Technical Report

DNS Load Balancing in ONTAP

Configuration and Best Practices

Justin Parisi, NetApp
February 2021 | TR-4523

Abstract

This document explains how to configure NetApp® storage systems with NetApp ONTAP® management software for use with DNS load balancing methodologies. This document covers the on-box DNS feature available in ONTAP, various configuration methods, and best practices.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF BEST PRACTICES

Domain name systems in ONTAP

ONTAP enables storage administrators to present multiple logical interfaces (data LIFs) per storage virtual machine (SVM) across multiple nodes to clients for NAS access. In NAS environments, clusters can have up to 24 nodes, so the number of potential data LIFs in a cluster is large. This scale can create confusion about access for clients if they rely on mounting through IP addresses, because end users are not expected to understand where an IP address resides in the storage system. Clients can overload a node with requests if they continuously mount the same data LIFs and attempting to remember specific IP addresses can be challenging.

Management of these IP addresses can also be challenging. If clients are accessing a known IP by the address, then an administrator must make clients explicitly aware of changes if they add or remove data LIFs.

To simplify client access to these data LIFs as well as the management of the NAS networking components from the storage side, the Domain Name System (DNS) is often implemented to shield multiple data LIFs behind a single host name.

For general name service best practices in ONTAP, see TR-4668: Name Services Best Practice Guide.

The following Requests for Comments (RFCs) cover DNS standards and provide general information about DNS:

- RFC 1035 – Domain Names
- RFC 1123 – Requirements for Internet Hosts
- RFC 2181 – Clarifications to the DNS Specification

What is DNS?

DNS is a hierarchical naming system for devices on a network that provides a way to associate human-readable names to less readily memorized items, such as IP addresses, service records, and so on. DNS relegates the issuance of these records to one or more servers that act as authoritative sources on the network.

DNS terminology

The following section covers different types of DNS terminology used with on-box DNS.

A/AAAA records

A/AAAA records (RFC-1101) map host names to IP addresses. An A record maps a host name to an IPv4 address. An AAAA record maps host names to IPv6 addresses. These maps are used for forward DNS lookups.

Canonical name

Canonical name (CNAME) is an alias of a host name.

Service records

Service (SRV) records (RFC-2782) define a DNS record for a specific domain service, including LDAP, CIFS, NFS, Exchange, and so on. These records can point to multiple A/AAAA records to provide round-robin load balancing and high availability.

Pointer records

Pointer (PTR) records map IP addresses to canonical names. This mapping is used for reverse DNS lookups.

Name server records

Name server (NS) NS records are used to delegate a subdomain to a set of name servers. These records can be authoritative or nonauthoritative records.

Start of Authority records

This type of record defines which name server is the authoritative answer for a DNS request. If a name server that does not have a State of Authority (SOA) record issues a response to a DNS request, the response returns to the client as a “nonauthoritative” response.

SOA records contain the following information:

- Primary name server from the DNS domain
- Time stamp of updates
- Zone refresh time
- Failed refresh retry times
- SOA record timeout
- Negative time to live (TTL) (how long failed resolvers live in failure cache)

DNS forwarder

A DNS forwarder is a DNS server on a network that forwards DNS queries for external DNS names to DNS servers outside that network. You can also forward queries according to specific domain names by using conditional forwarders, which override regular DNS forwarders.

Conditional forwarder

A conditional forwarder is a DNS server on a network that forwards DNS queries according to the DNS domain name in the query. For example, a DNS server can be configured to forward all the queries it receives for names ending with `example.newname.com` to the IP address of a specific DNS server or to the IP addresses of multiple DNS servers. A conditional forwarder is used when a DNS server’s domain differs from the desired DNS domain name.

For example:

```
example.newname.com → netapp.com
```

A conditional forwarder requires the data LIFs to be added to DNS as name servers and to have an SOA record. In addition, a forward lookup zone and reverse lookup entries must be created. Windows 2008 and later might require SOA records. Windows 2003 DNS does not require SOA records.

Stub zones

From the Microsoft article on stub zones:

A stub zone is a copy of a zone that contains only those resource records necessary to identify the authoritative Domain Name System (DNS) servers for that zone. A stub zone is used to resolve names between separate DNS namespaces. This type of resolution might be necessary when a corporate merger requires that the DNS servers for two separate DNS namespaces resolve names for clients in both namespaces.

A stub zone consists of the following:

- The SOA resource record, name server (NS) resource records, and the glue A resource records for the delegated zone.
- The IP address of one or more master servers that can be used to update the stub zone.

The master servers for a stub zone are one or more DNS servers authoritative for the child zone, usually the DNS server hosting the primary zone for the delegated domain name.

A stub zone is required if conditional forwarding does not work because the name servers are not SOA servers, and the DNS zone created is not a stub zone.

For a comparison of stub zones and conditional forwarders, see the Microsoft article on Contrasting stub zones and conditional forwarders.

Primary zones

A primary zone is a DNS zone that is the primary source of information for a zone and that stores a master copy of zone data in a local file or in the database. Unlike stub zones, primary zones allow the creation of records (A, AAAA, SRV, and so on).

DNS delegations

A DNS delegation delegates requests in the same domain to the DNS servers specified in the delegation zone. For example, use a delegation for `cdot.netapp.com` in the DNS domain of `netapp.com`.

For more information on zone delegations, see the Microsoft article on delegating zones and understanding zone delegation.

Subdomains

A subdomain is a DNS domain that is part of the primary DNS domain. For example, `dns.domain.com` is a subdomain of `domain.com`.

DNS options in ONTAP

NetApp ONTAP offers a variety of options for controlling DNS configurations, including the following:

- Dynamic DNS (IPv4 and IPv6)
- On-box DNS load balancing
- The ability to use data LIFs as name servers and/or name records

The following DNS configuration options are available with advanced privilege in ONTAP 9.7 and later:

```
cluster::*> dns ?
  check                       Display validation status of a DNS configuration
  create                      Create a new DNS table entry
  delete                      Remove a DNS table entry
  dynamic-update>             Manage Dynamic DNS Updates
  hosts>                      Manage local mapping for host names
  modify                      Change a DNS table entry
  show                        Display DNS configuration
```

Testing DNS lookups in ONTAP

ONTAP provides commands with advanced privilege to run forward and SRV record lookups against DNS servers. This allows storage administrators to see if DNS is functioning properly.

To test forward lookups, run the following command:

```
cluster::*> access-check dns forward-lookup -vserver SVM -hostname {hostname|fqdn}
```

To test SRV lookups, run the following command:

```
cluster::*> access-check dns srv-lookup -vserver SVM -lookup-string {_ldap._tcp.ntap.local}
```

If netgroups are being used, you can also check DNS by running the `getXXbyYY netgrpcheck` command with advanced privilege.

```
cluster::*> getxxbyyy netgrpcheck -node node1 -vserver DEMO -netgroup netgroup1 -clientIP
10.193.67.225

Client 10.193.67.225 is not a member of netgroup netgroup1
Searched using NETGROUP_BYHOST
```

DNS caching in ONTAP

ONTAP provides caching of DNS hostnames and IP addresses that help reduce load to DNS servers and speed up name resolution requests. These caches are managed with the `name-service cache hosts` command with advanced privilege.

The following table shows the default values for the cache settings.

Table 1) DNS cache settings in ONTAP.

Setting	Value
DNS cache enabled	True
Negative cache enabled	True
Time to live (TTL)	24 hours
Negative time to live (TTL)	1 minute
Is time to live (TTL) taken from DNS	True

Note: Negative cache entries expire sooner to prevent long-term caching of failed DNS attempts.

Viewing/managing DNS hosts caches

Caches can be viewed and manually cleared for host entries using the same the `name-service cache hosts` command sets with advanced privilege. Caches are populated by export access, netgroup checks, and from `access-check dns` commands**.**

To view a DNS cache entry, run the following command:

```
cluster::*> name-service cache hosts forward-lookup show -vserver DEMO -host centos7 -instance

                    IP       Address IP                    Create
Vserver   Host      Protocol Family  Address        Source  Time       TTL(sec)
--------- --------- -------- ------- -------------- ------- ---------- --------
DEMO      centos7   Any      Any     10.193.67.225  dns     4/29/2020  3600
                                                             11:15:49

cluster::*> name-service cache hosts reverse-lookup show -vserver DEMO -ip 10.193.67.225

Vserver    IP Address      Host                Source Create Time     TTL(sec)
---------- --------------- ------------------- ------ -------------- --------
DEMO       10.193.67.225   centos7.ntap.local  dns    4/29/2020 11:25:58 3600
```

To clear a DNS cache entry, run the following command:

```
cluster::*> name-service cache hosts forward-lookup

  delete                      *Delete an entry
  delete-all                  *Delete all the entries for the vserver
```

DNS load balancing

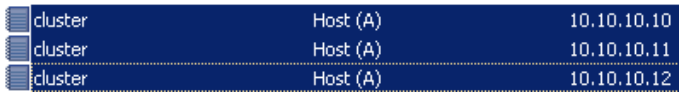
An added benefit of using DNS host names to point to multiple IP addresses is having the ability to leverage various load balancing mechanisms with DNS servers. DNS load balancing is a way to distribute client requests for host names across multiple IP addresses without needing client interaction. Generally, DNS load balancing is performed by round-robin. Load balancing can also be performed through third-party load balancers or through the ONTAP feature known as on-box DNS load balancing.

Off-box round-robin DNS

Round-robin DNS is the most common form of DNS load balancing. It is offered by default in DNS servers and is a simple way to offer IP addresses to the clients requesting them.

To create a round-robin A/AAAA record, create another A/AAAA record with the same name as the original record.

Figure 1) Example of off-box DNS round-robin method using A records.



- For more information on round-robin DNS in Windows, see Configuring Round-Robin DNS in Windows.
- For more information on round-robin DNS in BIND, see Round-Robin Load Distribution.

Round-robin DNS limitations

Round-robin balancing does not take into account server load, network connectivity, and so on. It simply serves up IP addresses in the order of request received. If a server or client IP address experiences issues in a round-robin configuration, the DNS server might still issue the IP address for the problematic server, which can create issues for clients. Because of this possibility, round-robin DNS might not be an ideal method for enterprise NAS environments because these environments might require a more discerning load balancing methodology. Fortunately, ONTAP offers an integrated, simple, and intelligent load balancing solution for DNS free of charge—no license is required.

On-box DNS load balancing

ONTAP allows you to use the DNS service on each node to service DNS requests from clients. ONTAP can also issue data LIF IP addresses based on an algorithm that evaluates CPU load and port throughput on the node. This process presents the least-used data LIF to make sure of proper load balancing across the cluster for mount requests. After a mount or map is successful, the client continues to use that connection until remount.

This approach differs from round-robin DNS because the external DNS server services all requests and has no insight into how busy a node in the cluster is.

On-box DNS considerations

Use of DNS load balancing is not necessary when using NFSv4.x referrals or SMB auto-location, because the initial connection is made to the node local to the volume being accessed regardless of which IP address was returned from DNS.

Using on-box DNS with FlexGroup volumes can offer some benefits, even though the volume can span multiple nodes in a cluster, because the TCP connections being established can be spread across multiple nodes.

Additionally, round-robin DNS issues IP addresses with a time to live (TTL). The TTL caches the DNS request in Windows for 24 hours by default. On-box DNS issues a TTL of 0, which means that DNS is never cached on the client and a new IP is always issued based on load. Clients do not set the TTL; the DNS server defines it. In this case, ONTAP is acting as a DNS server.

If you want to see what is in the Windows DNS cache, use ipconfig /displaydns.

On-box DNS interaction with pNFS

On-box DNS does not apply to pNFS data traffic, which redirects traffic for I/O consistently during mounts. However, on-box DNS can assist in load balancing connections to the metadata servers (MDS) in the cluster. For more information about pNFS, see TR-4067: NFS Best Practices and Implementation Guide and TR-4063: Parallel Network File System Configuration and Best Practices for Clustered Data ONTAP.

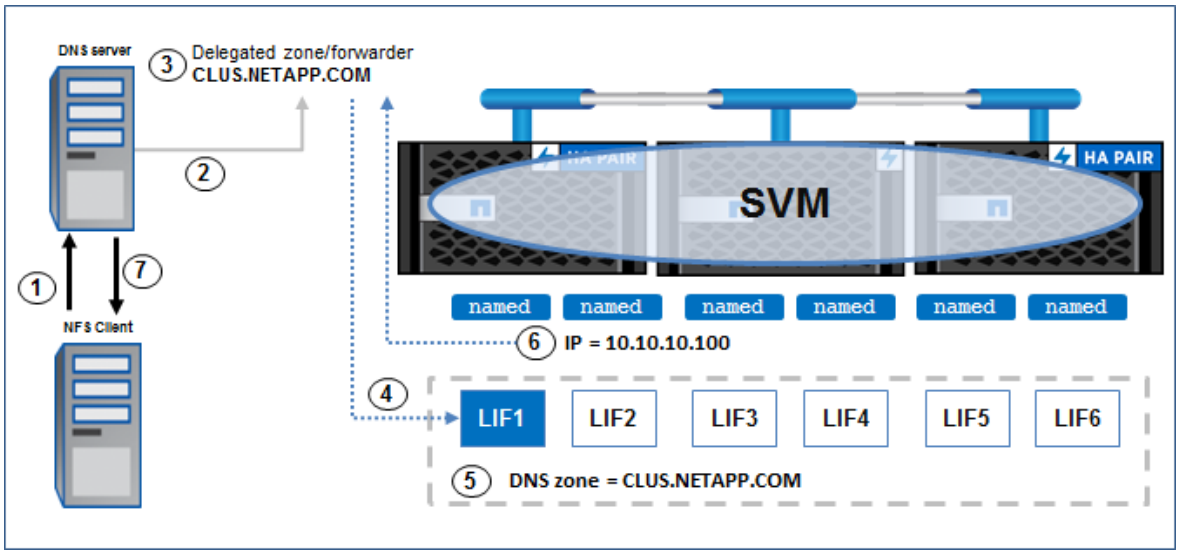
Best Practice 1: ONTAP Version Recommendation: On-Box DNS

When using on-box DNS, use the most recent patched release of ONTAP for best results.

How on-box DNS Load balancing works

Each node in the cluster has a service running (named) that handles incoming DNS requests from clients. The node also issues IP addresses based on a calculated weight that is determined with an algorithm based on CPU utilization and node throughput.

Figure 2) On-box DNS load balance example.



When a client attempts to access the cluster by DNS host name, the following process takes place:

1. The client issues a DNS request and uses the DNS server specified in its configuration.
2. The DNS server looks for the host name in the request.
3. When using on-box DNS, the host name is either a DNS delegation or a conditional forwarder. The record contains a list of data LIF IP addresses (presented as NS records) to use for DNS requests.
4. The request is forwarded or delegated to one of the data LIF IP addresses on a round-robin basis.
5. The data LIF receives the request if the LIF has the DNS zone configured and is set to listen for DNS queries (which opens port 53 on the LIF).
6. The node receiving the request checks the DNS weights for each node and issues an IP address based on the calculated load.

7. The IP address is returned to the DNS server, which then returns the IP address to the client.

Note: In ONTAP versions earlier than 8.2, on-box DNS load balancing does not work with ifgrps or VLANs. For implementations that have those configurations, use external round-robin DNS. ONTAP versions 8.2 and later allow on-box DNS load balancing on ifgrps and VLANs.

The on-box DNS algorithm

The ONTAP on-box DNS algorithm is covered in patent number US8271652. You can find complete details at the patent link. See the following abstract from that patent:

"DNS name resolution is integrated into each node in a network storage cluster, to allow load balancing of network addresses, using a weighted random distribution to resolve DNS requests. A node in the cluster gathers statistics on utilization of resources, such as CPU utilization and throughput, on nodes in the cluster and distributes those statistics to all other nodes. Each node uses the same algorithm to generate weights for the various IP addresses of the cluster, based on the statistics distributed to it. The weights are used to generate a weighted list of available network addresses. In response to a DNS request, a DNS in a given node randomly indexes into the weighted address list to resolve requests to a network address. The weights are chosen so that the DNS is likely to pick an IP address which has a low load, to balance port and node usage over time."

The algorithm incorporates a series of weights assigned to data LIFs participating in the DNS load balancing group. These weights are refreshed every minute and use CPU weight and throughput weight to calculate a final weight. The on-box DNS algorithm calculations are as follows:

- **CPU weight**

  cpu_weight=100.0−(% of CPU being used)/number of IP addresses on node where IP address resides)
- **Throughput weight**

  thpt_weight=100.0−(% of port throughput being used)/number of IP addresses on port where IP address resides
- **Final weight**

  final_weight=(thpt_weight+cpu_weight)/2

Geometric mean versus arithmetic mean

In ONTAP versions before the fix for bug 619247, the DNS load balance algorithm used an arithmetic mean rather than a geometric mean. The arithmetic mean was known to return IP addresses for nodes with low throughput and 100% CPU utilization, so it was changed. Current versions of ONTAP use the geometric mean by default. NetApp does not recommend that you change this option.

Best Practice 2: Geometric Mean Configuration

Do not modify the geometric mean for load balancing unless directed by NetApp Technical Support.

This behavior is controlled through a CLI option in advanced privilege mode:

```
cluster::*> network options load-balancing show
Geometric Mean Algorithm for load balancing: true
```

On-box DNS tested limits

Recent load testing of on-box DNS was performed to determine how many concurrent DNS requests per second, per node can be handled by ONTAP.

Table 2) Maximum DNS requests per second, per node – ONTAP 9.7.

DNS Over TCP	DNS Over UDP
4,000 – 5,000	1,000 – 1,500

Note: Additional details of the testing can be found in internal bug 1285445.

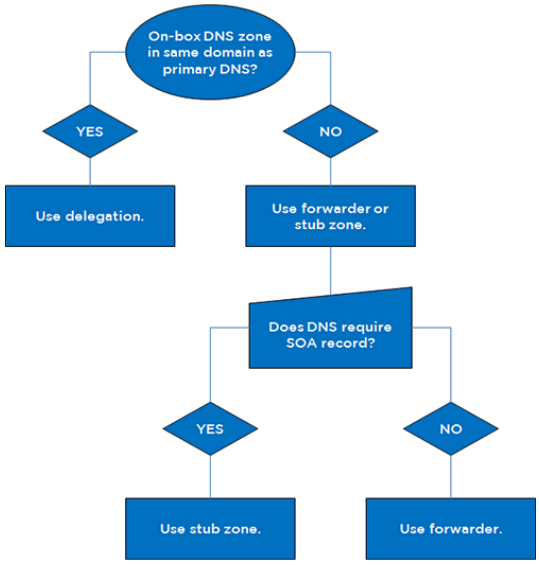
Deciding how to configure the on-box DNS zone

This section covers how to decide which DNS zone methodology to use to configure on-box DNS load balancing.

Note: The same concepts apply to non-Windows DNS servers (such as BIND). DNS is an Internet standard, as covered in RFC 1035.

Deciding Which Configuration to Use in Windows DNS

Figure 3) Factors to consider in setting up on-box DNS load balancing on Windows DNS servers.



When configuring on-box DNS load balancing, you need to make a design decision about whether to use conditional forwarding, a stub zone, or a DNS zone delegation. This blog covers use-case scenarios for when to use which type of forwarding zone.

The design decision is based on a variety of factors, as shown in the figure pictured to the left.

In some cases, it might make sense to configure clients to reference the data LIFs acting as DNS listeners directly as name servers. For guidance on doing so, see the section “Configuring clients to use ONTAP data LIFs as DNS .”

Best Practice 3: Windows DNS Configuration Recommendations

Use the following guidance to decide which type of DNS zone to use with Windows DNS servers.

- For data LIFs named with a DNS zone in the same domain as the primary DNS server, use DNS delegations.
- For data LIFs named with a DNS zone in a different DNS domain than the primary DNS server, use a stub zone unless SOA records are not required. In those cases, use forwarders.

Deciding Which Configuration to Use with BIND DNS

When configuring on-box DNS load balancing, you must make a design decision about whether to use forwarding, a subdomain zone, or a DNS zone delegation.

Best Practice 4: BIND DNS Configuration Recommendations

Use the following guidance to decide which type of DNS zone to use with BIND.

- Use forwarders if you do not use caching name servers and allow recursive requests.

- Ideally, use a zone delegation if the DNS domain is not a child domain. Delegations allow you to specify SOA and NS records, whereas forwarders do not. Additionally, delegations can be replicated to slave DNS servers automatically with BIND zone files, while forwarders are manually added to `named.conf`.
- If the DNS domain is a child domain, use subdomains.

Note: If you are using BIND9 DNS servers with on-box DNS, be sure to run ONTAP 8.2.3 or later because of bug 892388.

Using on-box DNS with data LIFs in different subnets and networks

In ONTAP, it is possible to have a configuration in which DNS servers live in a different physical or virtually segmented network or IP space than the data LIFs to which clients connect. With this configuration, you can still use on-box DNS to serve the desired data LIFs to clients.

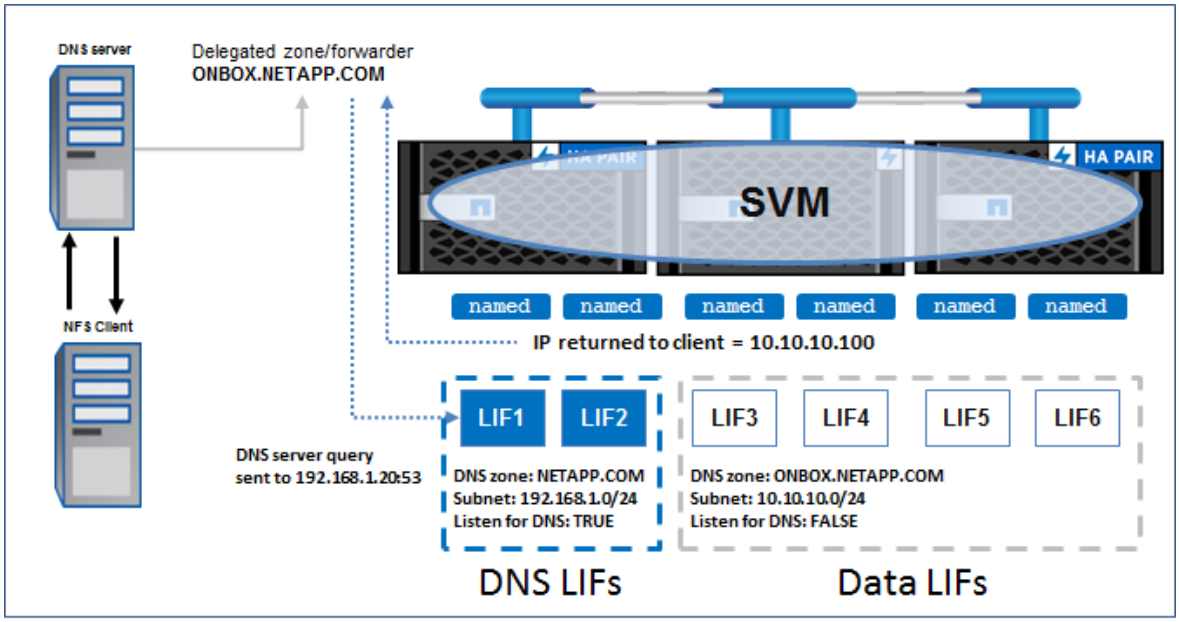
To do so, configure the LIFs that can communicate with the DNS servers to listen for DNS queries. The data LIFs that participate in the DNS zone should be configured to use the desired DNS zone and not listen for DNS queries (`-listen-for-dns-query false`).

Doing so enables the DNS server to communicate to the SVM using the DNS LIFs. It also enables the server to return a list of IP addresses to clients that might not be able to communicate with it.

Note: A data LIF that has `-listen-for-dns-query` set to "true" must also have a `-dns-zone` specified; otherwise, the cluster does not allow that LIF to listen for DNS queries.

Figure 4 illustrates a similar configuration.

Figure 4) On-box DNS with multiple subnets in same SVM.



The data LIF configuration looks like the following example. The data LIF called data1 can communicate with the DNS servers; the data LIF called dns-zone cannot:

```
cluster::*> net int show -vserver SVM -fields dns-zone,listen-for-dns-query,address
  (network interface show)
vserver lif      address      dns-zone                 listen-for-dns-query
------- -----    ------------ ------------------------ --------------------
SVM     data1    10.63.57.237 domain.netapp.com        true
SVM     dns-zone 10.10.10.200 onbox.domain.netapp.com  false
```

Enabling on-box DNS on data LIFs in ONTAP

For a data LIF to serve DNS queries, the `-listen-for-dns-query` option must be set to "true." For the SVM to return data LIFs in DNS queries, the desired data LIFs participating in the DNS zone must be assigned the DNS zone with `-dns-zone`. Any data LIF that acts as an SOA for DNS queries must have network connectivity to the DNS servers to which the clients point. This can be done nondisruptively.

Best Practice 5: Recommendations for Data LIFs Acting as DNS Servers

As a best practice, configure multiple data LIFs as DNS servers if at all possible to ensure resiliency and load balancing of DNS requests. It also makes sense to set the `lb-weight` for LIFs serving DNS requests to 0 so that they do not get used in the DNS zone for data traffic.

The data LIFs that participate in on-box DNS load balancing depend on the configuration of the network interface options described in Table 3.

Table 3) Data LIF options for on-box DNS load balancing in ONTAP.

Network Interface Option	What It Does	Privilege Level
`-dns-zone`	Specifies the DNS zone of the data LIF participating in the on-box DNS load balance operation. Multiple DNS zones can be specified in an SVM.	Admin
`-listen-for-dns-query`	Specifies that the data LIF will listen for DNS queries on port 53 and act as an SOA.	Admin
`-lb-weight`	Use this parameter to modify the load balancing weight of the data LIF. A valid load balancing weight is any integer between 1 and 100 or the word "load." If you specify the same load balancing weight for all data LIFs in a DNS zone, client requests are uniformly distributed, in a manner similar to round-robin DNS. A data LIF with a low load balancing weight is made available for client requests less frequently than one that has a high load balancing weight.	Advanced

It is possible to designate only specific data LIFs in a DNS zone to participate as the name servers through the `listen-for-dns-query` option while leaving other data LIFs to be used only for data traffic in the DNS zone. It is also possible to have data LIFs in the same SVM that do not participate in the on-box DNS load balancing zone but still can serve data traffic.

Manually modifying the lb-weight of data LIFs participating in on-box DNS

If multiple data LIFs are used in on-box DNS load balancing, it is possible to modify the `lb-weight` of specific data LIFs to be featured sooner in the load balancing algorithm. One use case for this is to favor nodes in a cluster using SSDs or All Flash FAS (AFF) systems in the weighting of data LIFs rather than nodes using spinning disks or favoring nodes with more RAM and CPU.

For example, if a four-node cluster has an HA pair with A800s and two nodes are FAS9xxx nodes with SAS shelves, it might make sense to configure the data LIFs owned by the AFF nodes to have higher weights than the nodes with SAS shelves. Doing so would take advantage of the enhanced performance capabilities of the AFF systems.

Consider the following guidelines when setting the LIF weights:

- Setting a LIF to a weight of 100 means that the data LIF is almost always used in DNS requests.
- Setting a LIF to a weight of 1 means that a data LIF is almost never used in DNS requests.
- If all `lb-weights` are the same, round-robin DNS is used.

- Keep in mind how the on-box DNS load balancing algorithm works when deciding whether to manually configure the `lb-weights` of data LIFs.

Configuring ONTAP to enable or disable sending of SOA records

In some cases, such as with non-Windows DNS servers, it might be necessary to disable the sending of SOA records from the cluster to get on-box DNS zones working with multiple subnets. You can disable these records with this advanced privilege command:

```
cluster::> set advanced
cluster::*> network options send-soa modify -enable true
```

Note: If you use multiprotocol NAS (CIFS/SMB and NFS) on the same cluster and choose to disable `send-soa`, be sure that both environments function properly with sending of SOA records disabled.

Disabling the sending of SOA records renders the on-box DNS zone as a nonauthoritative responder to DNS requests.

Using data LIFs as authoritative name servers for clients

Because data LIFs can be configured to listen on port 53 for DNS requests and act as SOA servers, they can also be used as name servers on clients and act as independent DNS servers. This configuration can be useful in environments in which DNS servers might not be able to be modified or when clients do not have access to DNS servers in the domain.

To use data LIFs as name servers, simply configure the client's DNS configuration (`resolv.conf` on Linux clients, DNS property boxes on Windows clients). For details and examples of this use, see the section Configuring Clients to Use ONTAP Data LIFs as DNS Servers.

Third-party load balancers

Hardware or software solutions that perform DNS load balancing (such as F5 networks) are supported by ONTAP for load balancing traffic, such as DNS. Contact the vendor for configuration and support information for third-party load balancers.

On-box DNS impact to applications

Some applications may not interact well with on-box DNS. For example, Oracle dNFS can experience potential corruption during a network issue/interruption (including storage failovers) because the application may not realize it received a new IP for a given host during a reconnect. For more information, see TR-3633: Oracle Databases on ONTAP.

This sort of issue does not impact all applications the same, so be sure to test properly in your environment and check with your application vendor before deploying on-box DNS.

Configuring on-box DNS load balancing

This section covers configuration of on-box DNS load balancing in ONTAP.

Configuring on-box DNS on the SVM

To configure on-box DNS for the cluster, select the appropriate data LIFs to participate in the load balance. Be sure to designate data LIFs to act as DNS servers that listen for DNS queries.

1. Enable DNS zones on the data LIF.

```
::> net int modify -vserver [SVM] -lif [LIF] -dns-zone [cdot.domain.com]
```

2. Set the desired LIF to listen for DNS queries (8.2 and later only).

```
::> net int modify -vserver [SVM] -lif [LIF] -listen-for-dns-query true
```

3. Configure the lb-weight in advanced privilege mode on the data LIF to “load” or the desired lb-weight.

```
::*> net int modify -vserver [SVM] -lif [LIF] -lb-weight load
```

Configuring Windows DNS Server to Work with On-Box DNS

The following configuration steps can be used to configure on-box DNS on Windows DNS servers. The following scenarios are covered in this section:

- Delegations
- Stub zones
- Conditional forwarders

Setting up DNS delegations in Windows DNS

The following steps show how to set up DNS delegations in Windows DNS servers. The server version used in the example is Windows 2008R2, but the same steps apply for other Windows servers. For official steps, refer to the Microsoft TechNet documentation.

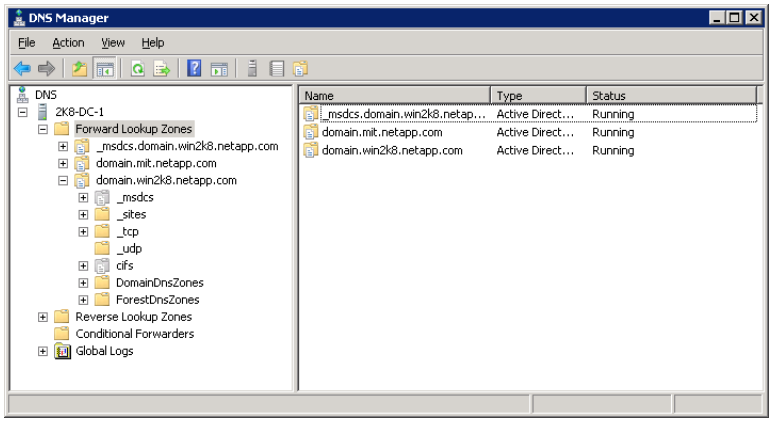
DNS delegations are used for the following purposes:

- Delegating management of a DNS namespace to another location in your organization
- Dividing large zones into smaller zones to distribute load among multiple servers or create better fault tolerance
- Extending the namespace to add additional subdomains

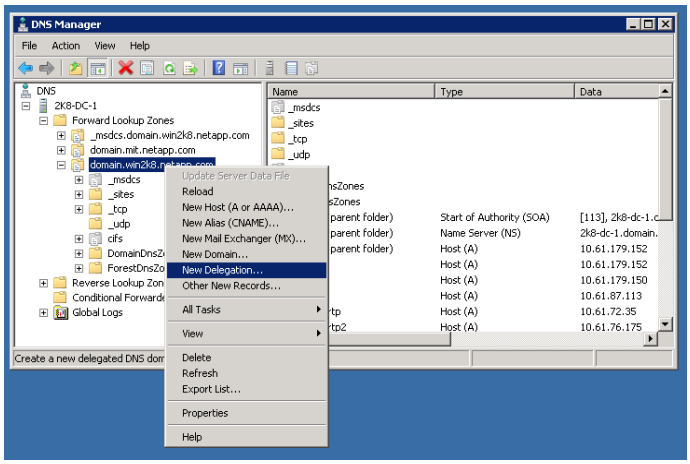
In the case of on-box DNS, delegations can be used to redirect DNS zone traffic to data LIFs on an SVM. Generally speaking, a delegation would be used if the data LIF DNS zones are in the same DNS domain as the DNS servers. One example is if the data LIFs use `cluster.domain.com` and the DNS servers’ domain is `domain.com`.

To set up DNS delegations, complete the following steps:

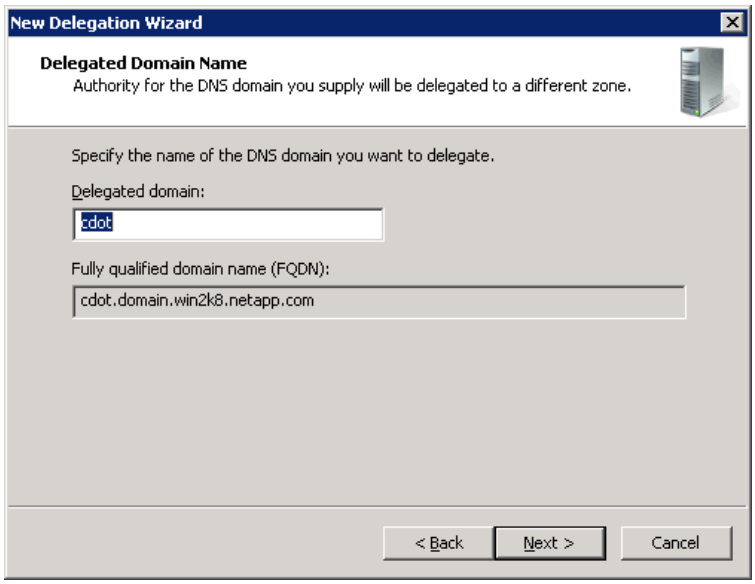
1. Open the DNS Manager console.



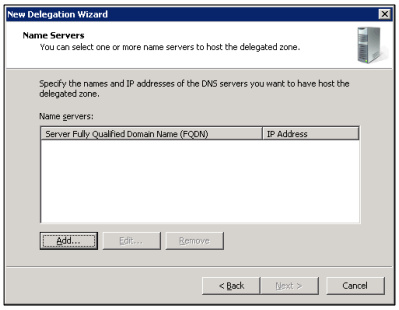
2. Right-click the DNS domain and select New Delegation.

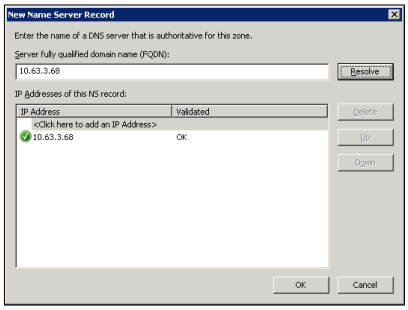


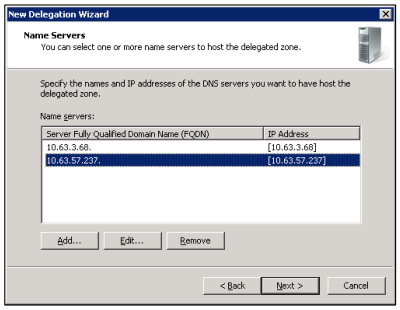
3. Enter the name of the delegated domain.



4. Add the ONTAP SVM data LIFs as name servers (one at a time).



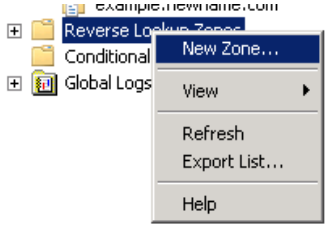




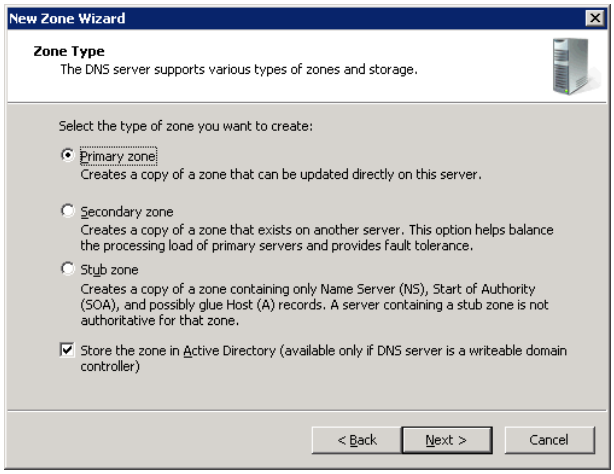
To setup reverse lookup zones and PTR records, complete the following steps:

Note: On-box DNS does not support reverse lookups for IPv4 earlier than ONTAP 8.2. IPv6 support was added in ONTAP 8.3. If you want to force clients to use the host name only for Kerberos, do not create PTR records. Doing so prevents direct IP mounts and makes sure that load balancing is enforced. However, in some cases, PTR records are required for Kerberized NFS to work.

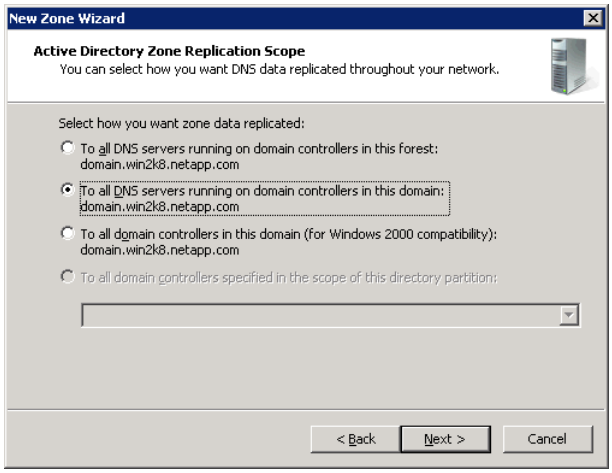
1. Create the reverse lookup zones for the data LIFs.



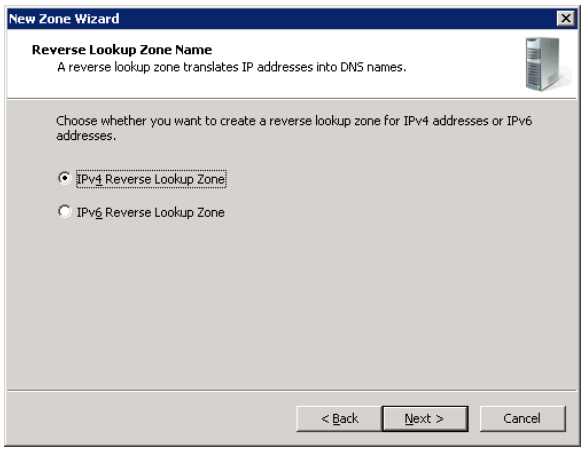
2. Select Primary Zone because DNS in ONTAP cannot service reverse lookups.



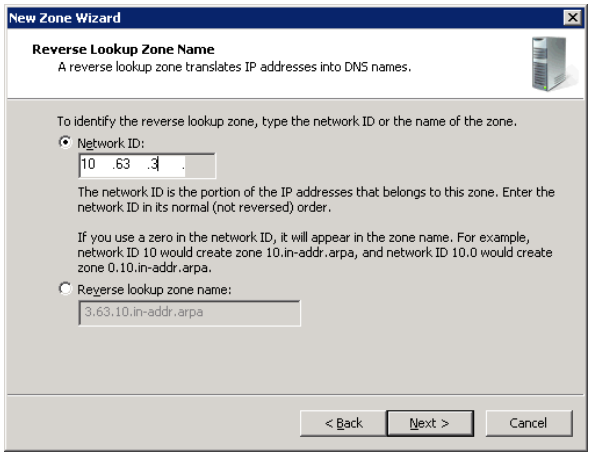
3. Select a zone replication policy to use.



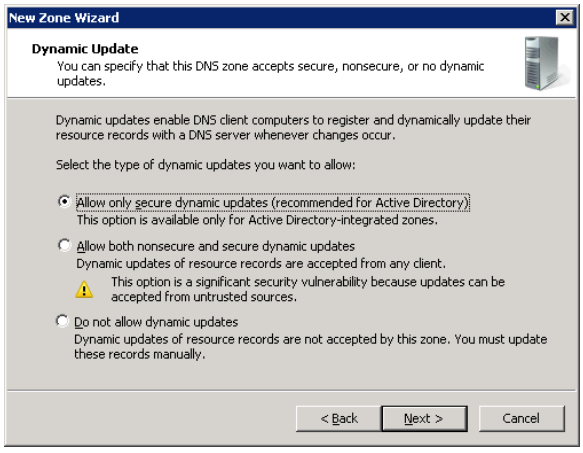
4. Select IPv4 or IPv6 for the lookup zone, depending on what the ONTAP version supports and what the data LIFs use.

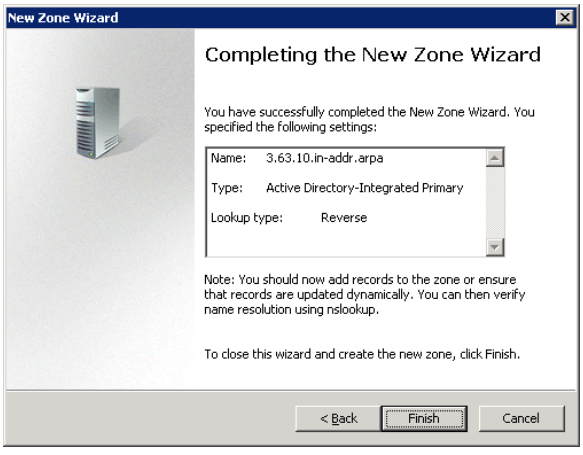


5. Enter the network ID/subnet (the first three octets of the IP address).



6. Select a dynamic update policy.





7. Repeat steps 6 through 11 for other subnets.
8. Test DNS lookups for the new zone by using `nslookup` or `dig`.

```
C:\>nslookup cdot
Server:  UnKnown
Address:  ::1
```

```
Non-authoritative answer:
Name:    cdot.domain.win2k8.netapp.com
Address:  10.63.57.237

C:\>nslookup cdot
Server:  UnKnown
Address:  ::1

Non-authoritative answer:
Name:    cdot.domain.win2k8.netapp.com
Address:  10.63.3.68
```

Setting up DNS stub zones in Windows DNS

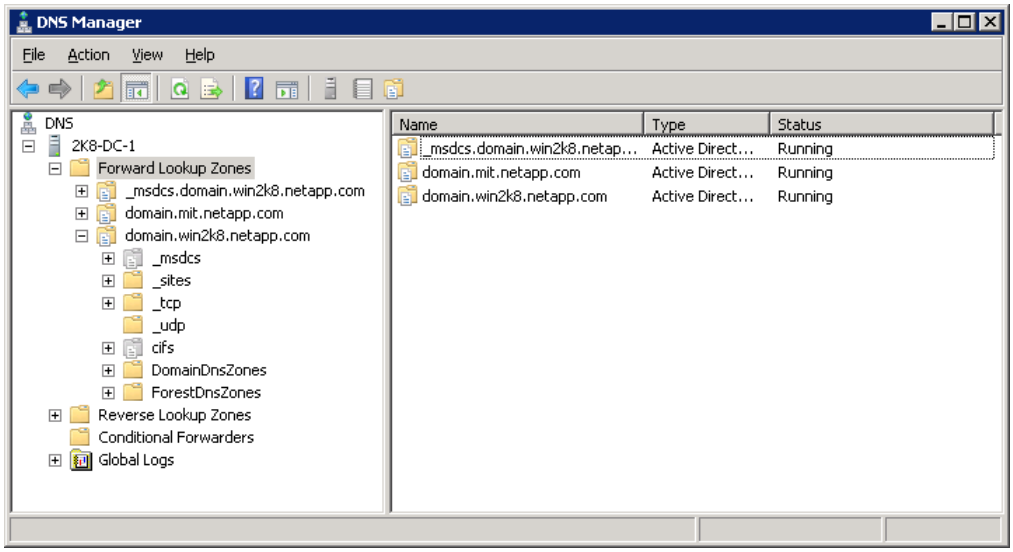
The following steps show how to set up DNS stub zones in Windows DNS servers. The server version used in the example is Windows 2008R2, but the same steps apply for other Windows servers. For official steps, see the Microsoft TechNet documentation.

Stub zones are used when a DNS zone must be integrated with Active Directory and/or when the zone requires SOA records. With on-box DNS, this is an ideal setup because data LIFs that listen as DNS servers can be listed as SOA records in stub zones.

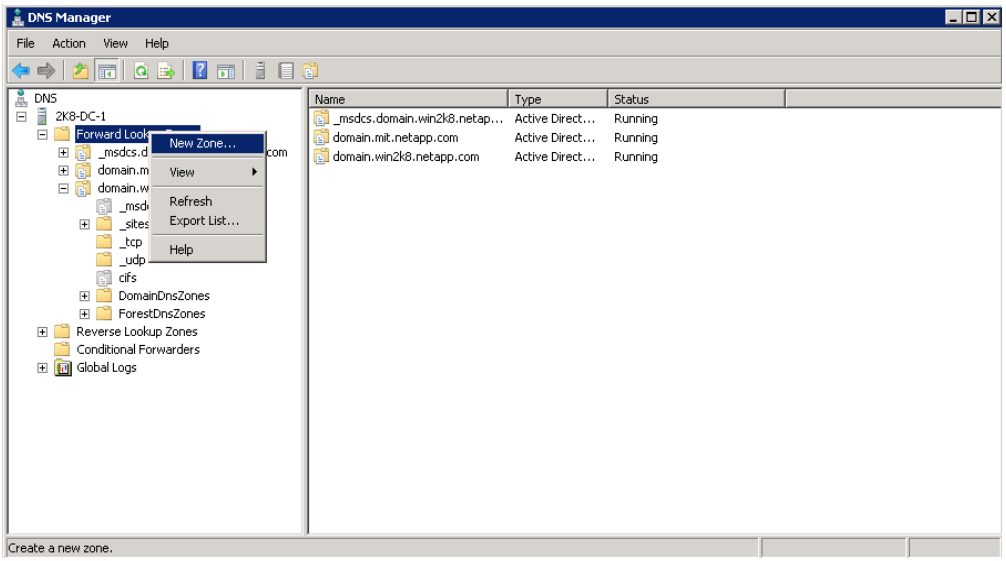
Note: ONTAP on-box DNS only sends SOA records. Glue records are not sent by ONTAP.

To set up stub zones, complete the following steps:

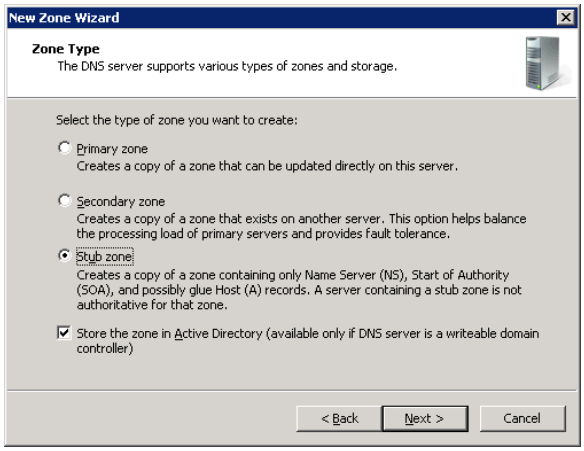
1. Open the DNS Manager console.



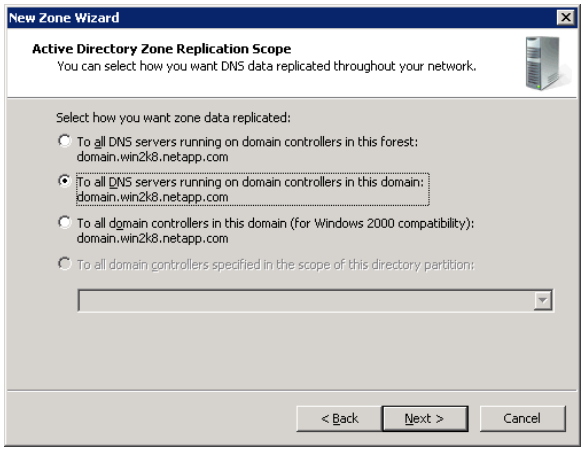
2. Right-click Forward Lookup Zones and select New Zone.



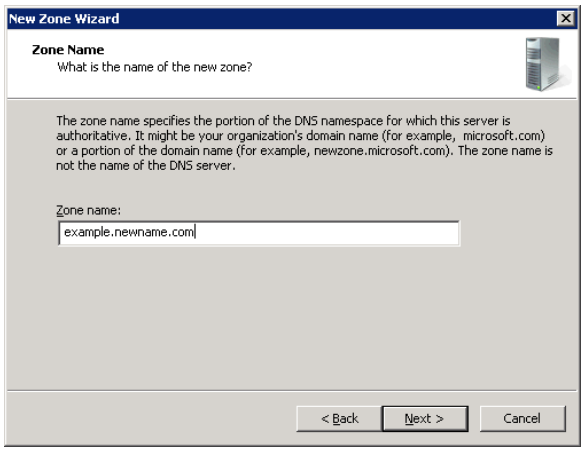
3. Select Stub Zone as the zone.



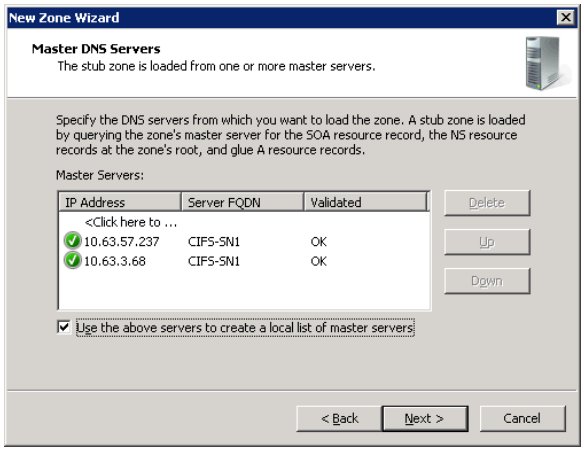
4. Select how zone replication should function.



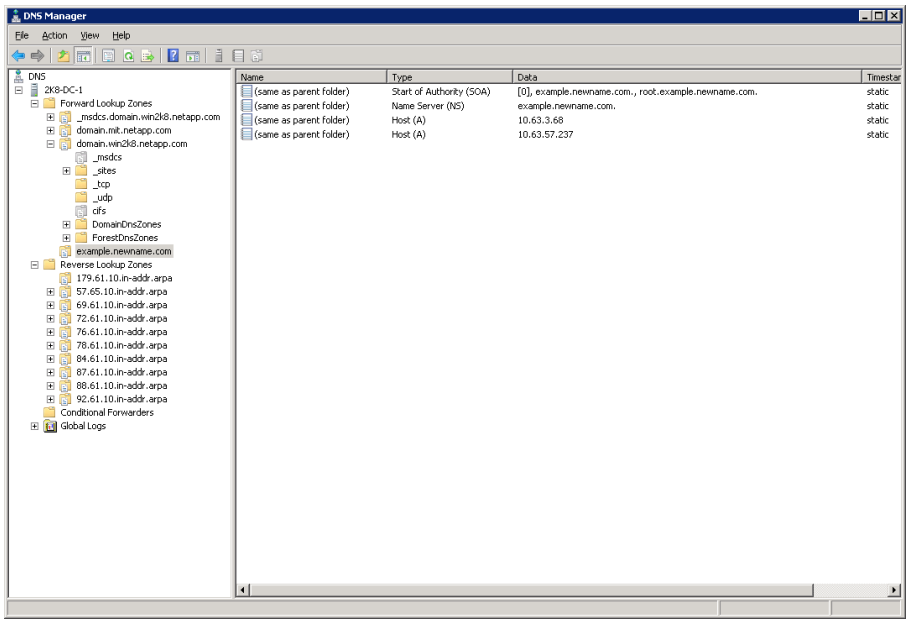
5. Specify the zone name.



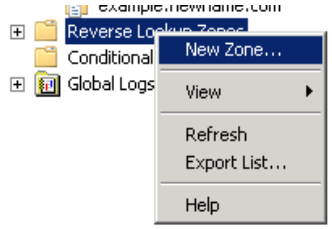
6. Add all data LIFs that are configured for on-box DNS to the master DNS server list. Select the Use the Above Servers to Create a Local List of Master Servers check box.



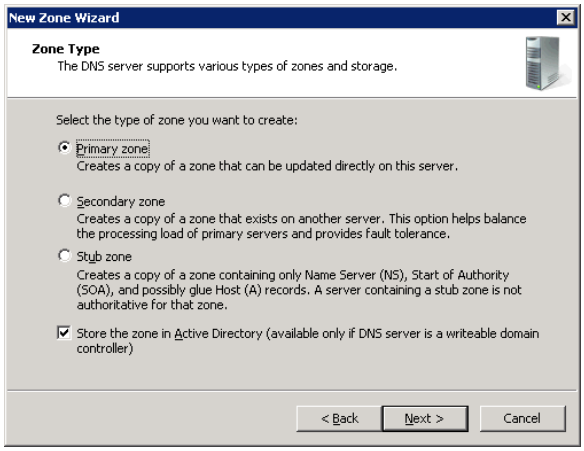
7. Verify that the stub zone has the SOA and NS records.



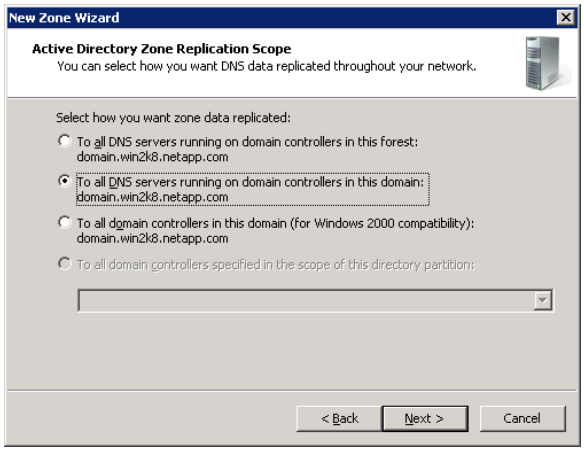
8. Create the reverse lookup zones for the data LIFs.



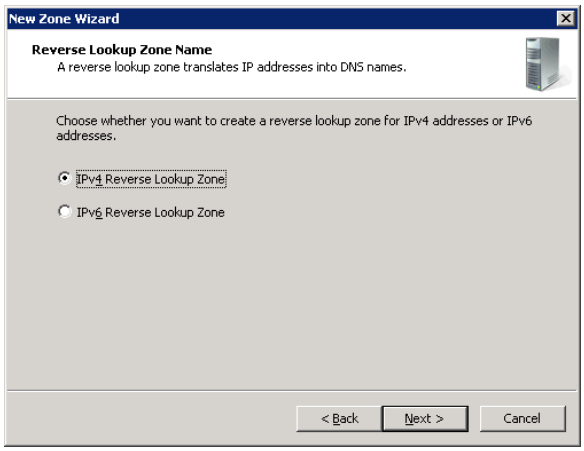
9. Select Primary Zone because DNS in ONTAP cannot service reverse lookups.



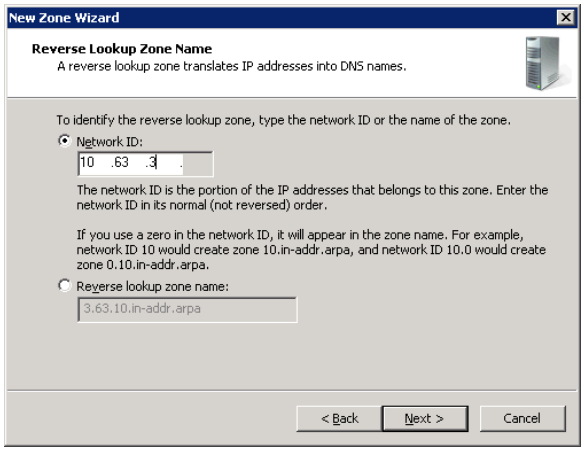
10. Select a zone replication policy to use.



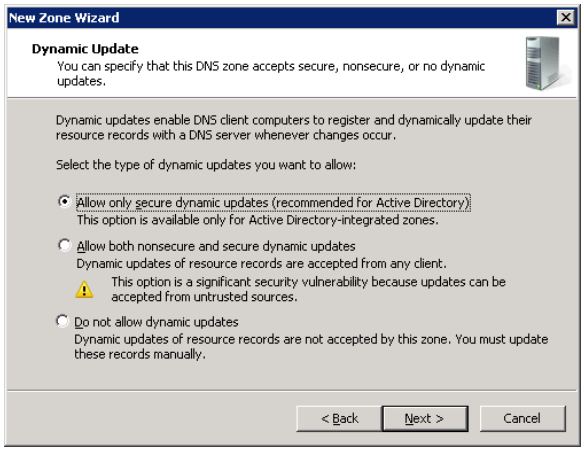
11. Select IPv4 or IPv6 for the lookup zone, depending on what the ONTAP version supports and what the data LIFs use.

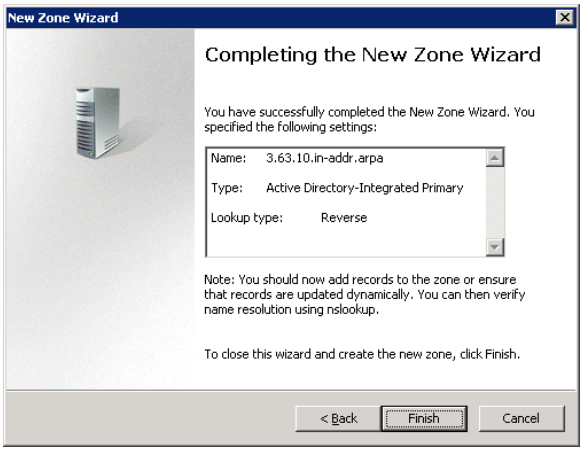


12. Enter the network ID/subnet (the first three octets of the IP address).

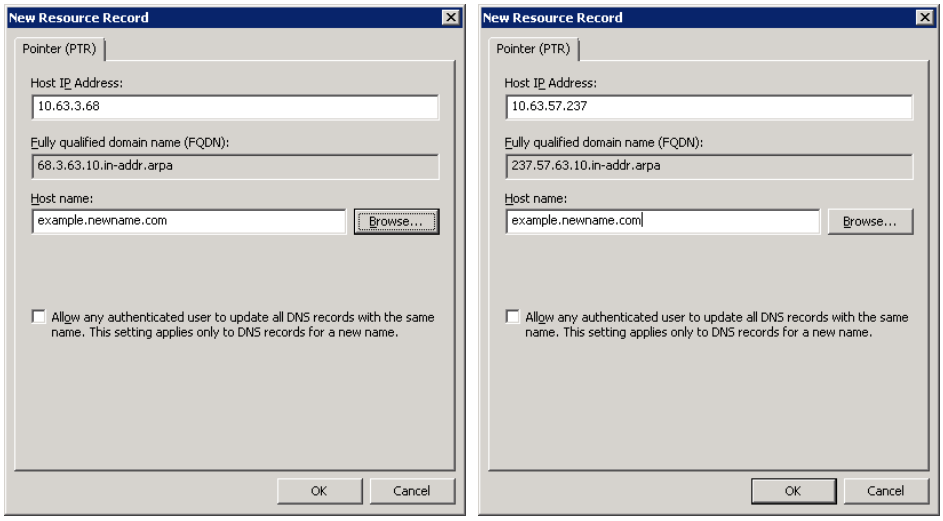


13. Select a dynamic update policy.





14. Repeat steps 8 through 13 for other subnets.
15. Add the PTR records for the data LIFs, because ONTAP does not support reverse name lookups.



16. Use `nslookup` to test the forward and reverse lookups in DNS.

```
C:\>nslookup example.newname.com
Server:  localhost
Address:  ::1

Name:    example.newname.com
Addresses:  10.63.57.237
          10.63.3.68

C:\>nslookup 10.63.57.237
Server:  localhost
Address:  ::1

Name:    example.newname.com
Address:  10.63.57.237


C:\>nslookup 10.63.3.68
Server:  localhost
Address:  ::1
```

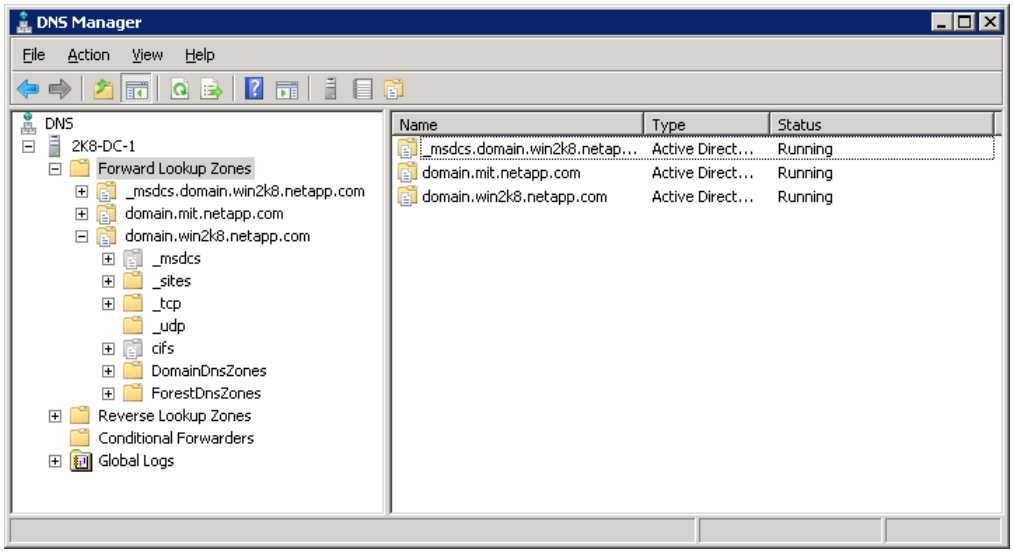
Setting up conditional forwarders in Windows DNS

The following steps show how to set up DNS conditional forwarders in Windows DNS servers. The server version used in the example is Windows 2008R2, but the same steps apply for other Windows servers. For official steps, refer to the Microsoft TechNet documentation.

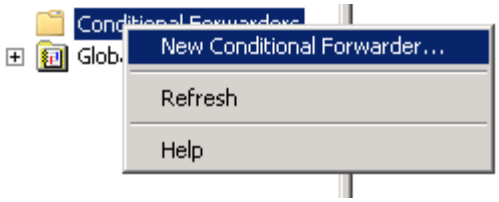
Conditional forwarders are used to forward DNS queries according to the DNS domain name in the query to a DNS server in the DNS domain. In most cases, conditional forwarders are appropriate to use with on-box DNS when the data LIF DNS domain name to be forwarded is in a different domain than the DNS domain of the main DNS servers. An example is when queries to `example.different.com` are forwarded with a conditional forwarder configured in the DNS domain `domain.com`.

To set up conditional forwarders in Windows 2008, complete the following steps:

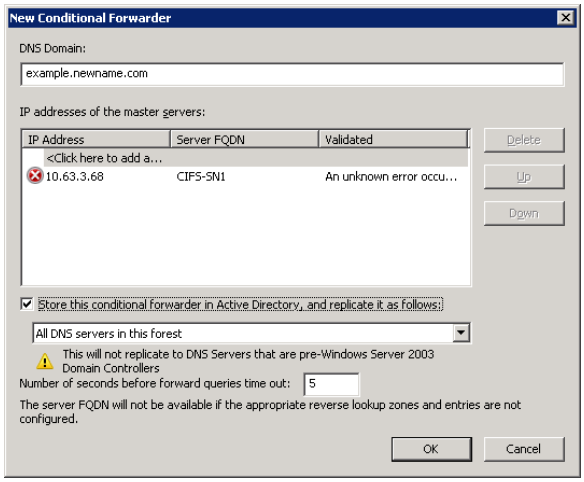
1. Open the DNS Manager console.



2. Right-click Conditional Forwarders and select New Conditional Forwarder.



3. Enter the DNS domain and data LIFs. If an error occurs, the server might not be sending SOA record requests. Either correct that issue or use a stub zone instead.



4. Click OK and use `nslookup` to test the forwarded zone.

```
C:\>nslookup example.newname.com
Server:  localhost
Address:  ::1

Name:    example.newname.com
Addresses:  10.63.57.237
          10.63.3.68
```

```
C:\>nslookup 10.63.57.237
Server:  localhost
Address:  ::1

Name:    example.newname.com
Address:  10.63.57.237


C:\>nslookup 10.63.3.68
Server:  localhost
Address:  ::1
```

Configuring BIND-style DNS servers to work with on-box DNS

In many cases, Windows servers are used for DNS resolution, particularly when Active Directory is present in an environment. This is because Active Directory requires DNS for functionality as well as the simple integration and GUI provided by Windows.

However, some environments leverage Linux-based DNS servers, such as BIND or BIND9. In those configurations, the same concepts apply when considering the design of on-box DNS, as covered in the section “Deciding Which Configuration to Use with BIND DNS.”

In the following example, the DNS server used is a CentOS/RHEL 7 box using BIND as its DNS server.

The following configurations are covered:

- Data LIFs with DNS zones in the same domain as the primary DNS server
- Data LIFs with DNS zones in a different domain than the primary DNS server

On-box DNS configuration: Data LIFs in same domain as BIND server

To use data LIFs in the same domain as the parent domain of the BIND server, use a subdomain entry in the zone file. Subdomains allow the DNS server to pass the requests for a specific zone on to the appropriate servers through zone transfers, providing fault tolerance. If subdomains are not used, the DNS server might think the request is an A/AAAA record request and the lookup fails with NXDOMAIN (domain does not exist).

In BIND servers, adding zones is as simple as modifying configuration files. To add a subdomain, complete the following steps:

1. Add a zone configuration for the on-box DNS subdomain to the master zone file.
2. Add NS and A (glue) records for the data LIFs that are listening for DNS queries.
3. Add an NS record for the parent DNS server.

The following example shows how to set up a subdomain for a zone in the same DNS domain as the parent DNS server. This is the SVM’s on-box DNS configuration:

```
cluster::> net int show-zones -vserver SVM
  (network interface show-zones)
                                                          Listen For
Vserver         Interface Name  DNS Zone                  DNS Query
--------------  --------------  ----------------          ----------
SVM
                data            onbox.bind.SVM.com
                                                          true
                data2           onbox.bind.SVM.com
                                                          false
2 entries were displayed.
```

This is the DNS server’s domain/host name:

```
# hostname
dns.bind.SVM.com
```

The following sample subdomain zone was added to the master zone file:

```
$ORIGIN onbox.bind.SVM.com.
@                       IN      NS      onbox.bind.SVM.com.
                        IN      NS      dns.bind.SVM.com.
onbox.bind.SVM.com.    IN      A       10.193.67.226
```

After these steps are taken, on-box DNS requests are returned for that zone from the cluster:

```
[root@centos7 ~]# nslookup onbox
Server:         10.193.67.227
Address:        10.193.67.227#53

Non-authoritative answer:
Name:   onbox.bind.SVM.com
Address: 10.193.67.226

[root@centos7 ~]# nslookup onbox
Server:         10.193.67.227
Address:        10.193.67.227#53

Non-authoritative answer:
Name:   onbox.bind.SVM.com
Address: 10.193.67.229
```

Adding PTR records to BIND DNS servers

In some cases, it might be necessary to add PTR records to BIND DNS servers so that reverse lookups work for the SVM data LIFs participating in the DNS zone. In particular, adding PTR records comes into play when Kerberos is involved.

Adding PTR records is done the way any other PTR record addition is made. Add the necessary entries to the zone file for the desired reverse lookup zone.

See the following example:

```
[root@dns named]# cat 67.193.10.in-addr.arpa.zone
$TTL 86400
@   IN  SOA     bind.SVM.com.        root.SVM.bind.com. (
        2013042202  ;Serial
        3600        ;Refresh
        1800        ;Retry
        604800      ;Expire
        86400       ;Minimum TTL
)

67.193.10.in-addr.arpa.                IN      NS      dns.bind.SVM.com.

225     IN      PTR     centos7.bind.SVM.com
227     IN      PTR     dns.bind.SVM.com
226     IN      PTR     onbox.cluster.com
229     IN      PTR     onbox.cluster.com
```

See the following example of working reverse lookups:

```
[root@centos7 ~]# dig PTR 10.193.67.226

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <<>> PTR 10.193.67.226
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 44516
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;10.193.67.226.                 IN      PTR
```

```
;; AUTHORITY SECTION:
.                       10793   IN      SOA     a.root-servers.net. nstld.verisign-grs.com.
2016062700 1800 900 604800 86400
```

BIND9 configurations and other third-party DNS servers

BIND9 DNS servers use the same general configuration, but the location of the files is different. For example, `named.conf` for BIND9 is stored in `/etc/bind` rather than `/etc/named`. Be sure to check your DNS application's product documentation and man pages for details.

For other third-party DNS servers, such as those that implement GUIs, the concepts behind the design are the same.

- Use subdomains for on-box DNS configurations in the same DNS domain as the parent.
- Use forwarders for on-box DNS configurations in different DNS domains.

For additional information, contact the provider of the third-party GUI.

On-box DNS configuration: Data LIFs in different domain than BIND server

To use data LIFs in a different domain than the parent domain of the BIND server, add a forwarding zone entry into `named.conf`. Keep in mind that the forwarding zone might not replicate to other DNS servers, so plan accordingly.

The forwarding entry requires the following information:

- Name of the DNS zone
- Type of forward
- Forwarder entries to IP addresses of the data LIFs to be used as DNS servers
- If you are using multiple DNS servers, add the zone to those as well, because `named.conf` might not be configured to replicate to those other servers.

This is the SVM's on-box DNS configuration:

```
cluster::> net int show-zones -vserver SVM
  (network interface show-zones)
                                                Listen For
Vserver         Interface Name  DNS Zone          DNS Query
--------------  --------------  ---------------- ----------
SVM
                data            onbox.cluster.com
                                                  true
                data2           onbox.cluster.com
                                                  false
2 entries were displayed.
```

This is the DNS server's domain/host name:

```
# hostname
dns.bind.SVM.com
```

See the following sample configuration for forwarding zone in named.conf for BIND.

```
zone "onbox.cluster.com" IN {
        type forward;
        forwarders {10.193.67.226;};
};
```

After these steps are taken, the following are the results of nslookup for that zone:

```
[root@centos7 ~]# nslookup onbox.cluster.com
Server:         10.193.67.227
```

```
Address:        10.193.67.227#53

Non-authoritative answer:
Name:   onbox.cluster.com
Address: 10.193.67.229

[root@centos7 ~]# nslookup onbox.cluster.com
Server:         10.193.67.227
Address:        10.193.67.227#53

Non-authoritative answer:
Name:   onbox.cluster.com
Address: 10.193.67.226
```

Configuring clients to use ONTAP data LIFs as DNS servers

In some cases, clients might need to be configured to use the cluster data LIFs as DNS servers. Instances when this might be necessary include the following:

- Clients do not have network access to primary DNS servers.
- Primary DNS servers cannot be modified to use zones, delegations, or forwarders.
- General preference.

Clients can use multiple name servers and zones when resolving host names, so clients can use both primary DNS domains as well as the data LIF domains configured on the cluster. It is also possible to use data LIFs as local DNS servers as well as use on-box DNS with general DNS zone configuration on the same SVM.

The first step to configure on-box DNS as the client's DNS name server is to configure on-box DNS on the SVM, enabling at least one data LIF to listen for DNS queries. It is also necessary to confirm that the cluster is sending SOA records.

```
cluster::> net int modify -vserver SVM -lif data -dns-zone cluster.local -listen-for-dns-query
true

cluster::> net int show -vserver SVM1 -lif data -fields dns-zone,listen-for-dns-query,address
  (network interface show)
vserver lif  address       dns-zone      listen-for-dns-query
------- ---- ------------- ------------- --------------------
SVM1    data 10.193.67.220 cluster.local true

cluster::> set advanced
cluster::*> network options send-soa show
Enable sending SOA: true
```

Next, configure the client to use the data LIF as a DNS name server and add the search domain configured on the data LIF.

Configuring Linux clients with resolv.conf

In Linux clients, such configuring would be done with `resolv.conf` files. The following client shows that, before configuring `resolv.conf`, the DNS domain `cluster.local` could not be resolved.

```
# dig cluster.local

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <<>> cluster.local
;; global options: +cmd
;; connection timed out; no servers could be reached
```

When the client is configured to use the data LIF with that DNS zone, it can resolve properly.

See the following example of configuring on-box data LIFs as DNS servers for Linux clients:

```
# cat /etc/resolv.conf
```

```
# Generated by NetworkManager
search cluster.local
nameserver 10.193.67.220

# dig cluster.local

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <<>> cluster.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15220
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;cluster.local.                 IN      A

;; ANSWER SECTION:
cluster.local.          0       IN      A       10.193.67.220

;; AUTHORITY SECTION:
cluster.local.          86400   IN      NS      cluster.local.

;; Query time: 24 msec
;; SERVER: 10.193.67.220#53(10.193.67.220)
;; WHEN: Tue Jun 21 13:02:44 EDT 2016
;; MSG SIZE  rcvd: 72
```

Reverse lookup works as well:

```
# dig 10.193.67.220

; <<>> DiG 9.9.4-RedHat-9.9.4-29.el7_2.3 <<>> 10.193.67.220
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 60475
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;10.193.67.220.                 IN      A

;; Query time: 12 msec
;; SERVER: 10.193.67.220#53(10.193.67.220)
;; WHEN: Tue Jun 21 13:03:18 EDT 2016
;; MSG SIZE  rcvd: 42
```

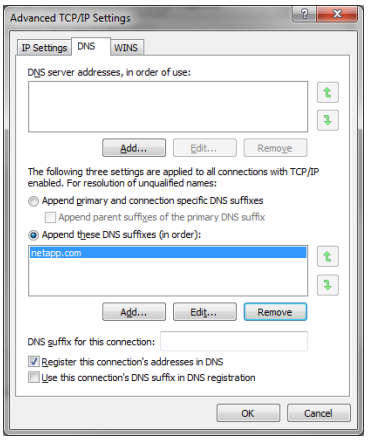
Other DNS servers can be added to the configuration and resolve names properly. For example, if we add a Google DNS server, we can resolve google.com:

```
# cat /etc/resolv.conf
# Generated by NetworkManager
search cluster.local
nameserver 10.193.67.220
nameserver 8.8.8.8

# nslookup google.com
;; Got recursion not available from 10.193.67.220, trying next server
Server:         8.8.8.8
Address:        8.8.8.8#53

Non-authoritative answer:
Name:   google.com
Address: 216.58.219.206
```

Configuring Windows clients to use on-cox DNS as independent DNS servers

Windows clients can also use ONTAP data LIFs as DNS servers for data access on an SVM. Windows configurations generally use a GUI, but CLI utilities such as PowerShell can also be used. This example covers the GUI configuration and leverages the use of the data LIFs as DNS servers in addition to an existing DNS configuration.

This is the existing DNS configuration for the Windows client:

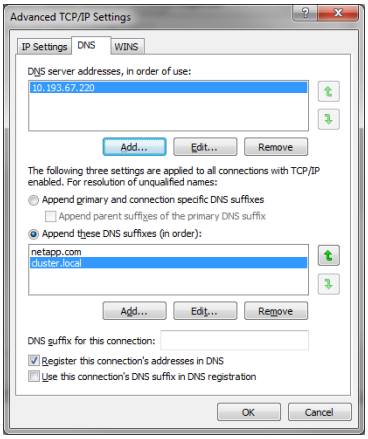


The DNS servers are being pulled through DHCP. The DNS suffixes have been manually configured.

As it currently stands, nslookup requests for the data LIF's zone (`cluster.local`) fail:

```
C:\>nslookup cluster.local
Server:  dns.netapp.com
Address:  10.193.67.200

*** dns.netapp.com can't find cluster.local: Non-existent domain
```

To use the cluster's data LIFs as DNS servers to return cluster data LIFs when `cluster.local` is queried, the configuration should look like this:



Here, we added only the data LIF participating in on-box load balancing as a DNS server. Other DNS servers would also need to be added.

After the new server is added, flush the DNS cache (Windows caches DNS for 24 hours by default) and try the nslookup for the cluster zone:

```
C:\> nslookup cluster.local
Server:  cluster.local
Address:  10.193.67.220

Name:    cluster.local
Address:  10.193.67.220
```

Conclusion

On-box DNS load balancing is a viable alternative to using external solutions, such as round-robin DNS load balancing. Being able to load balance DNS requests based on load helps alleviate the overall effect to a scale-out cluster and provides an intelligent method to serve NAS connectivity in enterprise environments.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

- ONTAP Documentation Center
  https://docs.netapp.com/ontap-9/index.jsp
- ONTAP & ONTAP System Manager Documentation Resources
  https://www.netapp.com/data-management/oncommand-system-documentation/
- NetApp Product Documentation
  https://www.netapp.com/support-and-training/documentation/

Version history

Version	Date	Document Version History
Version 1.0	July 2016	Initial release
Version 2.0	October 2016	Updated for ONTAP 9.1
Version 2.1	May 2020	Minor revision
Version 2.2	February 2021	Minor revision

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

TR-4523-0221

NetApp