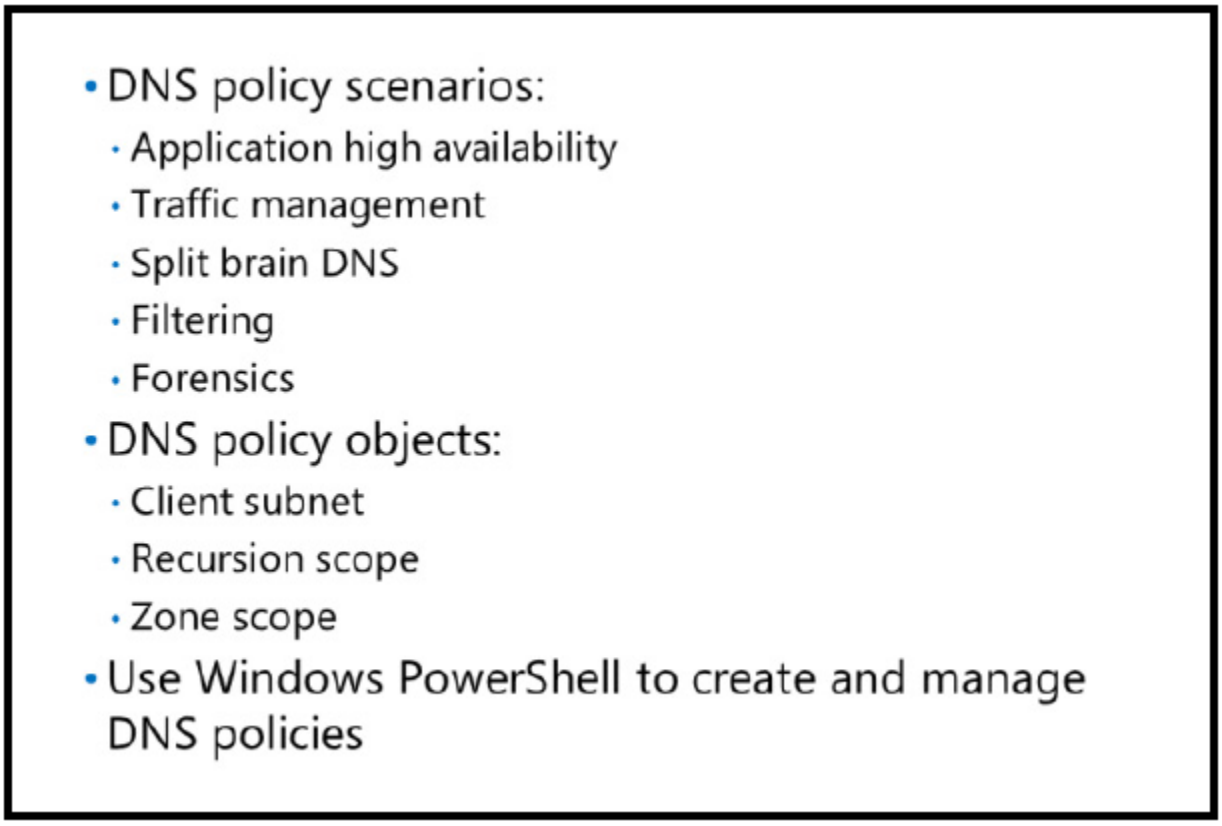
DNS policies



DNS Policy is a new feature for DNS in Windows Server 2016. You use DNS policies to manipulate how a DNS server handles queries based on different factors. As an example, you might create a DNS policy to respond to queries asking for the IP address of a web server to respond with a different IP address based on the closest datacenter to the client. This differs from netmask reordering because the client will not have the same local subnet address as the web server, but the particular web server is closer than others, from the perspective of the client.

Scenarios for using DNS policies

You can create several DNS policies depending on your needs. There are various factors that might benefit from creating a DNS policy, based on the following scenarios:

- Application high availability. Clients are redirected to the healthiest endpoint for an application, where healthiest is determined by high availability factors in a failover cluster.
- Traffic management. Clients are redirected to the closest datacenter or server location.
- Split-brain DNS. Clients receive a response based on whether they are internal or external, and the DNS records are split into different zone scopes.
- Filtering. DNS queries are blocked if they are from a list of malicious IP addresses or FQDNs.
- Forensics. Malicious DNS clients are redirected to a sinkhole instead of the computer they are

trying to reach.

Note: DNS sinkholes, sometimes referred to as black hole DNS, are used to spoof DNS servers to prevent resolving host names of specified Uniform Resource Locators (URLs). You can configure the DNS forwarder to return a false IP address to a specific URL. You can use a DNS sinkhole to prevent access to malicious URLs at the enterprise level. The malicious URLs are blocked by adding a false resource record in DNS, thereby creating a second level of protection.

Additional Reading: For more information on DNS sinkholes, refer to: "Applying Filters on DNS Queries using Windows DNS Server Policies" at: http://aka.ms/Efxdlc

- Time-of-day based redirection. Clients are redirected to datacenters based on the time of the day.

DNS policy objects

To use the above scenarios to create policies, you must identify groups of records in a zone, groups of clients on a network, or other elements. The elements are identified by the following new DNS objects:

- Client subnet. This represents the IPv4 or IPv6 subnet from which queries are sent to a DNS server. You create subnets to later define policies that you apply based on the subnet that generates the requests. For example, you might have a split-brain DNS scenario, where the name resolution request for *www.contoso.com* can be answered with an internal IP address to internal clients, and a different IP address to external clients.
- Recursion scope. This represents unique instances of a group of settings that control DNS server recursion. A recursion scope holds a list of forwarders and specifies whether recursion is used. A DNS server can have multiple recursion scopes. You can use DNS server recursion policies to choose a recursion scope for a given set of queries. If the DNS server is not authoritative for certain queries, DNS server recursion policies let you control how to resolve those queries. In this case, you can specify which forwarders to use and whether to use recursion.
- Zone scopes. DNS zones can have multiple zone scopes, and each zone scope can contain its own set of DNS resource records. The same resource record can be present across multiple scopes, with different IP addresses depending on the scope. Additionally, zone transfers can be done at the zone-scope level. This will allow resource records from a zone scope in a primary zone to be transferred to the same zone scope in a secondary zone.

Create and manage DNS policies

You create DNS policies based on level and type. You can use query-resolution policies to define how client name resolution queries get handled, and zone-transfer policies to define zone transfers. You can apply both policy types at the server or zone level.

You can create multiple query resolution policies of the same level, if they have a different value for the processing order. Recursion policies are a special kind of server-level policies. They control how a DNS server performs query recursion, if at all. Recursion policies only apply when query processing reaches the recursion path. You can choose a value of **DENY** or **IGNORE** for recursion for a given set of queries. Otherwise, you can choose a set of forwarders for a set of queries.

You use Windows PowerShell version 5.0 or higher to create and manage DNS policies. The following example shows how to create traffic management policies to direct the client name resolution requests from a certain subnet to an Asian datacenter, and from another subnet to an Australian datacenter:

```
Add-DnsServerClientSubnet -Name "AsiaSubnet" -IPv4Subnet "172.21.33.0/24"
Add-DnsServerClientSubnet -Name "AustraliaSubnet" -IPv4Subnet
"172.17.44.0/24"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "AsiaZoneScope"
Add-DnsServerZoneScope -ZoneName "Contoso.com" -Name "AustraliaZoneScope"
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -
IPv4Address
"172.17.97.97" -ZoneScope "AustraliaZoneScope"
Add-DnsServerResourceRecord -ZoneName "Contoso.com" -A -Name "www" -
IPv4Address
"172.21.21.21" -ZoneScope "AsiaZoneScope"
Add-DnsServerQueryResolutionPolicy -Name "AsiaPolicy" -Action ALLOW -
ClientSubnet
"eq,AsiaSubnet" -ZoneScope "AsiaZoneScope,1" -ZoneName "Contoso.com"
Add-DnsServerQueryResolutionPolicy -Name "AustraliaPolicy" -Action ALLOW -
ClientSubnet
"eq,AustraliaSubnet" -ZoneScope "AustraliaZoneScope,1" -ZoneName contoso.com
```