnet "eq,USSubnet" -ZoneSc
Add-DnsServerQueryResolutionPolicy -Name "EuropePolicy" -Action ALLOW -ClientSubnet "eq,EuropeSubnet"
```

For more information, see Add-DnsServerQueryResolutionPolicy.

Now the DNS server is configured with the required DNS policies to redirect traffic based on geo-location.

When the DNS server receives name resolution queries, the DNS server evaluates the fields in the DNS request against the configured DNS policies. If the source IP address in the name resolution request matches any of the policies, the associated zone scope is used to respond to the query, and the user is directed to the resource that is geographically closest to them.

You can create thousands of DNS policies according to your traffic management requirements, and all new policies are applied dynamically - without restarting the DNS server - on incoming queries.

Feedback

Submit and view feedback for

This product | This page

View all page feedback