Contents

Pricing calculator

Service updates

Private DNS zone migration guide

What is Azure DNS?

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

Azure DNS is a hosting service for DNS domains that provides name resolution by using Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records by using the same credentials, APIs, tools, and billing as your other Azure services.

You can't use Azure DNS to buy a domain name. For an annual fee, you can buy a domain name by using App Service domains or a third-party domain name registrar. Your domains then can be hosted in Azure DNS for record management. For more information, see Delegate a domain to Azure DNS.

The following features are included with Azure DNS.

Reliability and performance

DNS domains in Azure DNS are hosted on Azure's global network of DNS name servers. Azure DNS uses anycast networking. Each DNS query is answered by the closest available DNS server to provide fast performance and high availability for your domain.

Security

Azure DNS is based on Azure Resource Manager, which provides features such as:

- Role-based access control to control who has access to specific actions for your organization.
- Activity logs to monitor how a user in your organization modified a resource or to find an error when troubleshooting.
- Resource locking to lock a subscription, resource group, or resource. Locking prevents other users in your organization from accidentally deleting or modifying critical resources.

For more information, see How to protect DNS zones and records.

DNSSEC

Azure DNS does not currently support DNSSEC. In most cases, you can reduce the need for DNSSEC by consistently using HTTPS/TLS in your applications. If DNSSEC is a critical requirement for your DNS zones, you can host these zones with third party DNS hosting providers.

Ease of use

Azure DNS can manage DNS records for your Azure services and provide DNS for your external resources as well. Azure DNS is integrated in the Azure portal and uses the same credentials, support contract, and billing as your other Azure services.

DNS billing is based on the number of DNS zones hosted in Azure and on the number of DNS queries received. To learn more about pricing, see Azure DNS pricing.

Your domains and records can be managed by using the Azure portal, Azure PowerShell cmdlets, and the cross-platform Azure CLI. Applications that require automated DNS management can integrate with the service by using the REST API and SDKs.

Customizable virtual networks with private domains

Azure DNS also supports private DNS domains. This feature allows you to use your own custom domain names in your private virtual networks rather than the Azure-provided names available today.

For more information, see Use Azure DNS for private domains.

Alias records

Azure DNS supports alias record sets. You can use an alias record set to refer to an Azure resource, such as an Azure public IP address, an Azure Traffic Manager profile, or an Azure Content Delivery Network (CDN) endpoint. If the IP address of the underlying resource changes, the alias record set seamlessly updates itself during DNS resolution. The alias record set points to the service instance, and the service instance is associated with an IP address.

Also, you can now point your apex or naked domain to a Traffic Manager profile or CDN endpoint using an alias record. An example is contoso.com.

For more information, see Overview of Azure DNS alias records.

Next steps

- To learn about DNS zones and records, see DNS zones and records overview.
- To learn how to create a zone in Azure DNS, see Create a DNS zone.
- For frequently asked questions about Azure DNS, see the Azure DNS FAQ.

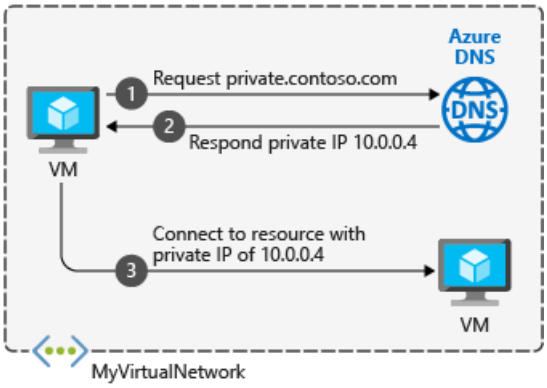
What is Azure Private DNS?

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

The Domain Name System, or DNS, is responsible for translating (or resolving) a service name to its IP address. Azure DNS is a hosting service for DNS domains, providing name resolution using the Microsoft Azure infrastructure. In addition to supporting internet-facing DNS domains, Azure DNS also supports private DNS zones.

Azure Private DNS provides a reliable, secure DNS service to manage and resolve domain names in a virtual network without the need to add a custom DNS solution. By using private DNS zones, you can use your own custom domain names rather than the Azure-provided names available today. Using custom domain names helps you to tailor your virtual network architecture to best suit your organization's needs. It provides name resolution for virtual machines (VMs) within a virtual network and between virtual networks. Additionally, you can configure zones names with a split-horizon view, which allows a private and a public DNS zone to share the name.

To resolve the records of a private DNS zone from your virtual network, you must link the virtual network with the zone. Linked virtual networks have full access and can resolve all DNS records published in the private zone. Additionally, you can also enable autoregistration on a virtual network link. If you enable autoregistration on a virtual network link, the DNS records for the virtual machines on that virtual network are registered in the private zone. When autoregistration is enabled, Azure DNS also updates the zone records whenever a virtual machine is created, changes its' IP address, or is deleted.



NOTE

As a best practice, do not use a *.local* domain for your private DNS zone. Not all operating systems support this.

Benefits

Azure Private DNS provides the following benefits:

- **Removes the need for custom DNS solutions**. Previously, many customers created custom DNS solutions to manage DNS zones in their virtual network. You can now manage DNS zones using the native Azure infrastructure, which removes the burden of creating and managing custom DNS solutions.
- **Use all common DNS records types**. Azure DNS supports A, AAAA, CNAME, MX, PTR, SOA, SRV, and TXT records.
- **Automatic hostname record management**. Along with hosting your custom DNS records, Azure automatically maintains hostname records for the VMs in the specified virtual networks. In this scenario, you can optimize the domain names you use without needing to create custom DNS solutions or modify

applications.

- **Hostname resolution between virtual networks**. Unlike Azure-provided host names, private DNS zones can be shared between virtual networks. This capability simplifies cross-network and service-discovery scenarios, such as virtual network peering.
- **Familiar tools and user experience**. To reduce the learning curve, this service uses well-established Azure DNS tools (Azure portal, Azure PowerShell, Azure CLI, Azure Resource Manager templates, and the REST API).
- **Split-horizon DNS support**. With Azure DNS, you can create zones with the same name that resolve to different answers from within a virtual network and from the public internet. A typical scenario for split-horizon DNS is to provide a dedicated version of a service for use inside your virtual network.
- **Available in all Azure regions**. The Azure DNS private zones feature is available in all Azure regions in the Azure public cloud.

Capabilities

Azure DNS provides the following capabilities:

- **Automatic registration of virtual machines from a virtual network that's linked to a private zone with autoregistration enabled**. The virtual machines are registered (added) to the private zone as A records pointing to their private IP addresses. When a virtual machine in a virtual network link with autoregistration enabled is deleted, Azure DNS also automatically removes the corresponding DNS record from the linked private zone.
- **Forward DNS resolution is supported across virtual networks that are linked to the private zone**. For cross-virtual network DNS resolution, there's no explicit dependency such that the virtual networks are peered with each other. However, you might want to peer virtual networks for other scenarios (for example, HTTP traffic).
- **Reverse DNS lookup is supported within the virtual-network scope**. Reverse DNS lookup for a private IP within the virtual network assigned to a private zone returns the FQDN that includes the host/record name and the zone name as the suffix.

Other considerations

Azure DNS has the following limitations:

- A specific virtual network can be linked to only one private zone if automatic registration of VM DNS records is enabled. You can however link multiple virtual networks to a single DNS zone.
- Reverse DNS works only for private IP space in the linked virtual network
- Reverse DNS for a private IP address for a linked virtual network returns *internal.cloudapp.net* as the default suffix for the virtual machine. For virtual networks that are linked to a private zone with autoregistration enabled, reverse DNS for a private IP address returns two FQDNs: one with default the suffix *internal.cloudapp.net* and another with the private zone suffix.
- Conditional forwarding is not currently natively supported. To enable resolution between Azure and on-premises networks. See Name resolution for VMs and role instances

Pricing

For pricing information, see Azure DNS Pricing.

Next steps

- Learn how to create a private zone in Azure DNS by using Azure PowerShell or Azure CLI.
- Read about some common private zone scenarios that can be realized with private zones in Azure DNS.
- For common questions and answers about private zones in Azure DNS, including specific behavior you can expect for certain kinds of operations, see Private DNS FAQ.
- Learn about DNS zones and records by visiting DNS zones and records overview.
- Learn about some of the other key networking capabilities of Azure.

Quickstart: Create an Azure DNS zone and record using the Azure portal

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

You can configure Azure DNS to resolve host names in your public domain. For example, if you purchased the *contoso.xyz* domain name from a domain name registrar, you can configure Azure DNS to host the *contoso.xyz* domain and resolve *www.contoso.xyz* to the IP address of your web server or web app.

In this quickstart, you will create a test domain, and then create an address record to resolve *www* to the IP address *10.10.10.10*.

> **IMPORTANT**
> All the names and IP addresses in this quickstart are examples that do not represent real-world scenarios.

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

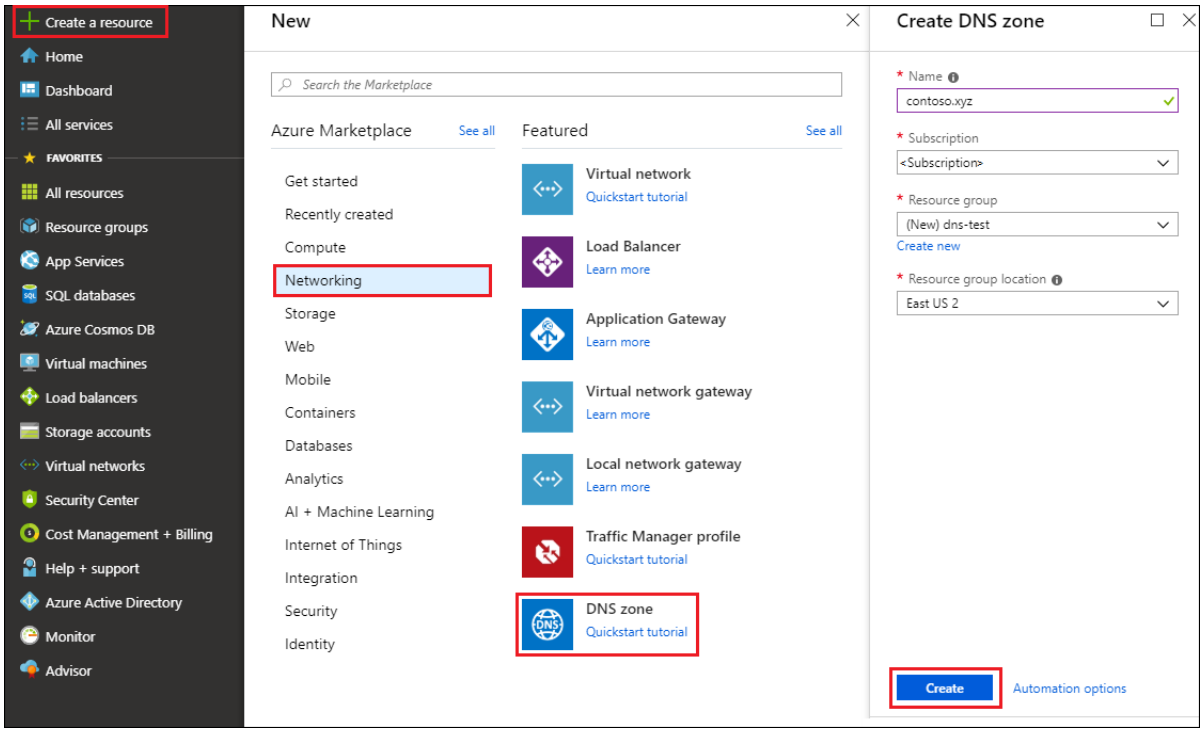
For all portal steps, sign in to the Azure portal.

Create a DNS zone

A DNS zone contains the DNS entries for a domain. To start hosting your domain in Azure DNS, you create a DNS zone for that domain name.

To create the DNS zone:

1. At upper left, select **Create a resource**, then **Networking**, and then **DNS zone**.
2. On the **Create DNS zone** page, type or select the following values:
   - **Name**: Type *contoso.xyz* for this quickstart example. The DNS zone name can be any value that is not already configured on the Azure DNS servers. A real-world value would be a domain that you bought from a domain name registrar.
   - **Resource group**: Select **Create new**, enter *MyResourceGroup*, and select **OK**. The resource group name must be unique within the Azure subscription.
3. Select **Create**.



It may take a few minutes to create the zone.

Create a DNS record

You create DNS entries or records for your domain inside the DNS zone. Create a new address record or 'A' record to resolve a host name to an IPv4 address.

To create an 'A' record:

1. In the Azure portal, under **All resources**, open the **contoso.xyz** DNS zone in the **MyResourceGroup** resource group. You can enter *contoso.xyz* in the **Filter by name** box to find it more easily.
2. At the top of the **DNS zone** page, select **+ Record set**.
3. On the **Add record set** page, type or select the following values:
   - **Name**: Type *www*. The record name is the host name that you want to resolve to the specified IP address.
   - **Type**: Select **A**. 'A' records are the most common, but there are other record types for mail servers ('MX'), IP v6 addresses ('AAAA'), and so on.
   - **TTL**: Type *1*. *Time-to-live* of the DNS request specifies how long DNS servers and clients can cache a response.
   - **TTL unit**: Select **Hours**. This is the time unit for the **TTL** value.
   - **IP address**: For this quickstart example, type *10.10.10.10*. This value is the IP address the record name resolves to. In a real-world scenario, you would enter the public IP address for your web server.

Since this quickstart is just for quick testing purposes, there's no need to configure the Azure DNS name servers at a domain name registrar. With a real production domain, you'll want anyone on the Internet to resolve the host name to connect to your web server or app. You'll visit your domain name registrar to replace the name server records with the Azure DNS name servers. For more information, see Tutorial: Host your domain in Azure DNS.

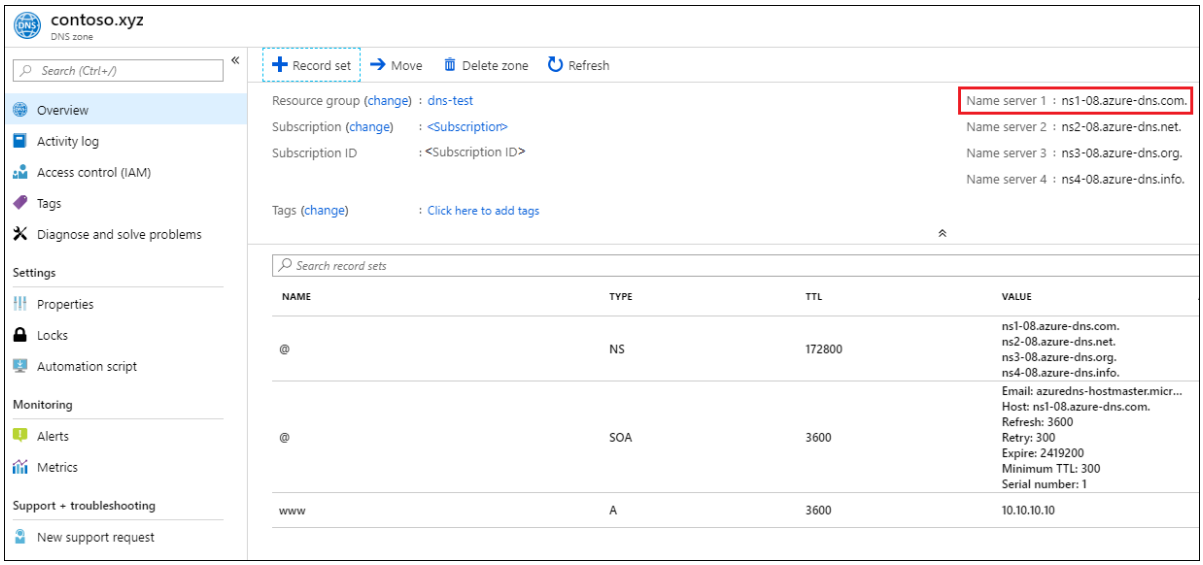
Test the name resolution

Now that you have a test DNS zone with a test 'A' record, you can test the name resolution with a tool called *nslookup*.

To test DNS name resolution:

1. In the Azure portal, under **All resources**, open the **contoso.xyz** DNS zone in the **MyResourceGroup** resource group. You can enter *contoso.xyz* in the **Filter by name** box to find it more easily.

2. Copy one of the name server names from the name server list on the **Overview** page.



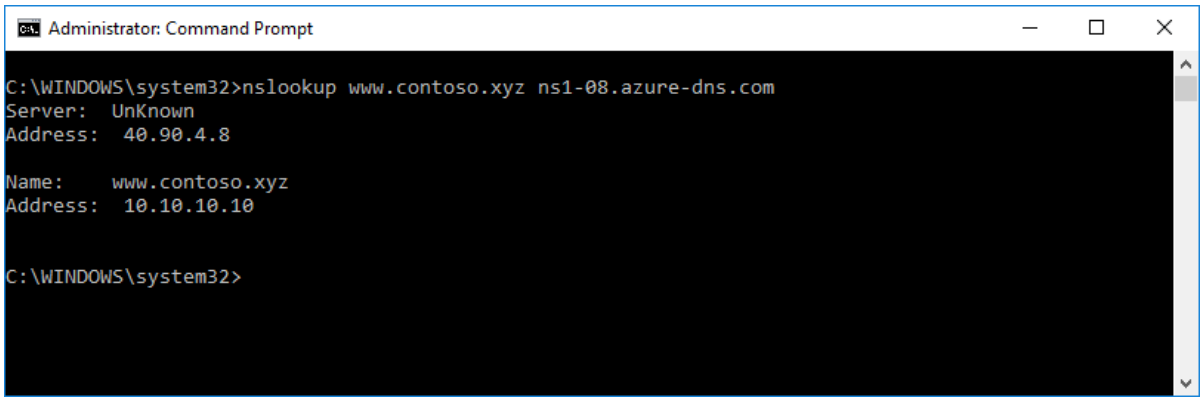
3. Open a command prompt, and run the following command:

```
nslookup www.contoso.xyz <name server name>
```

For example:

```
nslookup www.contoso.xyz ns1-08.azure-dns.com.
```

You should see something like the following screen:



The host name **www.contoso.xyz** resolves to **10.10.10.10**, just as you configured it. This result verifies that name resolution is working correctly.

Clean up resources

When you no longer need the resources you created in this quickstart, remove them by deleting the **MyResourceGroup** resource group. Open the **MyResourceGroup** resource group, and select **Delete resource group**.

Next steps

Create DNS records for a web app in a custom domain

Quickstart: Create an Azure DNS zone and record using Azure PowerShell

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

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.

In this quickstart, you create your first DNS zone and record using Azure PowerShell. You can also perform these steps using the Azure portal or the Azure CLI.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. Finally, to publish your DNS zone to the Internet, you need to configure the name servers for the domain. Each of these steps is described below.

Azure DNS also supports creating private domains. For step-by-step instructions about how create your first private DNS zone and record, see Get started with Azure DNS private zones using PowerShell.

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

Create the resource group

Before you create the DNS zone, create a resource group to contain the DNS zone:

```
New-AzResourceGroup -name MyResourceGroup -location "eastus"
```

Create a DNS zone

A DNS zone is created by using the `New-AzDnsZone` cmdlet. The following example creates a DNS zone called *contoso.xyz* in the resource group called *MyResourceGroup*. Use the example to create a DNS zone, substituting the values for your own.

```
New-AzDnsZone -Name contoso.xyz -ResourceGroupName MyResourceGroup
```

Create a DNS record

You create record sets by using the `New-AzDnsRecordSet` cmdlet. The following example creates a record with the relative name "www" in the DNS Zone "contoso.xyz", in resource group "MyResourceGroup". The fully qualified name of the record set is "www.contoso.xyz". The record type is "A", with IP address "10.10.10.10", and the TTL is 3600 seconds.

```
New-AzDnsRecordSet -Name www -RecordType A -ZoneName contoso.xyz -ResourceGroupName MyResourceGroup -Ttl 3600 -
DnsRecords (New-AzDnsRecordConfig -IPv4Address "10.10.10.10")
```

View records

To list the DNS records in your zone, use:

```
Get-AzDnsRecordSet -ZoneName contoso.xyz -ResourceGroupName MyResourceGroup
```

Test the name resolution

Now that you have a test DNS zone with a test 'A' record, you can test the name resolution with a tool called *nslookup*.

To test DNS name resolution:

1. Run the following cmdlet to get the list of name servers for your zone:

   ```
   Get-AzDnsRecordSet -ZoneName contoso.xyz -ResourceGroupName MyResourceGroup -RecordType ns
   ```

2. Copy one of the name server names from the output of the previous step.

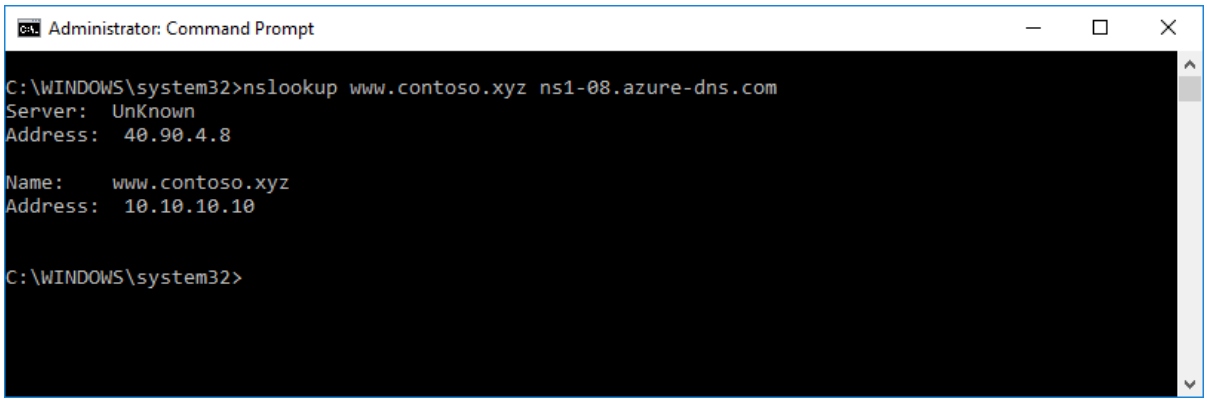
3. Open a command prompt, and run the following command:

   ```
   nslookup www.contoso.xyz <name server name>
   ```

For example:

```
nslookup www.contoso.xyz ns1-08.azure-dns.com.
```

You should see something like the following screen:



The host name **www.contoso.xyz** resolves to **10.10.10.10**, just as you configured it. This result verifies that name resolution is working correctly.

Delete all resources

When no longer needed, you can delete all resources created in this quickstart by deleting the resource group:

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

Next steps

Now that you've created your first DNS zone and record using Azure PowerShell, you can create records for a web app in a custom domain.

Create DNS records for a web app in a custom domain

Quickstart: Create an Azure DNS zone and record using Azure CLI

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

This article walks you through the steps to create your first DNS zone and record using Azure CLI, which is available for Windows, Mac and Linux. You can also perform these steps using the Azure portal or Azure PowerShell.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. Finally, to publish your DNS zone to the Internet, you need to configure the name servers for the domain. Each of these steps is described below.

Azure DNS also supports private DNS zones. To learn more about private DNS zones, see Using Azure DNS for private domains. For an example on how to create a private DNS zone, see Get started with Azure DNS private zones using 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 don't have an Azure subscription, create a free account before you begin.

Create the resource group

Before you create the DNS zone, create a resource group to contain the DNS zone:

```
az group create --name MyResourceGroup --location "East US"
```

Create a DNS zone

A DNS zone is created using the `az network dns zone create` command. To see help for this command, type `az network dns zone create -h`.

The following example creates a DNS zone called *contoso.xyz* in the resource group *MyResourceGroup*. Use the example to create a DNS zone, substituting the values for your own.

```
az network dns zone create -g MyResourceGroup -n contoso.xyz
```

Create a DNS record

To create a DNS record, use the `az network dns record-set [record type] add-record` command. For help on A records, see `azure network dns record-set A add-record -h`.

The following example creates a record with the relative name "www" in the DNS Zone "contoso.xyz" in the resource group "MyResourceGroup". The fully-qualified name of the record set is "www.contoso.xyz". The record type is "A", with IP address "10.10.10.10", and a default TTL of 3600 seconds (1 hour).

```
az network dns record-set a add-record -g MyResourceGroup -z contoso.xyz -n www -a 10.10.10.10
```

View records

To list the DNS records in your zone, run:

```
az network dns record-set list -g MyResourceGroup -z contoso.xyz
```

Test the name resolution

Now that you have a test DNS zone with a test 'A' record, you can test the name resolution with a tool called *nslookup*.

To test DNS name resolution:

1. Run the following cmdlet to get the list of name servers for your zone:

   ```
   az network dns record-set ns show --resource-group MyResourceGroup --zone-name contoso.xyz --name @
   ```

2. Copy one of the name server names from the output of the previous step.

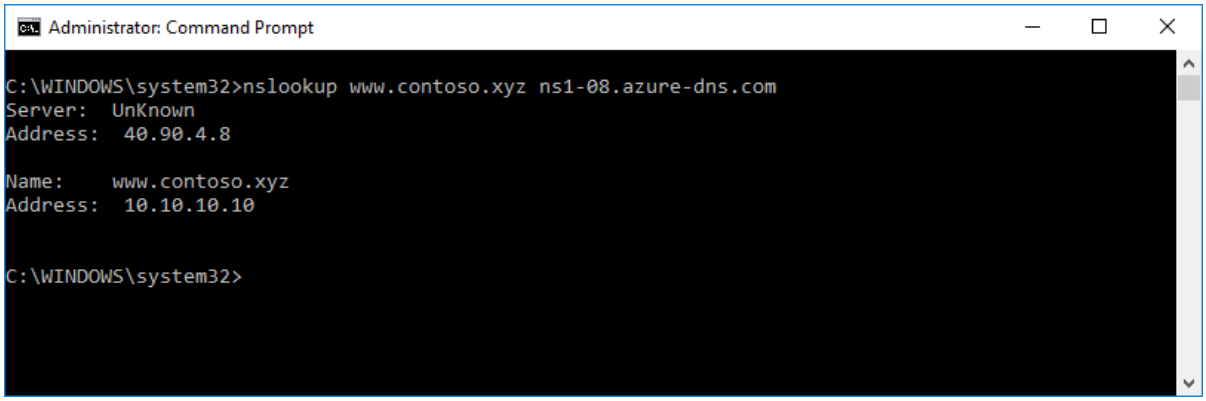
3. Open a command prompt, and run the following command:

   ```
   nslookup www.contoso.xyz <name server name>
   ```

   For example:

```
nslookup www.contoso.xyz ns1-08.azure-dns.com.
```

You should see something like the following screen:



The host name **www.contoso.xyz** resolves to **10.10.10.10**, just as you configured it. This result verifies that name resolution is working correctly.

Delete all resources

When no longer needed, you can delete all resources created in this quickstart by deleting the resource group:

```
az group delete --name MyResourceGroup
```

Next steps

Now that you've created your first DNS zone and record using Azure CLI, you can create records for a web app in a custom domain.

Create DNS records for a web app in a custom domain

Quickstart: Create an Azure private DNS zone using the Azure portal

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

This quickstart walks you through the steps to create your first private DNS zone and record using the Azure portal.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. To publish a private DNS zone to your virtual network, you specify the list of virtual networks that are allowed to resolve records within the zone. These are called *linked* virtual networks. When autoregistration is enabled, Azure DNS also updates the zone records whenever a virtual machine is created, changes its' IP address, or is deleted.

In this quickstart, you learn how to:

- Create a private DNS zone
- Create a virtual network
- Link the virtual network
- Create test virtual machines
- Create an additional DNS record
- Test the private zone

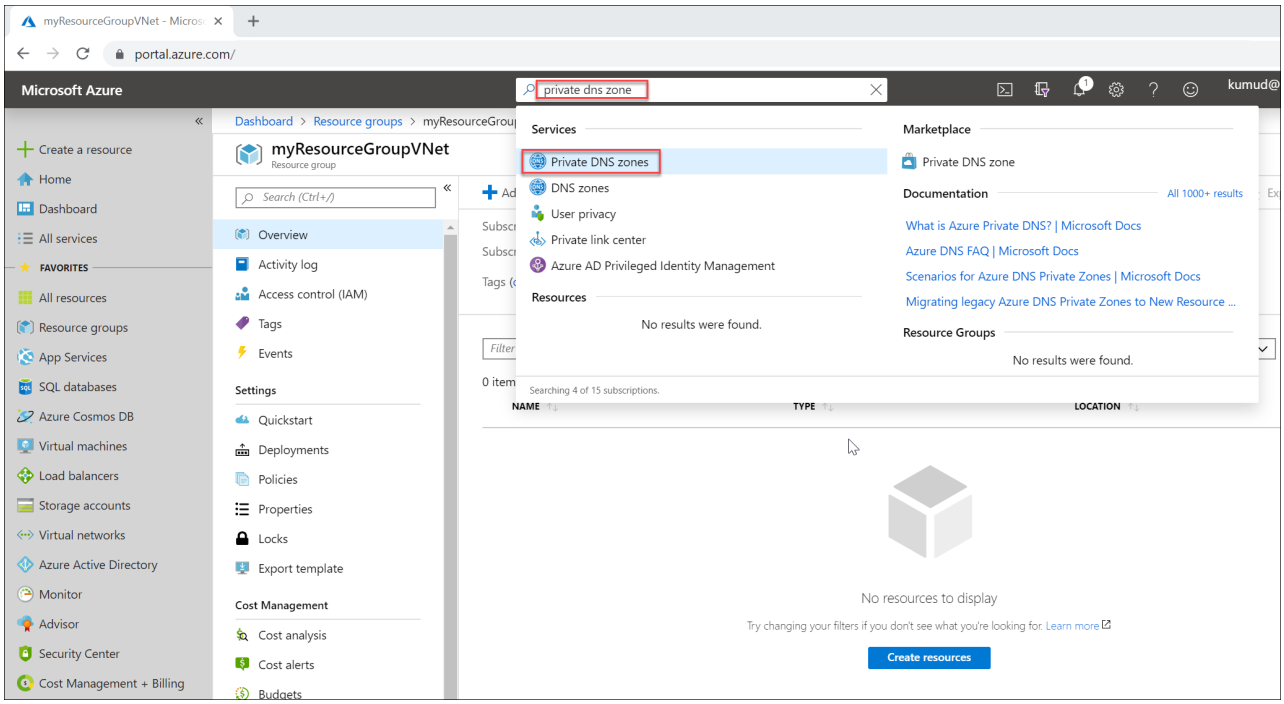
If you don't have an Azure subscription, create a free account before you begin.

If you prefer, you can complete this quickstart using Azure PowerShell or Azure CLI.

Create a private DNS zone

The following example creates a DNS zone called **private.contoso.com** in a resource group called **MyAzureResourceGroup**.

A DNS zone contains the DNS entries for a domain. To start hosting your domain in Azure DNS, you create a DNS zone for that domain name.



1. On the portal search bar, type **private dns zones** in the search text box and press **Enter**.
2. Select **Private DNS zone**.
3. Select **Create private dns zone**.
4. On the **Create Private DNS zone** page, type or select the following values:
   - **Resource group**: Select **Create new**, enter *MyAzureResourceGroup*, and select **OK**. The resource group name must be unique within the Azure subscription.
   - **Name**: Type *private.contoso.com* for this example.
5. For **Resource group location**, select **West Central US**.
6. Select **Review + Create**.
7. Select **Create**.

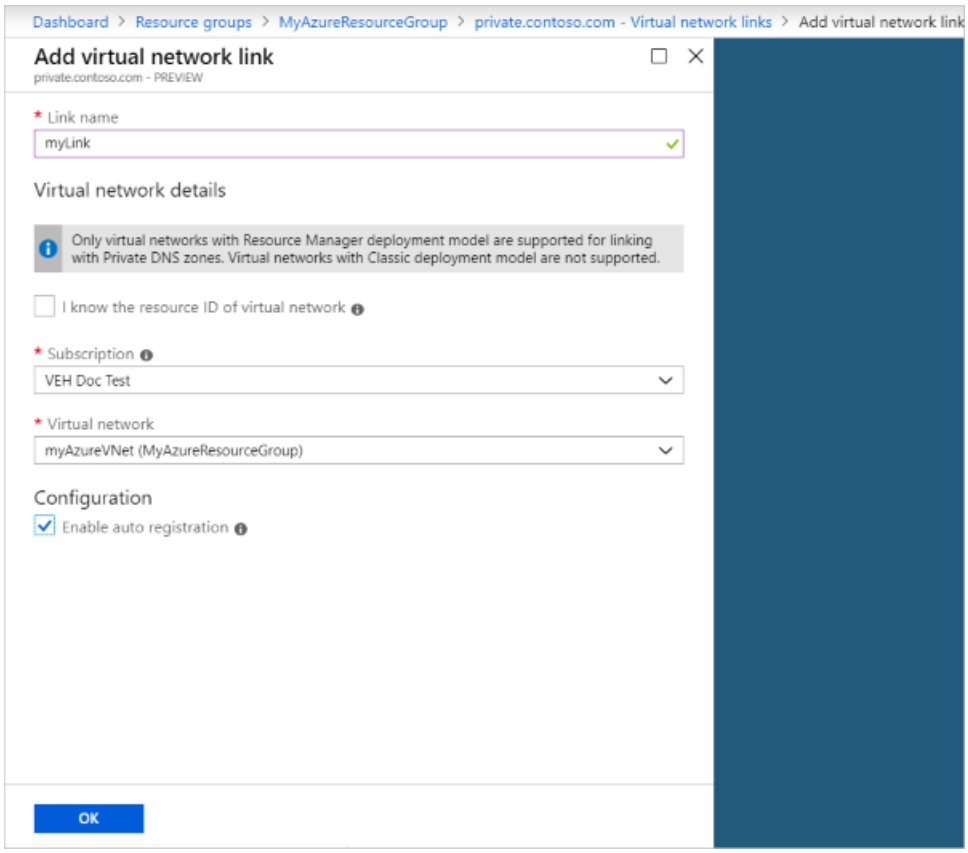
It may take a few minutes to create the zone.

Create a virtual network

1. On the portal page upper left, select **Create a resource**, then **Networking**, then select **Virtual network**.
2. For **Name**, type **myAzureVNet**.
3. For **Resource group**, select **MyAzureResourceGroup**.
4. For **Location**, select **West Central US**.
5. Accept the other default values and select **Create**.

Link the virtual network

To link the private DNS zone to a virtual network, you create a virtual network link.



1. Open the **MyAzureResourceGroup** resource group and select the **private.contoso.com** private zone.
2. On the left pane, select **Virtual network links**.
3. Select **Add**.
4. Type **myLink** for the **Link name**.
5. For **Virtual network**, select **myAzureVNet**.
6. Select the **Enable auto registration** check box.
7. Select **OK**.

Create the test virtual machines

Now, create two virtual machines so you can test your private DNS zone:

1. On the portal page upper left, select **Create a resource**, and then select **Windows Server 2016 Datacenter**.
2. Select **MyAzureResourceGroup** for the resource group.
3. Type **myVM01** - for the name of the virtual machine.
4. Select **West Central US** for the **Region**.
5. Type **azureadmin** for the administrator user name.
6. Type **Azure12345678** for the password and confirm the password.
7. For **Public inbound ports**, select **Allow selected ports**, and then select **RDP (3389)** for **Select inbound ports**.
8. Accept the other defaults for the page and then click **Next: Disks >**.
9. Accept the defaults on the **Disks** page, then click **Next: Networking >**.
10. Make sure that **myAzureVNet** is selected for the virtual network.

11. Accept the other defaults for the page, and then click **Next: Management >**.
12. For **Boot diagnostics**, select **Off**, accept the other defaults, and then select **Review + create**.
13. Review the settings and then click **Create**.

Repeat these steps and create another virtual machine named **myVM02**.

It will take a few minutes for both virtual machines to complete.

Create an additional DNS record

The following example creates a record with the relative name **db** in the DNS Zone **private.contoso.com**, in resource group **MyAzureResourceGroup**. The fully qualified name of the record set is **db.private.contoso.com**. The record type is "A", with the IP address of **myVM01**.

1. Open the **MyAzureResourceGroup** resource group and select the **private.contoso.com** private zone.
2. Select **+ Record set**.
3. For **Name**, type **db**.
4. For **IP Address**, type the IP address you see for **myVM01**. This should be auto registered when the virtual machine started.
5. Select **OK**.

Test the private zone

Now you can test the name resolution for your **private.contoso.com** private zone.

Configure VMs to allow inbound ICMP

You can use the ping command to test name resolution. So, configure the firewall on both virtual machines to allow inbound ICMP packets.

1. Connect to myVM01, and open a Windows PowerShell window with administrator privileges.
2. Run the following command:

```
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
```

Repeat for myVM02.

Ping the VMs by name

1. From the myVM02 Windows PowerShell command prompt, ping myVM01 using the automatically registered host name:

```
ping myVM01.private.contoso.com
```

   You should see output that looks similar to this:

```
PS C:\> ping myvm01.private.contoso.com

Pinging myvm01.private.contoso.com [10.2.0.4] with 32 bytes of data:
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
Reply from 10.2.0.4: bytes=32 time=1ms TTL=128
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.2.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
PS C:\>
```

2. Now ping the **db** name you created previously:

```
ping db.private.contoso.com
```

You should see output that looks similar to this:

```
PS C:\> ping db.private.contoso.com

Pinging db.private.contoso.com [10.2.0.4] with 32 bytes of data:
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

Ping statistics for 10.2.0.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\>
```

Delete all resources

When no longer needed, delete the **MyAzureResourceGroup** resource group to delete the resources created in this quickstart.

Next steps

Azure DNS Private Zones scenarios

Quickstart: Create an Azure private DNS zone using the Azure CLI

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

This quickstart walks you through the steps to create your first private DNS zone and record using the Azure CLI.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. To publish a private DNS zone to your virtual network, you specify the list of virtual networks that are allowed to resolve records within the zone. These are called *linked* virtual networks. When autoregistration is enabled, Azure DNS also updates the zone records whenever a virtual machine is created, changes its' IP address, or is deleted.

In this quickstart, you learn how to:

- Create a private DNS zone
- Create test virtual machines
- Create an additional DNS record
- Test the private zone

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

If you prefer, you can complete this quickstart using Azure PowerShell.

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.

Create the resource group

First, create a resource group to contain the DNS zone:

```
az group create --name MyAzureResourceGroup --location "East US"
```

Create a private DNS zone

The following example creates a virtual network named **myAzureVNet**. Then it creates a DNS zone named **private.contoso.com** in the **MyAzureResourceGroup** resource group, links the DNS zone to the **MyAzureVnet** virtual network, and enables automatic registration.

```
az network vnet create \
  --name myAzureVNet \
  --resource-group MyAzureResourceGroup \
  --location eastus \
  --address-prefix 10.2.0.0/16 \
  --subnet-name backendSubnet \
  --subnet-prefixes 10.2.0.0/24

az network private-dns zone create -g MyAzureResourceGroup \
   -n private.contoso.com

az network private-dns link vnet create -g MyAzureResourceGroup -n MyDNSLink \
   -z private.contoso.com -v myAzureVNet -e true
```

If you want to create a zone just for name resolution (no automatic hostname registration), you could use the `-e false` parameter.

List DNS private zones

To enumerate DNS zones, use `az network private-dns zone list`. For help, see `az network dns zone list --help`.

Specifying the resource group lists only those zones within the resource group:

```
az network private-dns zone list \
  -g MyAzureResourceGroup
```

Omitting the resource group lists all zones in the subscription:

```
az network private-dns zone list
```

Create the test virtual machines

Now, create two virtual machines so you can test your private DNS zone:

```
az vm create \
 -n myVM01 \
 --admin-username AzureAdmin \
 -g MyAzureResourceGroup \
 -l eastus \
 --subnet backendSubnet \
 --vnet-name myAzureVnet \
 --nsg NSG01 \
 --nsg-rule RDP \
 --image win2016datacenter

az vm create \
 -n myVM02 \
 --admin-username AzureAdmin \
 -g MyAzureResourceGroup \
 -l eastus \
 --subnet backendSubnet \
 --vnet-name myAzureVnet \
 --nsg NSG01 \
 --nsg-rule RDP \
 --image win2016datacenter
```

This will take a few minutes to complete.

Create an additional DNS record

To create a DNS record, use the `az network private-dns record-set [record type] add-record` command. For help with adding A records for example, see `az network private-dns record-set A add-record --help`.

The following example creates a record with the relative name **db** in the DNS Zone **private.contoso.com**, in resource group **MyAzureResourceGroup**. The fully qualified name of the record set is **db.private.contoso.com**. The record type is "A", with IP address "10.2.0.4".

```
az network private-dns record-set a add-record \
  -g MyAzureResourceGroup \
  -z private.contoso.com \
  -n db \
  -a 10.2.0.4
```

View DNS records

To list the DNS records in your zone, run:

```
az network private-dns record-set list \
  -g MyAzureResourceGroup \
  -z private.contoso.com
```

Test the private zone

Now you can test the name resolution for your **private.contoso.com** private zone.

Configure VMs to allow inbound ICMP

You can use the ping command to test name resolution. So, configure the firewall on both virtual machines to allow inbound ICMP packets.

1. Connect to myVM01, and open a Windows PowerShell window with administrator privileges.
2. Run the following command:

```
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
```

Repeat for myVM02.

Ping the VMs by name

1. From the myVM02 Windows PowerShell command prompt, ping myVM01 using the automatically registered host name:

   ```
   ping myVM01.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   PS C:\> ping myvm01.private.contoso.com

   Pinging myvm01.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

   Ping statistics for 10.2.0.4:
       Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
       Minimum = 0ms, Maximum = 1ms, Average = 0ms
   PS C:\>
   ```

2. Now ping the **db** name you created previously:

   ```
   ping db.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   PS C:\> ping db.private.contoso.com

   Pinging db.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

   Ping statistics for 10.2.0.4:
       Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
       Minimum = 0ms, Maximum = 0ms, Average = 0ms
   PS C:\>
   ```

Delete all resources

When no longer needed, delete the **MyAzureResourceGroup** resource group to delete the resources created in this quickstart.

```
az group delete --name MyAzureResourceGroup
```

Next steps

Quickstart: Create an Azure private DNS zone using Azure PowerShell

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

This article walks you through the steps to create your first private DNS zone and record using 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.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. To publish a private DNS zone to your virtual network, you specify the list of virtual networks that are allowed to resolve records within the zone. These are called *linked* virtual networks. When autoregistration is enabled, Azure DNS also updates the zone records whenever a virtual machine is created, changes its' IP address, or is deleted.

In this article, you learn how to:

- Create a private DNS zone
- Create test virtual machines
- Create an additional DNS record
- Test the private zone

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

If you prefer, you can complete this quickstart using Azure CLI.

Create the resource group

First, create a resource group to contain the DNS zone:

```
New-AzResourceGroup -name MyAzureResourceGroup -location "eastus"
```

Create a private DNS zone

A DNS zone is created by using the `New-AzPrivateDnsZone` cmdlet.

The following example creates a virtual network named **myAzureVNet**. Then it creates a DNS zone named **private.contoso.com** in the **MyAzureResourceGroup** resource group, links the DNS zone to the **MyAzureVnet** virtual network, and enables automatic registration.

```
Install-Module -Name Az.PrivateDns -force

$backendSubnet = New-AzVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix "10.2.0.0/24"
$vnet = New-AzVirtualNetwork `
  -ResourceGroupName MyAzureResourceGroup `
  -Location eastus `
  -Name myAzureVNet `
  -AddressPrefix 10.2.0.0/16 `
  -Subnet $backendSubnet

$zone = New-AzPrivateDnsZone -Name private.contoso.com -ResourceGroupName MyAzureResourceGroup

$link = New-AzPrivateDnsVirtualNetworkLink -ZoneName private.contoso.com `
  -ResourceGroupName MyAzureResourceGroup -Name "mylink" `
  -VirtualNetworkId $vnet.id -EnableRegistration
```

If you want to create a zone just for name resolution (no automatic hostname registration), you can omit the `-EnableRegistration` parameter.

List DNS private zones

By omitting the zone name from `Get-AzPrivateDnsZone`, you can enumerate all zones in a resource group. This operation returns an array of zone objects.

```
$zones = Get-AzPrivateDnsZone -ResourceGroupName MyAzureResourceGroup
$zones
```

By omitting both the zone name and the resource group name from `Get-AzPrivateDnsZone`, you can enumerate all zones in the Azure subscription.

```
$zones = Get-AzPrivateDnsZone
$zones
```

Create the test virtual machines

Now, create two virtual machines so you can test your private DNS zone:

```
New-AzVm `
    -ResourceGroupName "myAzureResourceGroup" `
    -Name "myVM01" `
    -Location "East US" `
    -subnetname backendSubnet `
    -VirtualNetworkName "myAzureVnet" `
    -addressprefix 10.2.0.0/24 `
    -OpenPorts 3389

New-AzVm `
    -ResourceGroupName "myAzureResourceGroup" `
    -Name "myVM02" `
    -Location "East US" `
    -subnetname backendSubnet `
    -VirtualNetworkName "myAzureVnet" `
    -addressprefix 10.2.0.0/24 `
    -OpenPorts 3389
```

This will take a few minutes to complete.

Create an additional DNS record

You create record sets by using the `New-AzPrivateDnsRecordSet` cmdlet. The following example creates a record with the relative name **db** in the DNS Zone **private.contoso.com**, in resource group **MyAzureResourceGroup**. The fully qualified name of the record set is **db.private.contoso.com**. The record type is "A", with IP address "10.2.0.4", and the TTL is 3600 seconds.

```
New-AzPrivateDnsRecordSet -Name db -RecordType A -ZoneName private.contoso.com `
   -ResourceGroupName MyAzureResourceGroup -Ttl 3600 `
   -PrivateDnsRecords (New-AzPrivateDnsRecordConfig -IPv4Address "10.2.0.4")
```

View DNS records

To list the DNS records in your zone, run:

```
Get-AzPrivateDnsRecordSet -ZoneName private.contoso.com -ResourceGroupName MyAzureResourceGroup
```

Test the private zone

Now you can test the name resolution for your **private.contoso.com** private zone.

Configure VMs to allow inbound ICMP

You can use the ping command to test name resolution. So, configure the firewall on both virtual machines to allow inbound ICMP packets.

1. Connect to myVM01, and open a Windows PowerShell window with administrator privileges.
2. Run the following command:

```
New-NetFirewallRule –DisplayName "Allow ICMPv4-In" –Protocol ICMPv4
```

Repeat for myVM02.

Ping the VMs by name

1. From the myVM02 Windows PowerShell command prompt, ping myVM01 using the automatically registered host name:

   ```
   ping myVM01.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   PS C:\> ping myvm01.private.contoso.com

   Pinging myvm01.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

   Ping statistics for 10.2.0.4:
       Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milli-seconds:
       Minimum = 0ms, Maximum = 1ms, Average = 0ms
   PS C:\>
   ```

2. Now ping the **db** name you created previously:

   ```
   ping db.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   PS C:\> ping db.private.contoso.com

   Pinging db.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

   Ping statistics for 10.2.0.4:
       Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
   Approximate round trip times in milliseconds:
       Minimum = 0ms, Maximum = 0ms, Average = 0ms
   PS C:\>
   ```

Delete all resources

When no longer needed, delete the **MyAzureResourceGroup** resource group to delete the resources created in this article.

```
Remove-AzResourceGroup -Name MyAzureResourceGroup
```

Next steps

Tutorial: Host your domain in Azure DNS

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

You can use Azure DNS to host your DNS domain and manage your DNS records. By hosting your domains in Azure, you can manage your DNS records by using the same credentials, APIs, tools, and billing as your other Azure services.

Suppose you buy the domain contoso.net from a domain name registrar and then create a zone with the name contoso.net in Azure DNS. Because you're the owner of the domain, your registrar offers you the option to configure the name server (NS) records for your domain. The registrar stores the NS records in the .net parent zone. Internet users around the world are then directed to your domain in your Azure DNS zone when they try to resolve DNS records in contoso.net.

In this tutorial, you learn how to:

- Create a DNS zone.
- Retrieve a list of name servers.
- Delegate the domain.
- Verify the delegation is working.

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

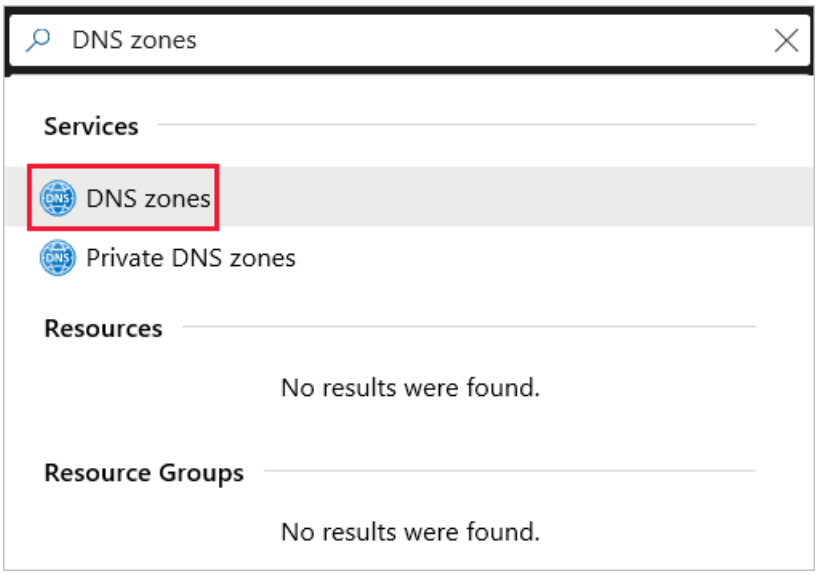
Prerequisites

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

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

Create a DNS zone

1. Go to the Azure portal to create a DNS zone. Search for and select **DNS zones**.



2. Select **Create DNS zone**.

3. On the **Create DNS zone** page, enter the following values, and then select **Create**:

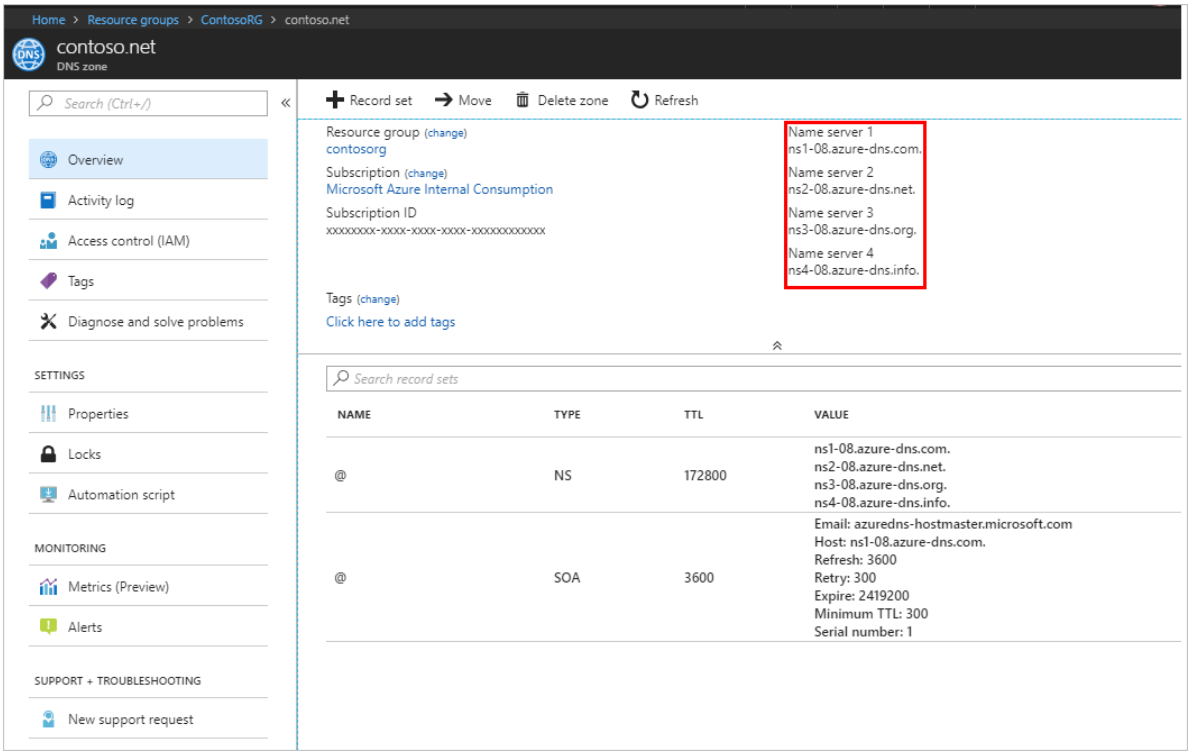
SETTING	VALUE	DETAILS
Name	[your domain name]	The domain name you bought. This tutorial uses contoso.net as an example.
Subscription	[Your subscription]	Select a subscription to create the zone in.
Resource group	**Create new:** contosoRG	Create a resource group. The resource group name must be unique within the subscription that you selected. The location of the resource group has no impact on the DNS zone. The DNS zone location is always "global," and isn't shown.
Location	East US	

Retrieve name servers

Before you can delegate your DNS zone to Azure DNS, you need to know the name servers for your zone. Azure DNS allocates name servers from a pool each time a zone is created.

1. With the DNS zone created, in the Azure portal **Favorites** pane, select **All resources**. On the **All resources** page, select your DNS zone. If the subscription that you selected already has several resources in it, you can enter your domain name in the **Filter by name** box to easily access the application gateway.

2. Retrieve the name servers from the DNS zone page. In this example, the zone contoso.net has been assigned name servers *ns1-01.azure-dns.com*, *ns2-01.azure-dns.net*, *ns3-01.azure-dns.org*, and *ns4-01.azure-dns.info*:



Azure DNS automatically creates authoritative NS records in your zone for the assigned name servers.

Delegate the domain

Now that the DNS zone is created and you have the name servers, you need to update the parent domain with the Azure DNS name servers. Each registrar has its own DNS management tools to change the name server records for a domain.

1. In the registrar's DNS management page, edit the NS records and replace the NS records with the Azure DNS name servers.
2. When you delegate a domain to Azure DNS, you must use the name servers that Azure DNS provides. Use all four name servers, regardless of the name of your domain. Domain delegation doesn't require a name server to use the same top-level domain as your domain.

> **NOTE**
> When you copy each name server address, make sure you copy the trailing period at the end of the address. The trailing period indicates the end of a fully qualified domain name. Some registrars append the period if the NS name doesn't have it at the end. To be compliant with the DNS RFC, include the trailing period.

Delegations that use name servers in your own zone, sometimes called *vanity name servers*, aren't currently supported in Azure DNS.

Verify the delegation

After you complete the delegation, you can verify that it's working by using a tool such as *nslookup* to query the Start of Authority (SOA) record for your zone. The SOA record is automatically created when the zone is created. You might need to wait 10 minutes or more after you complete the delegation, before you can successfully verify that it's working. It can take a while for changes to propagate through the DNS system.

You don't have to specify the Azure DNS name servers. If the delegation is set up correctly, the normal DNS resolution process finds the name servers automatically.

1. From a command prompt, enter an nslookup command similar to the following example:

```
nslookup -type=SOA contoso.net
```

2. Verify that your response looks similar to the following nslookup output:

```
Server: ns1-04.azure-dns.com
Address: 208.76.47.4

contoso.net
primary name server = ns1-04.azure-dns.com
responsible mail addr = msnhst.microsoft.com
serial = 1
refresh = 900 (15 mins)
retry = 300 (5 mins)
expire = 604800 (7 days)
default TTL = 300 (5 mins)
```

Clean up resources

You can keep the **contosoRG** resource group if you intend to do the next tutorial. Otherwise, delete the **contosoRG** resource group to delete the resources created in this tutorial.

- Select the **contosoRG** resource group, and then select **Delete resource group**.

Next steps

In this tutorial, you created a DNS zone for your domain and delegated it to Azure DNS. To learn about Azure DNS and web apps, continue with the tutorial for web apps.

Create DNS records for a web app in a custom domain

Tutorial: Create DNS records in a custom domain for a web app

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

You can configure Azure DNS to host a custom domain for your web apps. For example, you can create an Azure web app and have your users access it using either www.contoso.com or contoso.com as a fully qualified domain name (FQDN).

> **NOTE**
> Contoso.com is used as an example throughout this tutorial. Substitute your own domain name for contoso.com.

To do this, you have to create three records:

- A root "A" record pointing to contoso.com
- A root "TXT" record for verification
- A "CNAME" record for the www name that points to the A record

Keep in mind that if you create an A record for a web app in Azure, the A record must be manually updated if the underlying IP address for the web app changes.

In this tutorial, you learn how to:

- Create an A and TXT record for your custom domain
- Create a CNAME record for your custom domain
- Test the new records
- Add custom host names you your web app
- Test the custom host names

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

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.	

OPTION	EXAMPLE/LINK
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.

Prerequisites

> **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 must have a domain name available to test with that you can host in Azure DNS . You must have full control of this domain. Full control includes the ability to set the name server (NS) records for the domain.
- Create an App Service app, or use an app that you created for another tutorial.
- Create a DNS zone in Azure DNS, and delegate the zone in your registrar to Azure DNS.
  1. To create a DNS zone, follow the steps in Create a DNS zone.
  2. To delegate your zone to Azure DNS, follow the steps in DNS domain delegation.

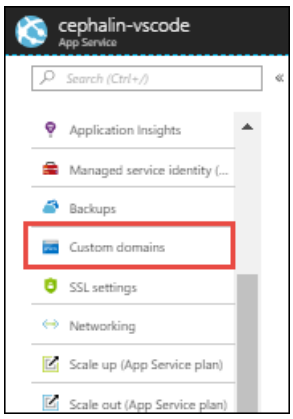
After creating a zone and delegating it to Azure DNS, you can then create records for your custom domain.

Create an A record and TXT record

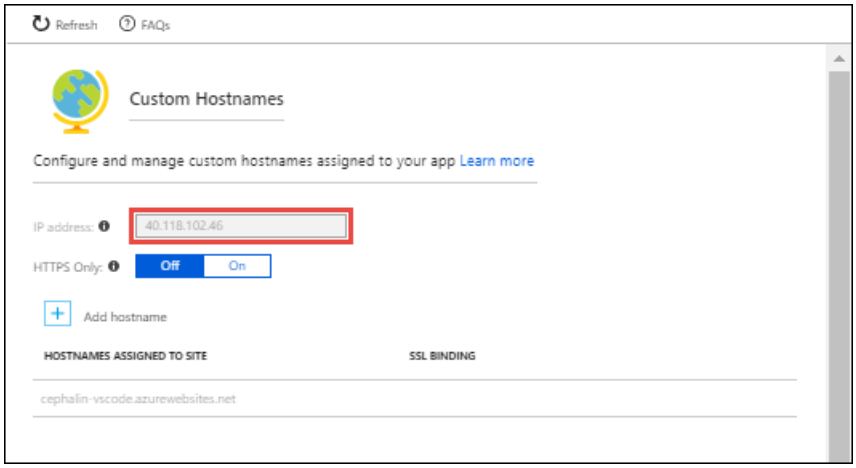
An A record is used to map a name to its IP address. In the following example, assign "@" as an A record using your web app IPv4 address. @ typically represents the root domain.

Get the IPv4 address

In the left navigation of the App Services page in the Azure portal, select **Custom domains**.



In the **Custom domains** page, copy the app's IPv4 address:



Create the A record

```
New-AzDnsRecordSet -Name "@" -RecordType "A" -ZoneName "contoso.com" `
 -ResourceGroupName "MyAzureResourceGroup" -Ttl 600 `
 -DnsRecords (New-AzDnsRecordConfig -IPv4Address "<your web app IP address>")
```

Create the TXT record

App Services uses this record only at configuration time to verify that you own the custom domain. You can delete this TXT record after your custom domain is validated and configured in App Service.

> **NOTE**
> If you want to verify the domain name, but not route production traffic to the web app, you only need to specify the TXT record for the verification step. Verification does not require an A or CNAME record in addition to the TXT record.

```
New-AzDnsRecordSet -ZoneName contoso.com -ResourceGroupName MyAzureResourceGroup `
 -Name "@" -RecordType "txt" -Ttl 600 `
 -DnsRecords (New-AzDnsRecordConfig -Value  "contoso.azurewebsites.net")
```

Create the CNAME record

If your domain is already managed by Azure DNS (see DNS domain delegation, you can use the following example to create a CNAME record for contoso.azurewebsites.net.

Open Azure PowerShell and create a new CNAME record. This example creates a record set type CNAME with a "time to live" of 600 seconds in DNS zone named "contoso.com" with the alias for the web app contoso.azurewebsites.net.

Create the record

```
New-AzDnsRecordSet -ZoneName contoso.com -ResourceGroupName "MyAzureResourceGroup" `
 -Name "www" -RecordType "CNAME" -Ttl 600 `
 -DnsRecords (New-AzDnsRecordConfig -cname "contoso.azurewebsites.net")
```

The following example is the response:

```
    Name              : www
    ZoneName          : contoso.com
    ResourceGroupName : myresourcegroup
    Ttl               : 600
    Etag              : 8baceeb9-4c2c-4608-a22c-229923ee185
    RecordType        : CNAME
    Records           : {contoso.azurewebsites.net}
    Tags              : {}
```

Test the new records

You can validate the records were created correctly by querying the "www.contoso.com" and "contoso.com" using nslookup, as shown below:

```
PS C:\> nslookup
Default Server:  Default
Address:  192.168.0.1

> www.contoso.com
Server:  default server
Address:  192.168.0.1

Non-authoritative answer:
Name:    <instance of web app service>.cloudapp.net
Address:  <ip of web app service>
Aliases:  www.contoso.com
contoso.azurewebsites.net
<instance of web app service>.vip.azurewebsites.windows.net

> contoso.com
Server:  default server
Address:  192.168.0.1

Non-authoritative answer:
Name:    contoso.com
Address:  <ip of web app service>

> set type=txt
> contoso.com

Server:  default server
Address:  192.168.0.1

Non-authoritative answer:
contoso.com text =

        "contoso.azurewebsites.net"
```

Add custom host names

Now you can add the custom host names to your web app:

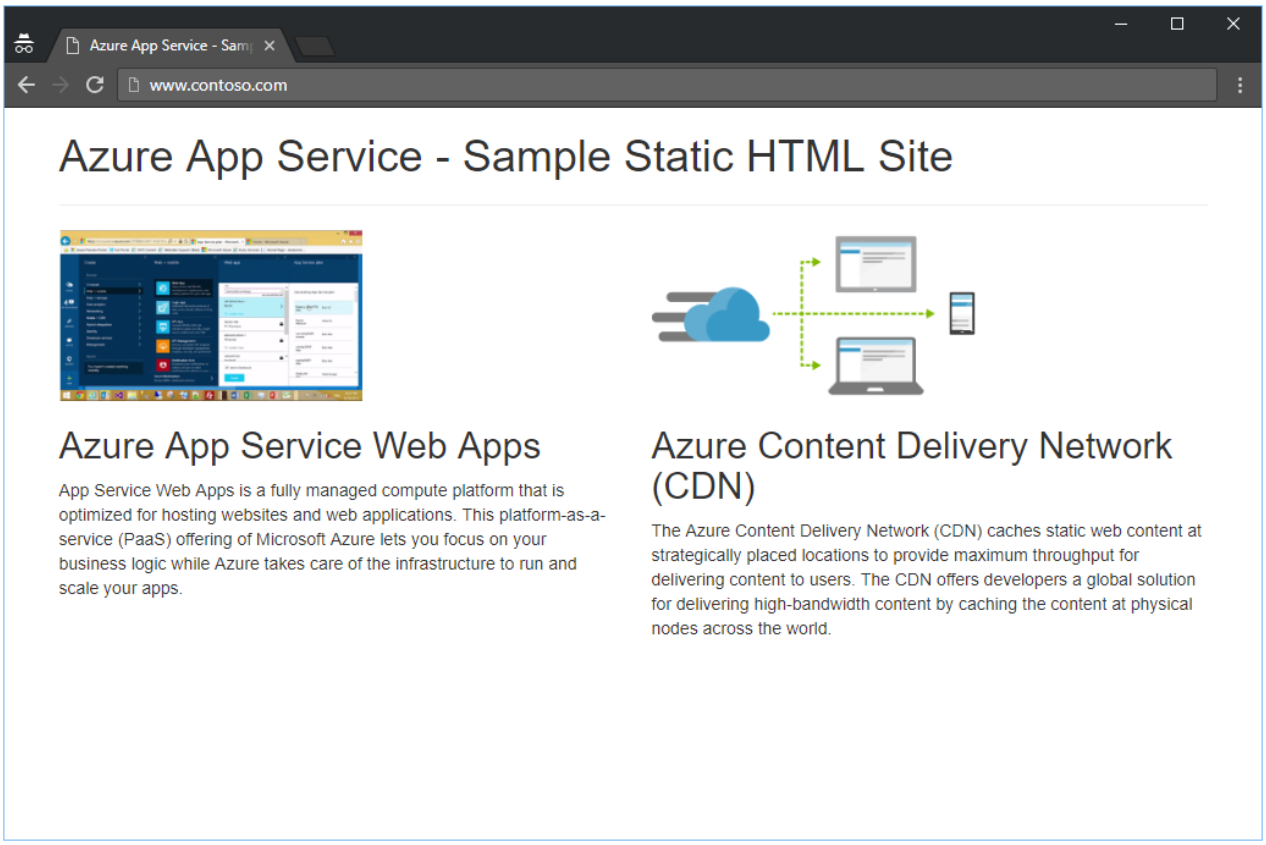
```
set-AzWebApp `
 -Name contoso `
 -ResourceGroupName MyAzureResourceGroup `
 -HostNames @("contoso.com","www.contoso.com","contoso.azurewebsites.net")
```

Test the custom host names

Open a browser and browse to `http://www.<your domainname>` and `http://<you domain name>`.

> **NOTE**
> Make sure you include the `http://` prefix, otherwise your browser may attempt to predict a URL for you!

You should see the same page for both URLs. For example:



Clean up resources

When you no longer need the resources created in this tutorial, you can delete the **myresourcegroup** resource group.

Next steps

Learn how to create Azure DNS private zones.

Get started with Azure DNS private zones using PowerShell

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

Tutorial: Configure an alias record to refer to an Azure public IP address

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

In this tutorial, you learn how to:

- Create a network infrastructure.
- Create a web server virtual machine.
- 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 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-pip**.
3. Select **Create a resource** > **Networking** > **Virtual network**.
4. Create a virtual network named **VNet-Server**. Place it in the **RG-DNS-Alias-pip** resource group, and name the subnet **SN-Web**.

Create a web server virtual machine

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 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. For public inbound ports, select **HTTP** > **HTTPS** > **RDP (3389)**, and then select **OK**.
5. On the **Summary** page, select **Create**.

This procedure takes a few minutes to finish.

Install IIS

Install IIS on **Web-01**.

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 then select **Next**.
5. Select **Next** four times, and then select **Install**. This procedure takes a few minutes to finish.
6. After the installation finishes, select **Close**.
7. Open a web browser. Browse to **localhost** to verify that the default IIS web page appears.

Create an alias record

Create an alias record that points to the public IP address.

1. Select your Azure DNS zone to open the zone.
2. Select **Record set**.
3. In the **Name** text box, select **web01**.
4. Leave the **Type** as an **A** record.
5. Select the **Alias Record Set** check box.
6. Select **Choose Azure service**, and then select the **Web-01-ip** public IP address.

Test the alias record

1. In the **RG-DNS-Alias-pip** resource group, select the **Web-01** virtual machine. Note the public IP address.
2. From a web browser, browse to the fully qualified domain name for the Web01-01 virtual machine. An example is **web01.contoso.com**. You now see the IIS default web page.
3. Close the web browser.
4. Stop the **Web-01** virtual machine, and then restart it.
5. After the virtual machine restarts, note the new public IP address for the virtual machine.
6. Open a new browser. Browse again to the fully qualified domain name for the Web01-01 virtual machine. An example is **web01.contoso.com**.

This procedure succeeds because you used an alias record to point to the public IP address resource, not a standard A record.

Clean up resources

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

Next steps

In this tutorial, you created an alias record to refer to an Azure public IP address. To learn about Azure DNS and web apps, continue with the tutorial for web apps.

Create DNS records for a web app in a custom domain

Tutorial: Create an alias record to refer to a zone resource record

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

Alias records can reference other record sets of the same type. For example, you can have a DNS CNAME record set be an alias to another CNAME record set of the same type. This capability is useful if you want to have some record sets as aliases and some as non-aliases in terms of behavior.

In this tutorial, you learn how to:

- Create an alias record for a resource record in the zone.
- 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 to host your domain in Azure DNS, see Tutorial: Host your domain in Azure DNS.

Create an alias record

Create an alias record that points to a resource record in the zone.

Create the target resource record

1. Select your Azure DNS zone to open the zone.
2. Select **Record set**.
3. In the **Name** text box, enter **server**.
4. For the **Type**, select **A**.
5. In the **IP ADDRESS** text box, enter **10.10.10.10**.
6. Select **OK**.

Create the alias record

1. Select your Azure DNS zone to open the zone.
2. Select **Record set**.
3. In the **Name** text box, enter **test**.
4. For the **Type**, select **A**.
5. Select **Yes** in the **Alias Record Set** check box. Then select the **Zone record set** option.
6. For the **Zone record set**, select the **server** record.
7. Select **OK**.

Test the alias record

1. Start your favorite nslookup tool. One option is to browse to https://network-tools.com/nslook.
2. Set the query type for A records, and look up **test.<your domain name>**. The answer is **10.10.10.10**.
3. In the Azure portal, change the **server** A record to **10.11.11.11**.
4. Wait a few minutes, and then use nslookup again for the **test** record. The answer is **10.11.11.11**.

Clean up resources

When you no longer need the resources created for this tutorial, delete the **server** and **test** resource records in your zone.

Next steps

In this tutorial, you created an alias record to refer to a resource record within the zone. To learn about Azure DNS and web apps, continue with the tutorial for web apps.

Create DNS records for a web app in a custom domain

Overview of DNS zones and records

2/1/2020 • 12 minutes to read • Edit Online

This page explains the key concepts of domains, DNS zones, and DNS records and record sets, and how they are supported in Azure DNS.

Domain names

The Domain Name System is a hierarchy of domains. The hierarchy starts from the 'root' domain, whose name is simply '.'. Below this come top-level domains, such as 'com', 'net', 'org', 'uk' or 'jp'. Below these are second-level domains, such as 'org.uk' or 'co.jp'. The domains in the DNS hierarchy are globally distributed, hosted by DNS name servers around the world.

A domain name registrar is an organization that allows you to purchase a domain name, such as `contoso.com`. Purchasing a domain name gives you the right to control the DNS hierarchy under that name, for example allowing you to direct the name `www.contoso.com` to your company web site. The registrar may host the domain in its own name servers on your behalf, or allow you to specify alternative name servers.

Azure DNS provides a globally distributed, high-availability name server infrastructure, which you can use to host your domain. By hosting your domains in Azure DNS, you can manage your DNS records with the same credentials, APIs, tools, billing, and support as your other Azure services.

Azure DNS does not currently support purchasing of domain names. If you want to purchase a domain name, you need to use a third-party domain name registrar. The registrar typically charges a small annual fee. The domains can then be hosted in Azure DNS for management of DNS records. See Delegate a Domain to Azure DNS for details.

DNS zones

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone.

For example, the domain 'contoso.com' may contain several DNS records, such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site).

When creating a DNS zone in Azure DNS:

- The name of the zone must be unique within the resource group, and the zone must not exist already. Otherwise, the operation fails.
- The same zone name can be reused in a different resource group or a different Azure subscription.
- Where multiple zones share the same name, each instance is assigned different name server addresses. Only one set of addresses can be configured with the domain name registrar.

> **NOTE**
>
> You do not have to own a domain name to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to configure the Azure DNS name servers as the correct name servers for the domain name with the domain name registrar.
>
> For more information, see Delegate a domain to Azure DNS.

DNS records

Record names

In Azure DNS, records are specified by using relative names. A *fully qualified* domain name (FQDN) includes the zone name, whereas a *relative* name does not. For example, the relative record name `www` in the zone `contoso.com` gives the fully qualified record name `www.contoso.com`.

An *apex* record is a DNS record at the root (or *apex*) of a DNS zone. For example, in the DNS zone `contoso.com`, an apex record also has the fully qualified name `contoso.com` (this is sometimes called a *naked* domain). By convention, the relative name '@' is used to represent apex records.

Record types

Each DNS record has a name and a type. Records are organized into various types according to the data they contain. The most common type is an 'A' record, which maps a name to an IPv4 address. Another common type is an 'MX' record, which maps a name to a mail server.

Azure DNS supports all common DNS record types: A, AAAA, CAA, CNAME, MX, NS, PTR, SOA, SRV, and TXT. Note that SPF records are represented using TXT records.

Record sets

Sometimes you need to create more than one DNS record with a given name and type. For example, suppose the 'www.contoso.com' web site is hosted on two different IP addresses. The website requires two different A records, one for each IP address. Here is an example of a record set:

```
www.contoso.com.        3600    IN    A    134.170.185.46
www.contoso.com.        3600    IN    A    134.170.188.221
```

Azure DNS manages all DNS records using *record sets*. A record set (also known as a *resource* record set) is the collection of DNS records in a zone that have the same name and are of the same type. Most record sets contain a single record. However, examples like the one above, in which a record set contains more than one record, are not uncommon.

For example, suppose you have already created an A record 'www' in the zone 'contoso.com', pointing to the IP address '134.170.185.46' (the first record above). To create the second record you would add that record to the existing record set, rather than create an additional record set.

The SOA and CNAME record types are exceptions. The DNS standards don't permit multiple records with the same name for these types, therefore these record sets can only contain a single record.

Time-to-live

The time to live, or TTL, specifies how long each record is cached by clients before being requeried. In the above example, the TTL is 3600 seconds or 1 hour.

In Azure DNS, the TTL is specified for the record set, not for each record, so the same value is used for all records within that record set. You can specify any TTL value between 1 and 2,147,483,647 seconds.

Wildcard records

Azure DNS supports wildcard records. Wildcard records are returned in response to any query with a matching name (unless there is a closer match from a non-wildcard record set). Azure DNS supports wildcard record sets for all record types except NS and SOA.

To create a wildcard record set, use the record set name '*'. Alternatively, you can also use a name with '*' as its left-most label, for example, '*.foo'.

CAA records

CAA records allow domain owners to specify which Certificate Authorities (CAs) are authorized to issue certificates for their domain. This allows CAs to avoid mis-issuing certificates in some circumstances. CAA records have three properties:

- **Flags**: This is an integer between 0 and 255, used to represent the critical flag that has special meaning per the RFC
- **Tag**: an ASCII string that can be one of the following:
  - **issue**: use this if you want to specify CAs that are permitted to issue certs (all types)
  - **issuewild**: use this if you want to specify CAs that are permitted to issue certs (wildcard certs only)
  - **iodef**: specify an email address or hostname to which CAs can notify for unauthorized cert issue requests
- **Value**: the value for the specific Tag chosen

CNAME records

CNAME record sets cannot coexist with other record sets with the same name. For example, you cannot create a CNAME record set with the relative name 'www' and an A record with the relative name 'www' at the same time.

Because the zone apex (name = '@') always contains the NS and SOA record sets that were created when the zone was created, you can't create a CNAME record set at the zone apex.

These constraints arise from the DNS standards and are not limitations of Azure DNS.

NS records

The NS record set at the zone apex (name '@') is created automatically with each DNS zone, and is deleted automatically when the zone is deleted (it cannot be deleted separately).

This record set contains the names of the Azure DNS name servers assigned to the zone. You can add additional name servers to this NS record set, to support co-hosting domains with more than one DNS provider. You can also modify the TTL and metadata for this record set. However, you cannot remove or modify the pre-populated Azure DNS name servers.

This applies only to the NS record set at the zone apex. Other NS record sets in your zone (as used to delegate child zones) can be created, modified, and deleted without constraint.

SOA records

A SOA record set is created automatically at the apex of each zone (name = '@'), and is deleted automatically when the zone is deleted. SOA records cannot be created or deleted separately.

You can modify all properties of the SOA record except for the 'host' property, which is pre-configured to refer to the primary name server name provided by Azure DNS.

The zone serial number in the SOA record is not updated automatically when changes are made to the records in the zone. It can be updated manually by editing the SOA record, if necessary.

SPF records

Sender policy framework (SPF) records are used to specify which email servers can send email on behalf of a domain name. Correct configuration of SPF records is important to prevent recipients from marking your email as junk.

The DNS RFCs originally introduced a new SPF record type to support this scenario. To support older name servers, they also allowed the use of the TXT record type to specify SPF records. This ambiguity led to confusion, which was resolved by RFC 7208. It states that SPF records must be created by using the TXT record type. It also states that the SPF record type is deprecated.

SPF records are supported by Azure DNS and must be created by using the TXT record type. The obsolete SPF record type isn't supported. When you import a DNS zone file, any SPF records that use the SPF

record type are converted to the TXT record type.

SRV records

SRV records are used by various services to specify server locations. When specifying an SRV record in Azure DNS:

- The *service* and *protocol* must be specified as part of the record set name, prefixed with underscores. For example, '_sip._tcp.name'. For a record at the zone apex, there is no need to specify '@' in the record name, simply use the service and protocol, for example '_sip._tcp'.
- The *priority*, *weight*, *port*, and *target* are specified as parameters of each record in the record set.

TXT records

TXT records are used to map domain names to arbitrary text strings. They are used in multiple applications, in particular related to email configuration, such as the Sender Policy Framework (SPF) and DomainKeys Identified Mail (DKIM).

The DNS standards permit a single TXT record to contain multiple strings, each of which may be up to 254 characters in length. Where multiple strings are used, they are concatenated by clients and treated as a single string.

When calling the Azure DNS REST API, you need to specify each TXT string separately. When using the Azure portal, PowerShell or CLI interfaces you should specify a single string per record, which is automatically divided into 254-character segments if necessary.

The multiple strings in a DNS record should not be confused with the multiple TXT records in a TXT record set. A TXT record set can contain multiple records, *each of which* can contain multiple strings. Azure DNS supports a total string length of up to 1024 characters in each TXT record set (across all records combined).

Tags and metadata

Tags

Tags are a list of name-value pairs and are used by Azure Resource Manager to label resources. Azure Resource Manager uses tags to enable filtered views of your Azure bill, and also enables you to set a policy on which tags are required. For more information about tags, see Using tags to organize your Azure resources.

Azure DNS supports using Azure Resource Manager tags on DNS zone resources. It does not support tags on DNS record sets, although as an alternative 'metadata' is supported on DNS record sets as explained below.

Metadata

As an alternative to record set tags, Azure DNS supports annotating record sets using 'metadata'. Similar to tags, metadata enables you to associate name-value pairs with each record set. This can be useful, for example to record the purpose of each record set. Unlike tags, metadata cannot be used to provide a filtered view of your Azure bill and cannot be specified in an Azure Resource Manager policy.

Etags

Suppose two people or two processes try to modify a DNS record at the same time. Which one wins? And does the winner know that they've overwritten changes created by someone else?

Azure DNS uses Etags to handle concurrent changes to the same resource safely. Etags are separate from Azure Resource Manager 'Tags'. Each DNS resource (zone or record set) has an Etag associated with it. Whenever a resource is retrieved, its Etag is also retrieved. When updating a resource, you can choose to pass back the Etag so Azure DNS can verify that the Etag on the server matches. Since each update to a resource results in the Etag being regenerated, an Etag mismatch indicates a concurrent change has occurred. Etags can also be used when

creating a new resource to ensure that the resource does not already exist.

By default, Azure DNS PowerShell uses Etags to block concurrent changes to zones and record sets. The optional *-Overwrite* switch can be used to suppress Etag checks, in which case any concurrent changes that have occurred are overwritten.

At the level of the Azure DNS REST API, Etags are specified using HTTP headers. Their behavior is given in the following table:

HEADER	BEHAVIOR
None	PUT always succeeds (no Etag checks)
If-match <etag>	PUT only succeeds if resource exists and Etag matches
If-match *	PUT only succeeds if resource exists
If-none-match *	PUT only succeeds if resource does not exist

Limits

The following default limits apply when using Azure DNS:

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
Number of DNS queries a virtual machine can send to Azure DNS resolver, per second	500 [2]

RESOURCE	DEFAULT LIMIT
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.

Next steps

- To start using Azure DNS, learn how to create a DNS zone and create DNS records.
- To migrate an existing DNS zone, learn how to import and export a DNS zone file.

Azure DNS alias records overview

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

Azure DNS alias records are qualifications on a DNS record set. They can reference other Azure resources from within your DNS zone. For example, you can create an alias record set that references an Azure public IP address instead of an A record. Your alias record set points to an Azure public IP address service instance dynamically. As a result, the alias record set seamlessly updates itself during DNS resolution.

An alias record set is supported for the following record types in an Azure DNS zone:

- A
- AAAA
- CNAME

> **NOTE**
> If you intend to use an alias record for the A or AAAA record types to point to an Azure Traffic Manager profile you must make sure that the Traffic Manager profile has only external endpoints. You must provide the IPv4 or IPv6 address for external endpoints in Traffic Manager. You can't use fully-qualified domain names (FQDNs) in endpoints. Ideally, use static IP addresses.

Capabilities

- **Point to a public IP resource from a DNS A/AAAA record set.** You can create an A/AAAA record set and make it an alias record set to point to a public IP resource (standard or basic). The DNS record set changes automatically if the public IP address changes or is deleted. Dangling DNS records that point to incorrect IP addresses are avoided.

  There is a current limit of 20 alias records sets per resource.

- **Point to a Traffic Manager profile from a DNS A/AAAA/CNAME record set.** You can create an A/AAAA or CNAME record set and use alias records to point it to a Traffic Manager profile. It's especially useful when you need to route traffic at a zone apex, as traditional CNAME records aren't supported for a zone apex. For example, say your Traffic Manager profile is myprofile.trafficmanager.net and your business DNS zone is contoso.com. You can create an alias record set of type A/AAAA for contoso.com (the zone apex) and point to myprofile.trafficmanager.net.

- **Point to an Azure Content Delivery Network (CDN) endpoint**. This is useful when you create static websites using Azure storage and Azure CDN.

- **Point to another DNS record set within the same zone.** Alias records can reference other record sets of the same type. For example, a DNS CNAME record set can be an alias to another CNAME record set. This arrangement is useful if you want some record sets to be aliases and some non-aliases.

Scenarios

There are a few common scenarios for Alias records.

Prevent dangling DNS records

A common problem with traditional DNS records is dangling records. For example, DNS records that haven't been updated to reflect changes to IP addresses. The issue occurs especially with A/AAAA or CNAME record types.

With a traditional DNS zone record, if the target IP or CNAME no longer exists, the DNS record associated with it must be manually updated. In some organizations, a manual update might not happen in time because of process issues or the separation of roles and associated permission levels. For example, a role might have the authority to delete a CNAME or IP address that belongs to an application. But it doesn't have sufficient authority to update the DNS record that points to those targets. A delay in updating the DNS record can potentially cause an outage for the users.

Alias records prevent dangling references by tightly coupling the life cycle of a DNS record with an Azure resource. For example, consider a DNS record that's qualified as an alias record to point to a public IP address or a Traffic Manager profile. If you delete those underlying resources, the DNS alias record becomes an empty record set. It no longer references the deleted resource.

Update DNS record-set automatically when application IP addresses change

This scenario is similar to the previous one. Perhaps an application is moved, or the underlying virtual machine is restarted. An alias record then updates automatically when the IP address changes for the underlying public IP resource. This avoids potential security risks of directing the users to another application that has been assigned the old public IP address.

Host load-balanced applications at the zone apex

The DNS protocol prevents the assignment of CNAME records at the zone apex. For example if your domain is contoso.com; you can create CNAME records for somelabel.contoso.com; but you can't create CNAME for contoso.com itself. This restriction presents a problem for application owners who have load-balanced applications behind Azure Traffic Manager. Since using a Traffic Manager profile requires creation of a CNAME record, it isn't possible to point at the Traffic Manager profile from the zone apex.

This problem is solved using alias records. Unlike CNAME records, alias records are created at the zone apex and application owners can use it to point their zone apex record to a Traffic Manager profile that has external endpoints. Application owners point to the same Traffic Manager profile that's used for any other domain within their DNS zone.

For example, contoso.com and www.contoso.com can point to the same Traffic Manager profile. To learn more about using alias records with Azure Traffic Manager profiles, see the Next steps section.

Point zone apex to Azure CDN endpoints

Just like a Traffic Manager profile, you can also use alias records to point your DNS zone apex to Azure CDN endpoints. This is useful when you create static websites using Azure storage and Azure CDN. You can then access the website without prepending "www" to your DNS name.

For example, if your static website is named www.contoso.com, your users can access your site using contoso.com without the need to prepend www to the DNS name.

As described previously, CNAME records aren't supported at the zone apex. So, you can't use a CNAME record to point contoso.com to your CDN endpoint. Instead, you can use an alias record to point the zone apex to a CDN endpoint directly.

> **NOTE**
> Pointing a zone apex to CDN endpoints for Azure CDN from Akamai is currently not supported.

Next steps

To learn more about alias records, see the following articles:

- Tutorial: Configure an alias record to refer to an Azure public IP address
- Tutorial: Configure an alias record to support apex domain names with Traffic Manager

- DNS FAQ

Delegation of DNS zones with Azure DNS

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

Azure DNS allows you to host a DNS zone and manage the DNS records for a domain in Azure. In order for DNS queries for a domain to reach Azure DNS, the domain has to be delegated to Azure DNS from the parent domain. Keep in mind Azure DNS is not the domain registrar. This article explains how domain delegation works and how to delegate domains to Azure DNS.

How DNS delegation works

Domains and zones

The Domain Name System is a hierarchy of domains. The hierarchy starts from the 'root' domain, whose name is simply '.'. Below this come top-level domains, such as 'com', 'net', 'org', 'uk' or 'jp'. Below these top-level domains are second-level domains, such as 'org.uk' or 'co.jp'. And so on. The domains in the DNS hierarchy are hosted using separate DNS zones. These zones are globally distributed, hosted by DNS name servers around the world.

DNS zone - A domain is a unique name in the Domain Name System, for example 'contoso.com'. A DNS zone is used to host the DNS records for a particular domain. For example, the domain 'contoso.com' may contain several DNS records such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a website).

Domain registrar - A domain registrar is a company who can provide Internet domain names. They verify if the Internet domain you want to use is available and allow you to purchase it. Once the domain name is registered, you are the legal owner for the domain name. If you already have an Internet domain, you will use the current domain registrar to delegate to Azure DNS.

For more information about accredited domain registrars, see ICANN-Accredited Registrars.

Resolution and delegation

There are two types of DNS servers:

- An *authoritative* DNS server hosts DNS zones. It answers DNS queries for records in those zones only.
- A *recursive* DNS server does not host DNS zones. It answers all DNS queries by calling authoritative DNS servers to gather the data it needs.

Azure DNS provides an authoritative DNS service. It does not provide a recursive DNS service. Cloud Services and VMs in Azure are automatically configured to use a recursive DNS service that is provided separately as part of Azure's infrastructure. For information on how to change these DNS settings, see Name Resolution in Azure.

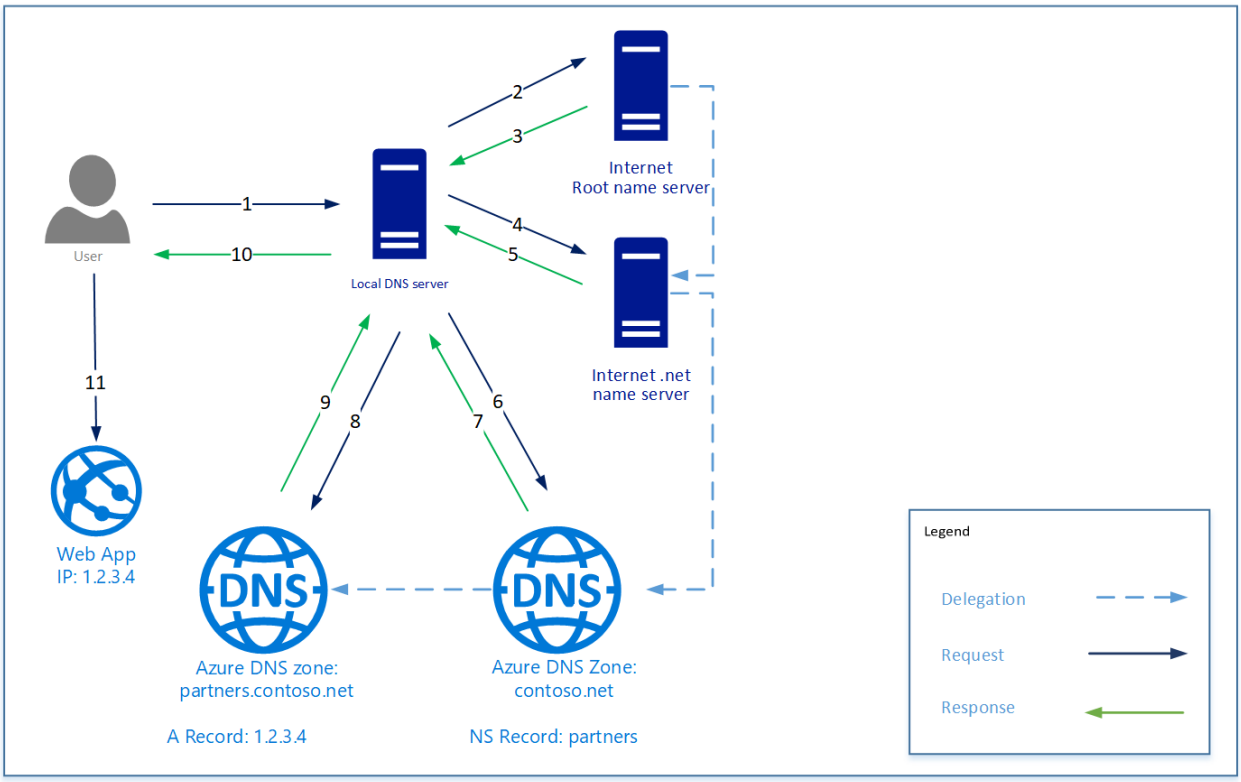
DNS clients in PCs or mobile devices typically call a recursive DNS server to perform any DNS queries the client applications need.

When a recursive DNS server receives a query for a DNS record such as 'www.contoso.com', it first needs to find the name server hosting the zone for the 'contoso.com' domain. To find the name server, it starts at the root name servers, and from there finds the name servers hosting the 'com' zone. It then queries the 'com' name servers to find the name servers hosting the 'contoso.com' zone. Finally, it is able to query these name servers for 'www.contoso.com'.

This procedure is called resolving the DNS name. Strictly speaking, DNS resolution includes additional steps such as following CNAMEs, but that's not important to understanding how DNS delegation works.

How does a parent zone 'point' to the name servers for a child zone? It does this using a special type of DNS record called an NS record (NS stands for 'name server'). For example, the root zone contains NS records for 'com' and shows the name servers for the 'com' zone. In turn, the 'com' zone contains NS records for 'contoso.com', which shows the name servers for the 'contoso.com' zone. Setting up the NS records for a child zone in a parent zone is called delegating the domain.

The following image shows an example DNS query. The contoso.net and partners.contoso.net are Azure DNS zones.



1. The client requests `www.partners.contoso.net` from their local DNS server.
2. The local DNS server does not have the record so it makes a request to their root name server.
3. The root name server does not have the record, but knows the address of the `.net` name server, it provides that address to the DNS server
4. The local DNS server sends the request to the `.net` name server.
5. The `.net` name server does not have the record but does know the address of the `contoso.net` name server. In this case, it responds with the address of the name server for the DNS zone hosted in Azure DNS.
6. The local DNS server sends the request to the name server for the `contoso.net` zone hosted in Azure DNS.
7. The zone `contoso.net` does not have the record but knows the name server for `partners.contoso.net` and responds with the address. In this case, it is a DNS zone hosted in Azure DNS.
8. The local DNS server sends the request to the name server for the `partners.contoso.net` zone.
9. The `partners.contoso.net` zone has the A record and responds with the IP address.
10. The local DNS server provides the IP address to the client
11. The client connects to the website `www.partners.contoso.net`.

Each delegation actually has two copies of the NS records; one in the parent zone pointing to the child, and another in the child zone itself. The 'contoso.net' zone contains the NS records for 'contoso.net' (in addition to the NS records in 'net'). These records are called authoritative NS records and they sit at the apex of the child zone.

Next steps

Learn how to delegate your domain to Azure DNS

Azure DNS FAQ

2/10/2020 • 12 minutes to read • Edit Online

About Azure DNS

What is Azure DNS?

The Domain Name System (DNS) translates, or resolves, a website or service name to its IP address. Azure DNS is a hosting service for DNS domains. It provides name resolution by using Microsoft Azure infrastructure. By hosting your domains in Azure, you can manage your DNS records by using the same credentials, APIs, tools, and billing as your other Azure services.

DNS domains in Azure DNS are hosted on the Azure global network of DNS name servers. This system uses Anycast networking so that each DNS query is answered by the closest available DNS server. Azure DNS provides fast performance and high availability for your domain.

Azure DNS is based on Azure Resource Manager. Azure DNS benefits from Resource Manager features such as role-based access control, audit logs, and resource locking. You can manage domains and records via the Azure portal, Azure PowerShell cmdlets, and the cross-platform Azure CLI. Applications that require automatic DNS management can integrate with the service via the REST API and SDKs.

How much does Azure DNS cost?

The Azure DNS billing model is based on the number of DNS zones hosted in Azure DNS. It's also based on the number of DNS queries they receive. Discounts are provided based on usage.

For more information, see the Azure DNS pricing page.

What is the SLA for Azure DNS?

Azure guarantees that valid DNS requests receive a response from at least one Azure DNS name server 100% of the time.

For more information, see the Azure DNS SLA page.

What is a DNS zone? Is it the same as a DNS domain?

A domain is a unique name in the domain name system. An example is contoso.com.

A DNS zone is used to host the DNS records for a particular domain. For example, the domain contoso.com might contain several DNS records. The records might include mail.contoso.com for a mail server and www.contoso.com for a website. These records are hosted in the DNS zone contoso.com.

A domain name is *just a name*. A DNS zone is a data resource that contains the DNS records for a domain name. You can use Azure DNS to host a DNS zone and manage the DNS records for a domain in Azure. It also provides DNS name servers to answer DNS queries from the Internet.

Do I need to buy a DNS domain name to use Azure DNS?

Not necessarily.

You don't need to buy a domain to host a DNS zone in Azure DNS. You can create a DNS zone at any time without owning the domain name. DNS queries for this zone resolve only if they're directed to the Azure DNS name servers assigned to the zone.

To link your DNS zone into the global DNS hierarchy, you must buy the domain name. Then, DNS queries from anywhere in the world find your DNS zone and answer with your DNS records.

Azure DNS features

Are there any restrictions when using alias records for a domain name apex with Traffic Manager?

Yes. You must use static public IP addresses with Azure Traffic Manager. Configure the **External endpoint** target by using a static IP address.

Does Azure DNS support DNS-based traffic routing or endpoint failover?

DNS-based traffic routing and endpoint failover are provided by Traffic Manager. Traffic Manager is a separate Azure service that can be used with Azure DNS. For more information, see the Traffic Manager overview.

Azure DNS only supports hosting static DNS domains, where each DNS query for a given DNS record always receives the same DNS response.

Does Azure DNS support domain name registration?

No. Azure DNS doesn't currently support the option to buy domain names. To buy domains, you must use a third-party domain name registrar. The registrar typically charges a small annual fee. The domains then can be hosted in Azure DNS for management of DNS records. For more information, see Delegate a domain to Azure DNS.

The feature to buy domain names is tracked in the Azure backlog. Use the feedback site to register your support for this feature.

Does Azure DNS support DNSSEC?

No. Azure DNS doesn't currently support the Domain Name System Security Extensions (DNSSEC).

The DNSSEC feature is tracked in the Azure DNS backlog. Use the feedback site to register your support for this feature.

Does Azure DNS support zone transfers (AXFR/IXFR)?

No. Azure DNS doesn't currently support zone transfers. DNS zones can be imported into Azure DNS by using the Azure CLI. DNS records are managed via the Azure DNS management portal, REST API, SDK, PowerShell cmdlets, or the CLI tool.

The zone transfer feature is tracked in the Azure DNS backlog. Use the feedback site to register your support for this feature.

Does Azure DNS support URL redirects?

No. URL redirect services aren't a DNS service. They work at the HTTP level rather than the DNS level. Some DNS providers bundle a URL redirect service as part of their overall offering. This service isn't currently supported by Azure DNS.

The URL redirect feature is tracked in the Azure DNS backlog. Use the feedback site to register your support for this feature.

Does Azure DNS support the extended ASCII encoding (8-bit) set for TXT record sets?

Yes. Azure DNS supports the extended ASCII encoding set for TXT record sets. But you must use the latest version of the Azure REST APIs, SDKs, PowerShell, and the CLI. Versions older than October 1, 2017, or SDK 2.1 don't support the extended ASCII set.

For example, you might provide a string as the value for a TXT record that has the extended ASCII character \128. An example is "abcd\128efgh." Azure DNS uses the byte value of this character, which is 128, in internal representation. At the time of DNS resolution, this byte value is returned in the response. Also note that "abc" and "\097\098\099" are interchangeable as far as resolution is concerned.

We follow RFC 1035 zone file master format escape rules for TXT records. For example, `\` now actually escapes everything per the RFC. If you specify `A\B` as the TXT record value, it's represented and resolved as just `AB`. If you really want the TXT record to have `A\B` at resolution, you need to escape the `\` again. As an example, specify

`A\\B`.

This support currently isn't available for TXT records created from the Azure portal.

Alias records

What are some scenarios where alias records are useful?

See the scenarios section in the Azure DNS alias records overview.

What record types are supported for alias record sets?

Alias record sets are supported for the following record types in an Azure DNS zone:

- A
- AAAA
- CNAME

What resources are supported as targets for alias record sets?

- **Point to a public IP resource from a DNS A/AAAA record set.** You can create an A/AAAA record set and make it an alias record set to point to a public IP resource.
- **Point to a Traffic Manager profile from a DNS A/AAAA/CNAME record set.** You can point to the CNAME of a Traffic Manager profile from a DNS CNAME record set. An example is contoso.trafficmanager.net. Now, you also can point to a Traffic Manager profile that has external endpoints from an A or AAAA record set in your DNS zone.
- **Point to an Azure Content Delivery Network (CDN) endpoint**. This is useful when you create static websites using Azure storage and Azure CDN.
- **Point to another DNS record set within the same zone.** Alias records can reference to other record sets of the same type. For example, you can have a DNS CNAME record set be an alias to another CNAME record set of the same type. This arrangement is useful if you want some record sets to be aliases and some non-aliases.

Can I create and update alias records from the Azure portal?

Yes. You can create or manage alias records in the Azure portal along with the Azure REST APIs, PowerShell, the CLI, and SDKs.

Will alias records help to make sure my DNS record set is deleted when the underlying public IP is deleted?

Yes. This feature is one of the core capabilities of alias records. It helps you avoid potential outages for users of your application.

Will alias records help to make sure my DNS record set is updated to the correct IP address when the underlying public IP address changes?

Yes. This feature is one of the core capabilities of alias records. It helps you avoid potential outages or security risks for your application.

Are there any restrictions when using alias record sets for A or AAAA records to point to Traffic Manager?

Yes. To point to a Traffic Manager profile as an alias from an A or AAAA record set, the Traffic Manager profile must use only external endpoints. When you create the external endpoints in Traffic Manager, provide the actual IP addresses of the endpoints.

Is there an additional charge to use alias records?

Alias records are a qualification on a valid DNS record set. There's no additional billing for alias records.

Use Azure DNS

Can I co-host a domain by using Azure DNS and another DNS provider?

Yes. Azure DNS supports co-hosting domains with other DNS services.

To set up co-hosting, modify the NS records for the domain to point to the name servers of both providers. The name server (NS) records control which providers receive DNS queries for the domain. You can modify these NS records in Azure DNS, in the other provider, and in the parent zone. The parent zone is typically configured via the domain name registrar. For more information on DNS delegation, see DNS domain delegation.

Also, make sure that the DNS records for the domain are in sync between both DNS providers. Azure DNS doesn't currently support DNS zone transfers. DNS records must be synchronized by using either the Azure DNS management portal, REST API, SDK, PowerShell cmdlets, or the CLI tool.

Do I have to delegate my domain to all four Azure DNS name servers?

Yes. Azure DNS assigns four name servers to each DNS zone. This arrangement is for fault isolation and increased resilience. To qualify for the Azure DNS SLA, delegate your domain to all four name servers.

What are the usage limits for Azure DNS?

The following default limits apply when you use Azure DNS.

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
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.

Can I move an Azure DNS zone between resource groups or between subscriptions?

Yes. DNS zones can be moved between resource groups or between subscriptions.

There's no effect on DNS queries when you move a DNS zone. The name servers assigned to the zone stay the same. DNS queries are processed as normal throughout.

For more information and instructions on how to move DNS zones, see Move resources to a new resource group or subscription.

How long does it take for DNS changes to take effect?

New DNS zones and DNS records typically appear in the Azure DNS name servers quickly. The timing is a few seconds.

Changes to existing DNS records can take a little longer. They typically appear in the Azure DNS name servers within 60 seconds. DNS caching by DNS clients and DNS recursive resolvers outside of Azure DNS also can affect timing. To control this cache duration, use the Time-To-Live (TTL) property of each record set.

How can I protect my DNS zones against accidental deletion?

Azure DNS is managed by using Azure Resource Manager. Azure DNS benefits from the access control features that Azure Resource Manager provides. Role-based access control controls which users have read or write access to DNS zones and record sets. Resource locks prevent accidental modification or deletion of DNS zones and record sets.

For more information, see Protect DNS zones and records.

How do I set up SPF records in Azure DNS?

Sender policy framework (SPF) records are used to specify which email servers can send email on behalf of a domain name. Correct configuration of SPF records is important to prevent recipients from marking your email as junk.

The DNS RFCs originally introduced a new SPF record type to support this scenario. To support older name servers, they also allowed the use of the TXT record type to specify SPF records. This ambiguity led to confusion, which was resolved by RFC 7208. It states that SPF records must be created by using the TXT record type. It also states that the SPF record type is deprecated.

SPF records are supported by Azure DNS and must be created by using the TXT record type. The obsolete SPF record type isn't supported. When you import a DNS zone file, any SPF records that use the SPF record type are converted to the TXT record type.

Do Azure DNS name servers resolve over IPv6?

Yes. Azure DNS name servers are dual stack. Dual stack means they have IPv4 and IPv6 addresses. To find the IPv6 address for the Azure DNS name servers assigned to your DNS zone, use a tool such as nslookup. An example is `nslookup -q=aaaa <Azure DNS Nameserver>`.

How do I set up an IDN in Azure DNS?

Internationalized domain names (IDNs) encode each DNS name by using punycode. DNS queries are made by using these punycode-encoded names.

To configure IDNs in Azure DNS, convert the zone name or record set name to punycode. Azure DNS doesn't currently support built-in conversion to or from punycode.

Next steps

- Learn more about Azure DNS.
- Learn more about how to use Azure DNS for private domains.
- Learn more about DNS zones and records.

- Get started with Azure DNS.

Azure DNS metrics and alerts

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

Azure DNS is a hosting service for DNS domains that provides name resolution using the Microsoft Azure infrastructure. This article describes metrics and alerts for the Azure DNS service.

Azure DNS metrics

Azure DNS provides metrics for customers to enable them to monitor specific aspects of their DNS zones hosted in the service. In addition, with Azure DNS metrics, you can configure and receive alerts based on conditions of interest. The metrics are provided via the Azure Monitor service. Azure DNS provides the following metrics via Azure Monitor for your DNS zones:

- QueryVolume
- RecordSetCount
- RecordSetCapacityUtilization

You can also see the definition of these metrics on the Azure Monitor documentation page.

> **NOTE**
> At this time, these metrics are only available for Public DNS zones hosted in Azure DNS. If you have Private Zones hosted in Azure DNS, these metrics will not provide data for those zones. In addition, the metrics and alerting feature is only supported in Azure Public cloud. Support for sovereign clouds will follow at a later time.

The most granular element that you can see metrics for is a DNS zone. You cannot currently see metrics for individual resource records within a zone.

Query volume

The *Query Volume* metric in Azure DNS shows the volume of DNS queries (query traffic) that is received by Azure DNS for your DNS zone. The unit of measurement is Count and the aggregation is the total of all the queries received over a period of time.

To view this metric, select Metrics (preview) explorer experience from the Monitor tab in the Azure portal. Select your DNS zone from the Resource drop-down, select the Query Volume metric, and select Sum as the Aggregation. Below screenshot shows an example. For more information on the Metrics Explorer experience and charting, see Azure Monitor Metrics Explorer.

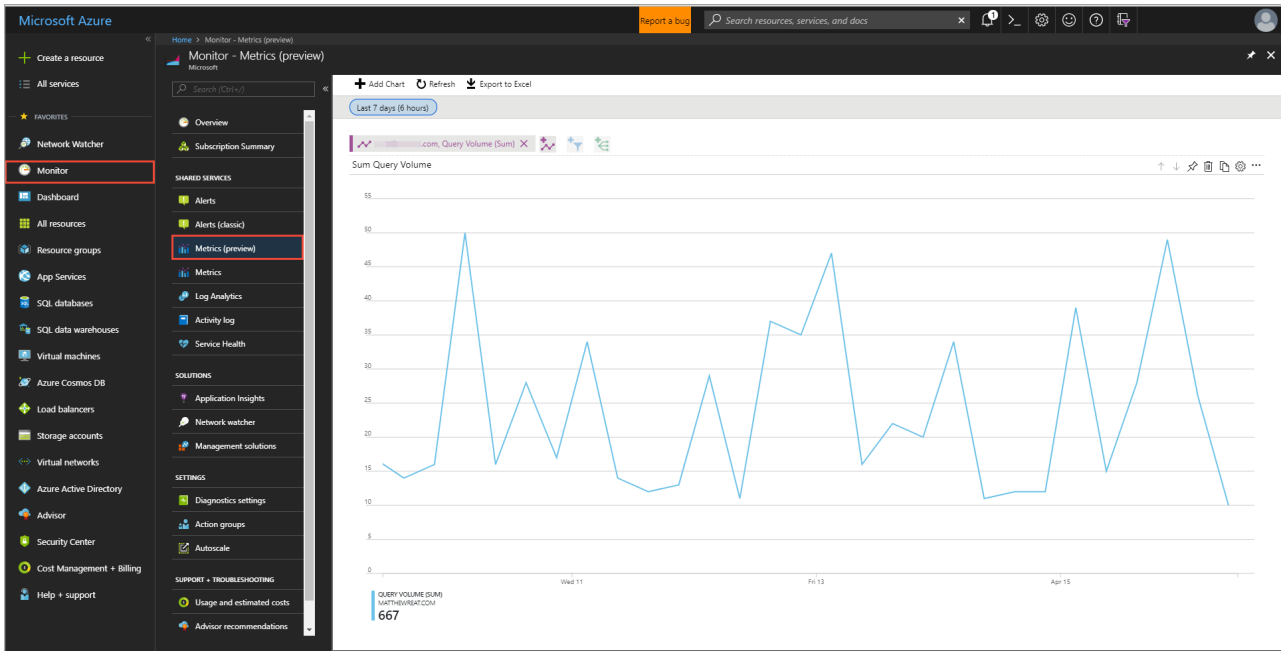


Figure: Azure DNS Query Volume metrics

Record Set Count

The *Record Set Count* metric shows the number of Recordsets in Azure DNS for your DNS zone. All the Recordsets defined in your zone are counted. The unit of measurement is Count and the aggregation is the Maximum of all the Recordsets. To view this metric, select **Metrics (preview)** explorer experience from the **Monitor** tab in the Azure portal. Select your DNS zone from the **Resource** drop-down, select the **Record Set Count** metric, and then select **Max** as the **Aggregation**. For more information on the Metrics Explorer experience and charting, see Azure Monitor Metrics Explorer.

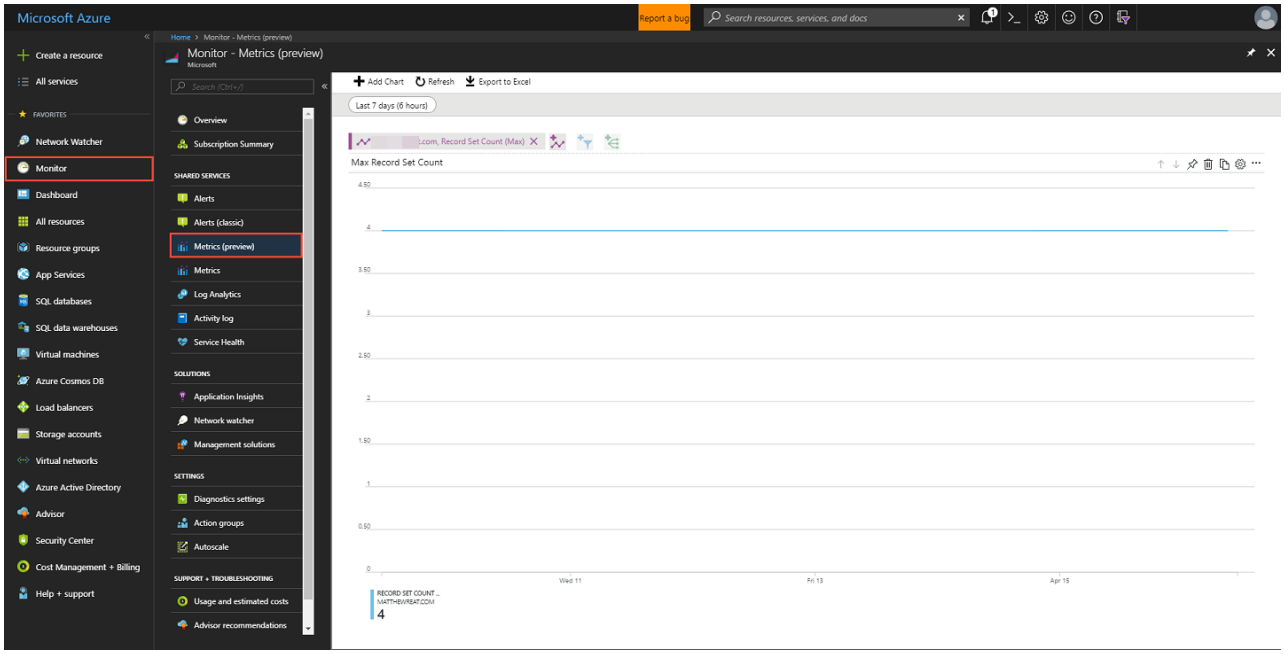


Figure: Azure DNS Record Set Count metrics

Record Set Capacity Utilization

The *Record Set Capacity Utilization* metric in Azure DNS shows the percentage of utilization of your Recordset capacity for a DNS Zone. Every DNS zone in Azure DNS is subject to a Recordset limit that defines the maximum number of Recordsets that are allowed for the Zone (see DNS limits). Hence, this metric shows you how close you are to hitting the Recordset limit. For example, if you have 500 Recordsets configured for your DNS zone, and the zone has the default Recordset limit of 5000, the RecordSetCapacityUtilization metric will show the value of 10% (which is obtained by dividing 500 by 5000). The unit of measurement is **Percentage** and the **Aggregation** type is

Maximum. To view this metric, select Metrics (preview) explorer experience from the Monitor tab in the Azure portal. Select your DNS zone from the Resource drop-down, select the Record Set Capacity Utilization metric, and select Max as the Aggregation. Below screenshot shows an example. For more information on the Metrics Explorer experience and charting, see Azure Monitor Metrics Explorer.

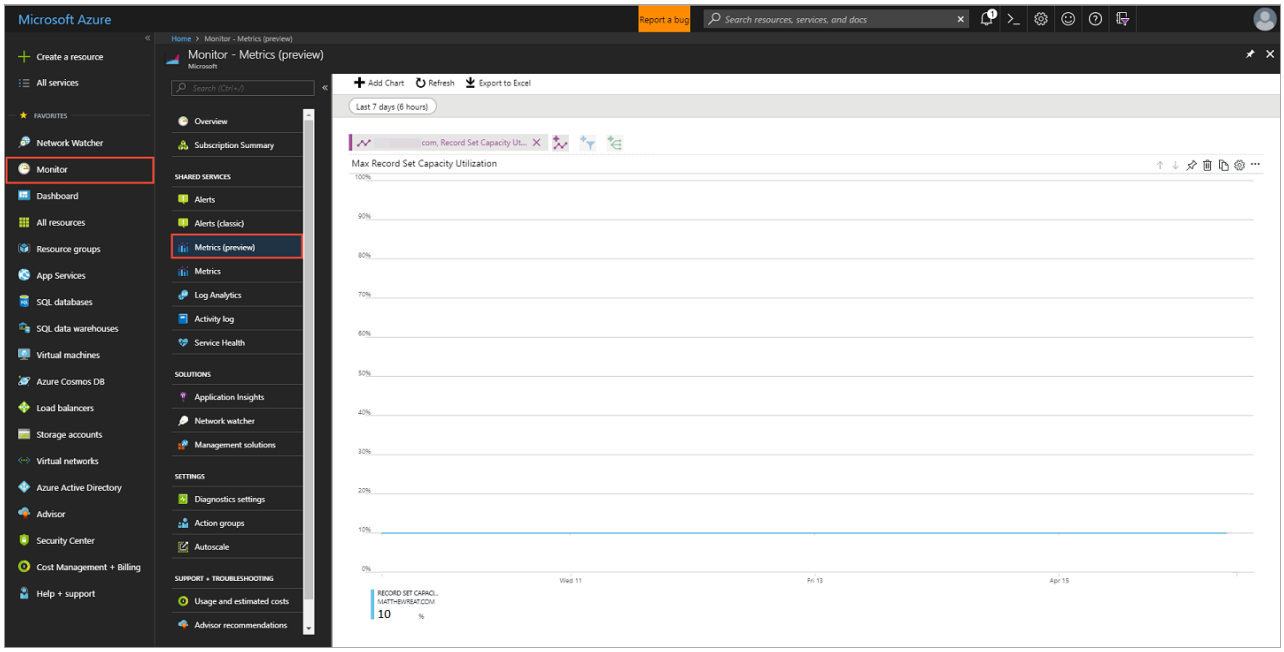


Figure: Azure DNS Record Set Capacity Utilization metrics

Alerts in Azure DNS

Azure Monitor provides the capability to alert against available metric values. The DNS metrics are available in the new Alert configuration experience. As described in detail in the Azure Monitor alerts documentation, you can select DNS Zone as the resource, choose the Metric signal type, and configure the alert logic and other parameters such as **Period** and **Frequency**. You can further define an Action Group for when the alert condition is met, whereby the alert will be delivered via the chosen actions. For more information on how to configure alerting for Azure Monitor metrics, see Create, view, and manage alerts using Azure Monitor.

Next steps

- Learn more about Azure DNS.

Overview of reverse DNS and support in Azure

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

This article gives an overview of how reverse DNS works, and the reverse DNS scenarios supported in Azure.

What is reverse DNS?

Conventional DNS records enable a mapping from a DNS name (such as 'www.contoso.com') to an IP address (such as 64.4.6.100). Reverse DNS enables the translation of an IP address (64.4.6.100) back to a name ('www.contoso.com').

Reverse DNS records are used in a variety of situations. For example, reverse DNS records are widely used in combating e-mail spam by verifying the sender of an e-mail message. The receiving mail server retrieves the reverse DNS record of the sending server's IP address, and verifies if that host is authorized to send e-mail from the originating domain.

How reverse DNS works

Reverse DNS records are hosted in special DNS zones, known as 'ARPA' zones. These zones form a separate DNS hierarchy in parallel with the normal hierarchy hosting domains such as 'contoso.com'.

For example, the DNS record 'www.contoso.com' is implemented using a DNS 'A' record with the name 'www' in the zone 'contoso.com'. This A record points to the corresponding IP address, in this case 64.4.6.100. The reverse lookup is implemented separately, using a 'PTR' record named '100' in the zone '6.4.64.in-addr.arpa' (note that IP addresses are reversed in ARPA zones.) This PTR record, if it has been configured correctly, points to the name 'www.contoso.com'.

When an organization is assigned an IP address block, they also acquire the right to manage the corresponding ARPA zone. The ARPA zones corresponding to the IP address blocks used by Azure are hosted and managed by Microsoft. Your ISP may host the ARPA zone for your own IP addresses for you, or may allow you to host the ARPA zone in a DNS service of your choice, such as Azure DNS.

> **NOTE**
> Forward DNS lookups and reverse DNS lookups are implemented in separate, parallel DNS hierarchies. The reverse lookup for 'www.contoso.com' is **not** hosted in the zone 'contoso.com', rather it is hosted in the ARPA zone for the corresponding IP address block. Separate zones are used for IPv4 and IPv6 address blocks.

IPv4

The name of an IPv4 reverse lookup zone should be in the following format: `<IPv4 network prefix in reverse order>.in-addr.arpa`.

For example, when creating a reverse zone to host records for hosts with IPs that are in the 192.0.2.0/24 prefix, the zone name would be created by isolating the network prefix of the address (192.0.2) and then reversing the order (2.0.192) and adding the suffix `.in-addr.arpa`.

SUBNET CLASS	NETWORK PREFIX	REVERSED NETWORK PREFIX	STANDARD SUFFIX	REVERSE ZONE NAME
Class A	203.0.0.0/8	203	.in-addr.arpa	`203.in-addr.arpa`
Class B	198.51.0.0/16	51.198	.in-addr.arpa	`51.198.in-addr.arpa`
Class C	192.0.2.0/24	2.0.192	.in-addr.arpa	`2.0.192.in-addr.arpa`

Classless IPv4 delegation

In some cases, the IP range allocated to an organization is smaller than a Class C (/24) range. In this case, the IP range does

not fall on a zone boundary within the `.in-addr.arpa` zone hierarchy, and hence cannot be delegated as a child zone.

Instead, a different mechanism is used to transfer control of individual reverse lookup (PTR) records to a dedicated DNS zone. This mechanism delegates a child zone for each IP range, then maps each IP address in the range individually to that child zone using CNAME records.

For example, suppose an organization is granted the IP range 192.0.2.128/26 by its ISP. This represents 64 IP addresses, from 192.0.2.128 to 192.0.2.191. Reverse DNS for this range is implemented as follows:

- The organization creates a reverse lookup zone called 128-26.2.0.192.in-addr.arpa. The prefix '128-26' represents the network segment assigned to the organization within the Class C (/24) range.
- The ISP creates NS records to set up the DNS delegation for the above zone from the Class C parent zone. It also creates CNAME records in the parent (Class C) reverse lookup zone, mapping each IP address in the IP range to the new zone created by the organization:

```
$ORIGIN 2.0.192.in-addr.arpa
; Delegate child zone
128-26    NS       <name server 1 for 128-26.2.0.192.in-addr.arpa>
128-26    NS       <name server 2 for 128-26.2.0.192.in-addr.arpa>
; CNAME records for each IP address
129       CNAME    129.128-26.2.0.192.in-addr.arpa
130       CNAME    130.128-26.2.0.192.in-addr.arpa
131       CNAME    131.128-26.2.0.192.in-addr.arpa
; etc
```

- The organization then manages the individual PTR records within their child zone.

```
$ORIGIN 128-26.2.0.192.in-addr.arpa
; PTR records for each UIP address. Names match CNAME targets in parent zone
129      PTR    www.contoso.com
130      PTR    mail.contoso.com
131      PTR    partners.contoso.com
; etc
```

A reverse lookup for the IP address '192.0.2.129' queries for a PTR record named '129.2.0.192.in-addr.arpa'. This query resolves via the CNAME in the parent zone to the PTR record in the child zone.

IPv6

The name of an IPv6 reverse lookup zone should be in the following form:
`<IPv6 network prefix in reverse order>.ip6.arpa`

For example,. When creating a reverse zone to host records for hosts with IPs that are in the 2001:db8:1000:abdc::/64 prefix, the zone name would be created by isolating the network prefix of the address (2001:db8:abdc::). Next expand the IPv6 network prefix to remove zero compression, if it was used to shorten the IPv6 address prefix (2001:0db8:abdc:0000::). Reverse the order, using a period as the delimiter between each hexadecimal number in the prefix, to build the reversed network prefix (`0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2`) and add the suffix `.ip6.arpa`.

NETWORK PREFIX	EXPANDED AND REVERSED NETWORK PREFIX	STANDARD SUFFIX	REVERSE ZONE NAME
2001:db8:abdc::/64	0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2	.ip6.arpa	`0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip`
2001:db8:1000:9102::/64	2.0.1.9.0.0.0.1.8.b.d.0.1.0.0.2	.ip6.arpa	`2.0.1.9.0.0.0.1.8.b.d.0.1.0.0.2.ip`

Azure support for reverse DNS

Azure supports two separate scenarios relating to reverse DNS:

Hosting the reverse lookup zone corresponding to your IP address block. Azure DNS can be used to host your reverse lookup zones and manage the PTR records for each reverse DNS lookup, for both IPv4 and IPv6. The process of

creating the reverse lookup (ARPA) zone, setting up the delegation, and configuring PTR records is the same as for regular DNS zones. The only differences are that the delegation must be configured via your ISP rather than your DNS registrar, and only the PTR record type should be used.

Configure the reverse DNS record for the IP address assigned to your Azure service. Azure enables you to configure the reverse lookup for the IP addresses allocated to your Azure service. This reverse lookup is configured by Azure as a PTR record in the corresponding ARPA zone. These ARPA zones, corresponding to all the IP ranges used by Azure, are hosted by Microsoft

Next steps

For more information on reverse DNS, see reverse DNS lookup on Wikipedia.
Learn how to host the reverse lookup zone for your ISP-assigned IP range in Azure DNS.
Learn how to manage reverse DNS records for your Azure services.

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.

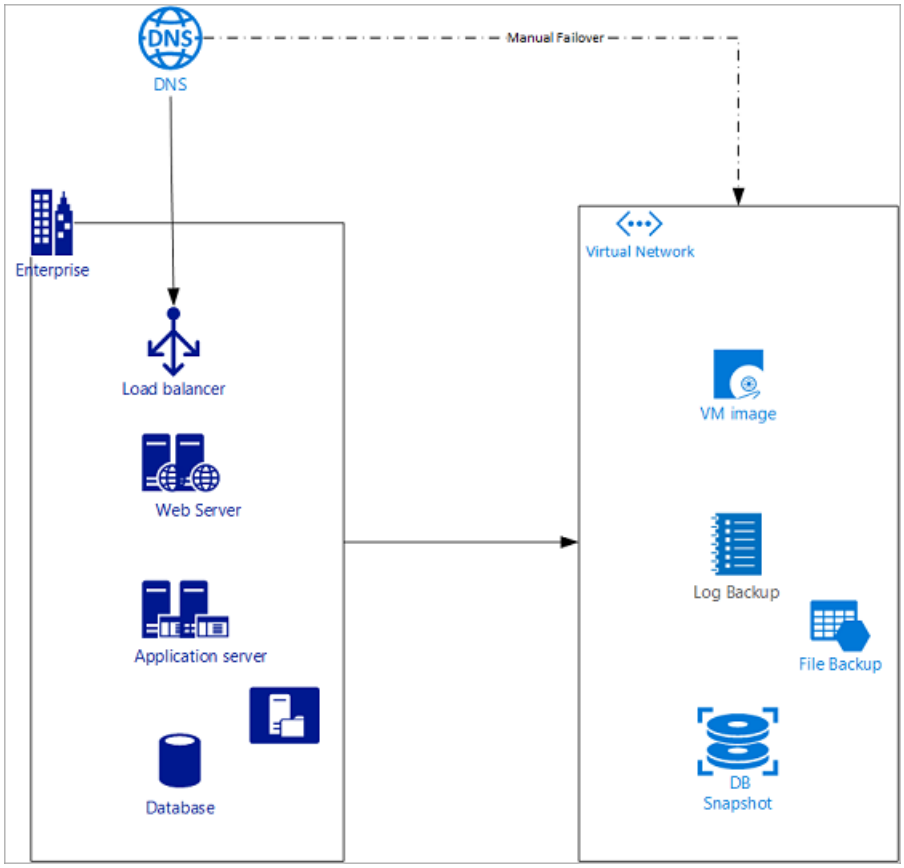


Figure - Active/Passive with cold standby disaster recovery configuration

- **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.

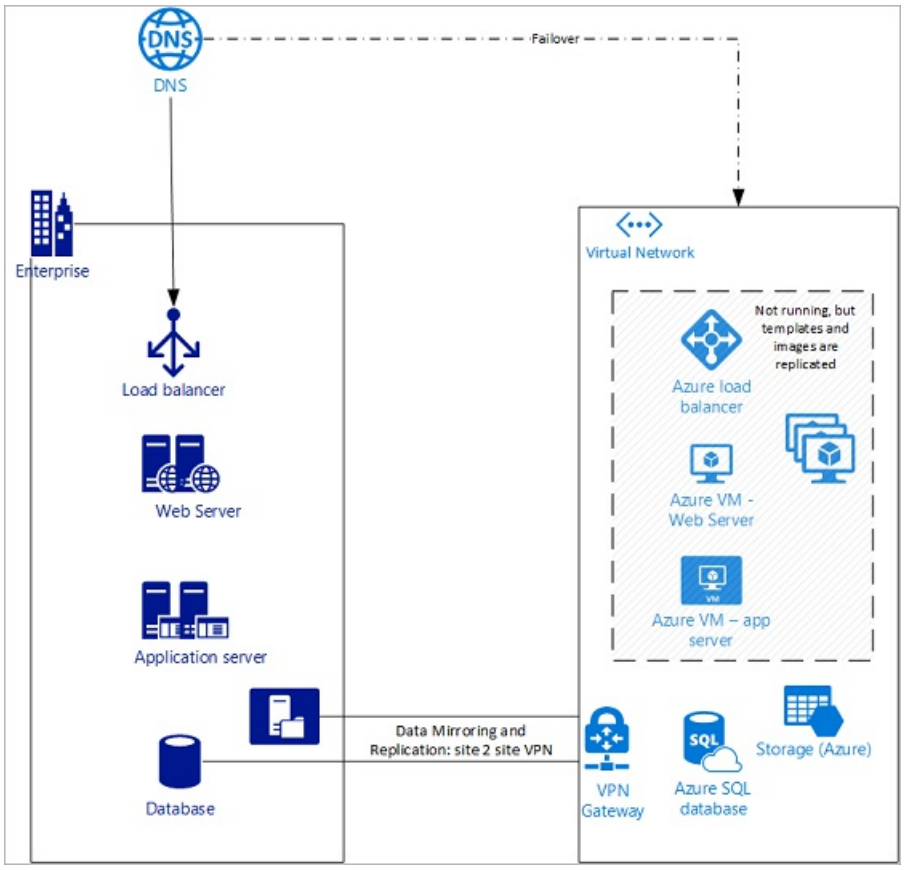


Figure: Active/Passive with pilot light disaster recovery configuration

- **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.

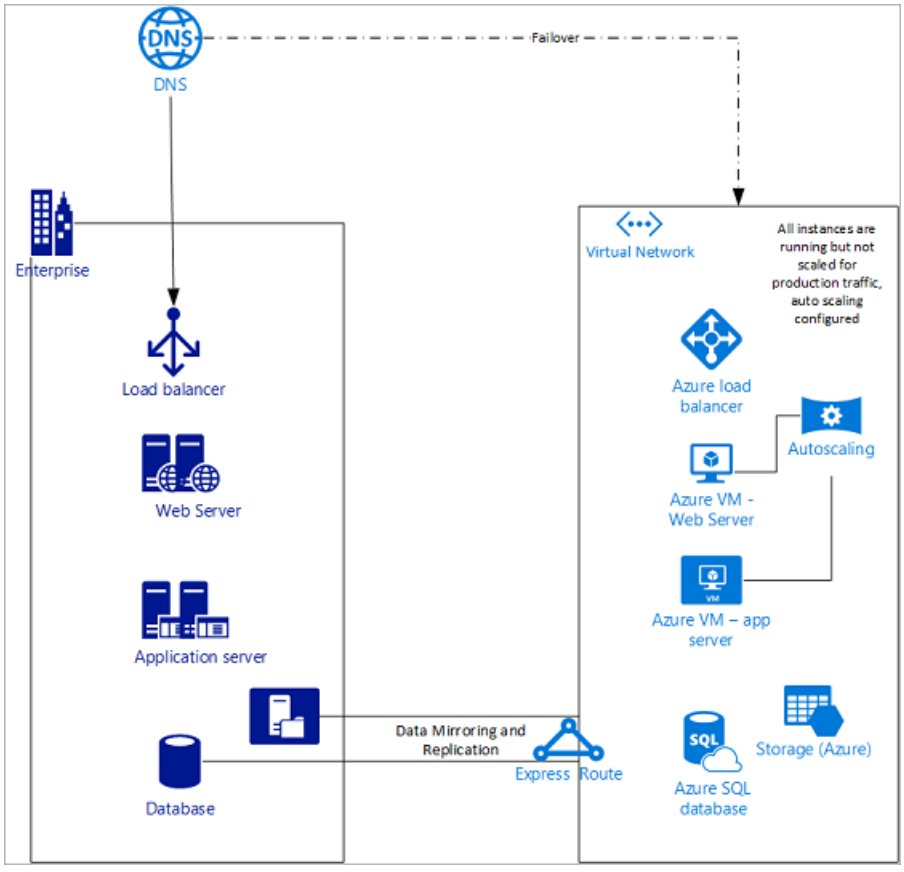


Figure: Active/Passive with warm standby disaster recovery configuration

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.

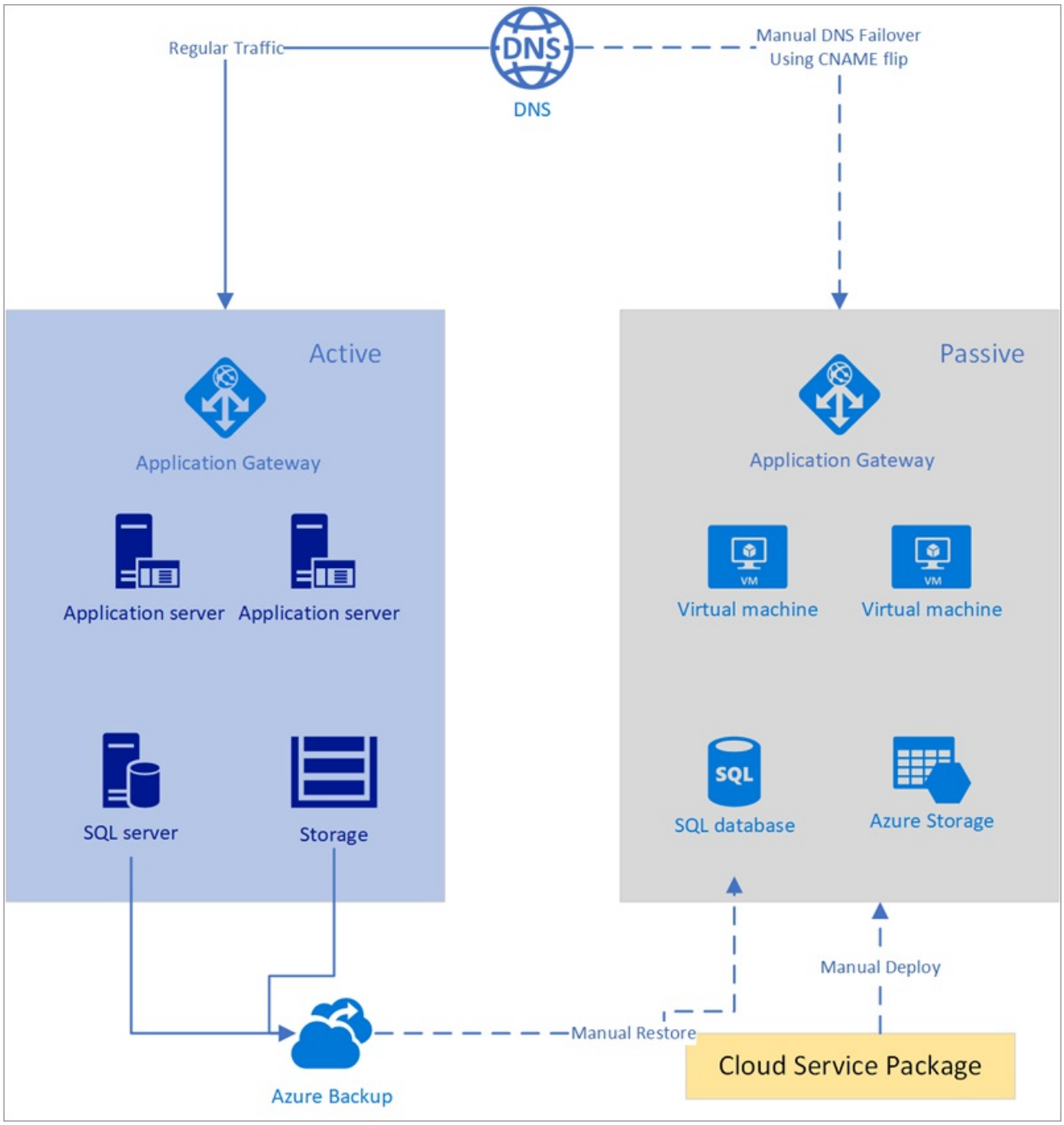


Figure - Manual failover using Azure DNS

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:

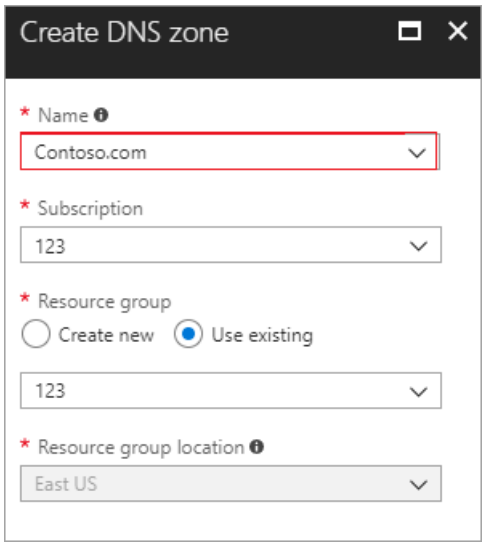


Figure - Create a DNS zone in Azure

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.

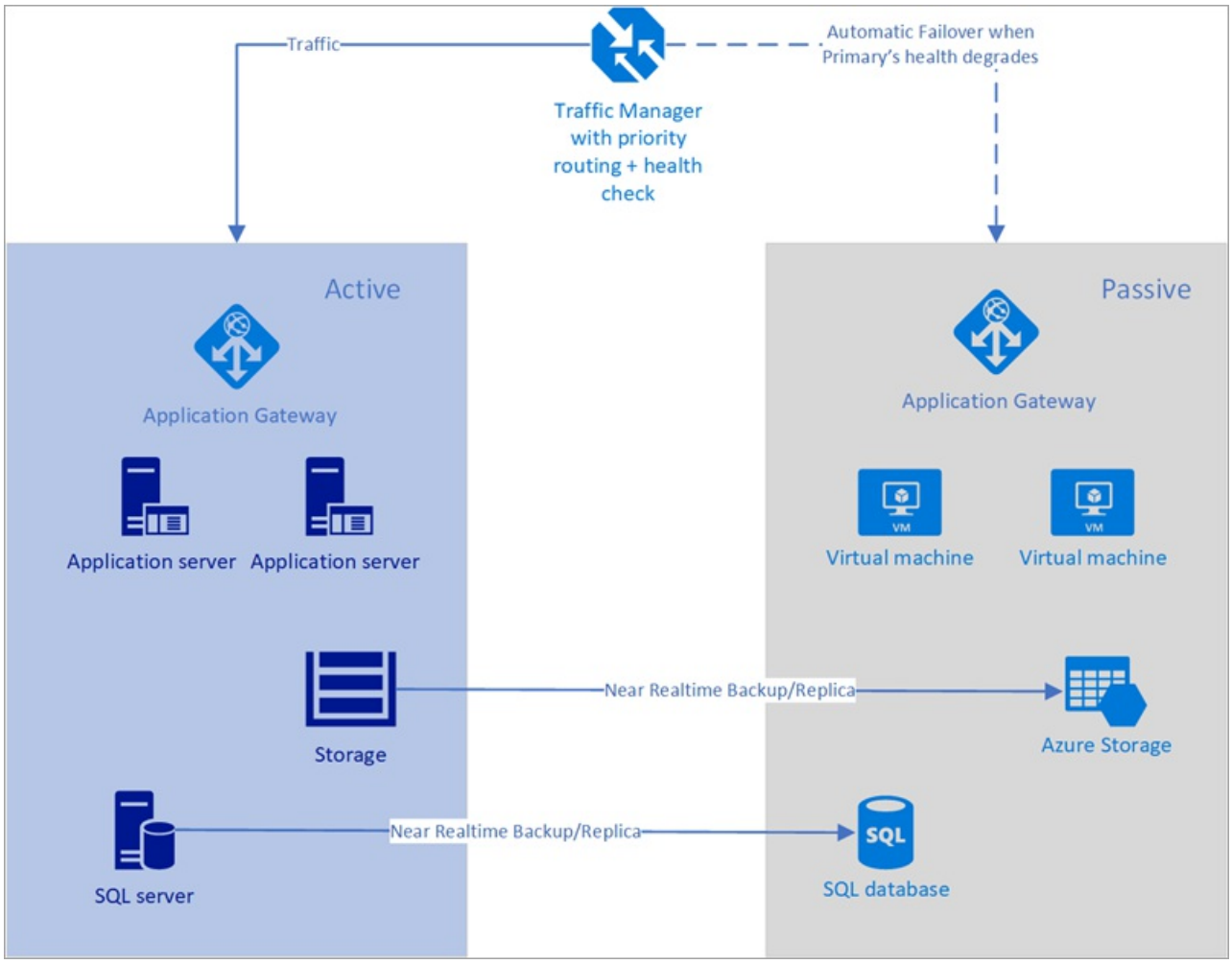


Figure - Automatic failover using Azure Traffic Manager

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.

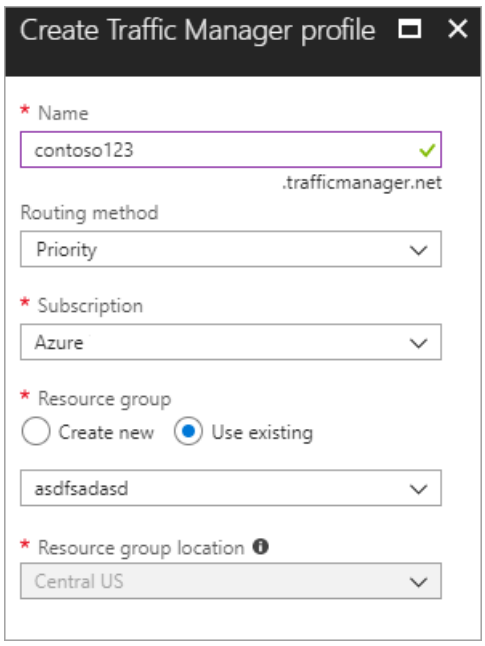


Figure - Create a Traffic Manager profile

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.

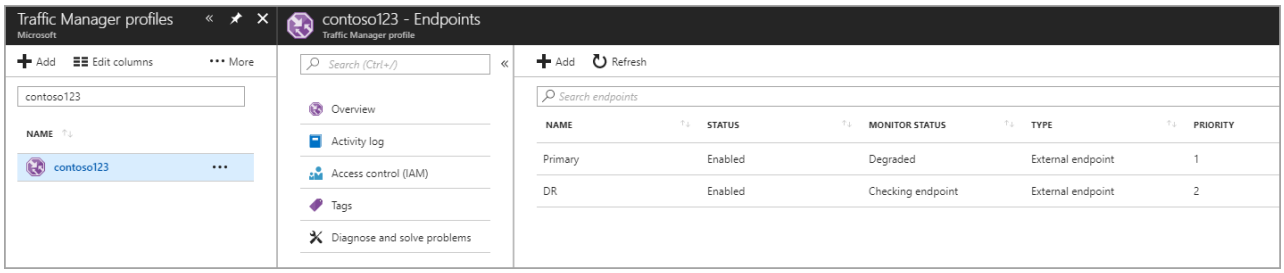


Figure - Create disaster recovery endpoints

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.

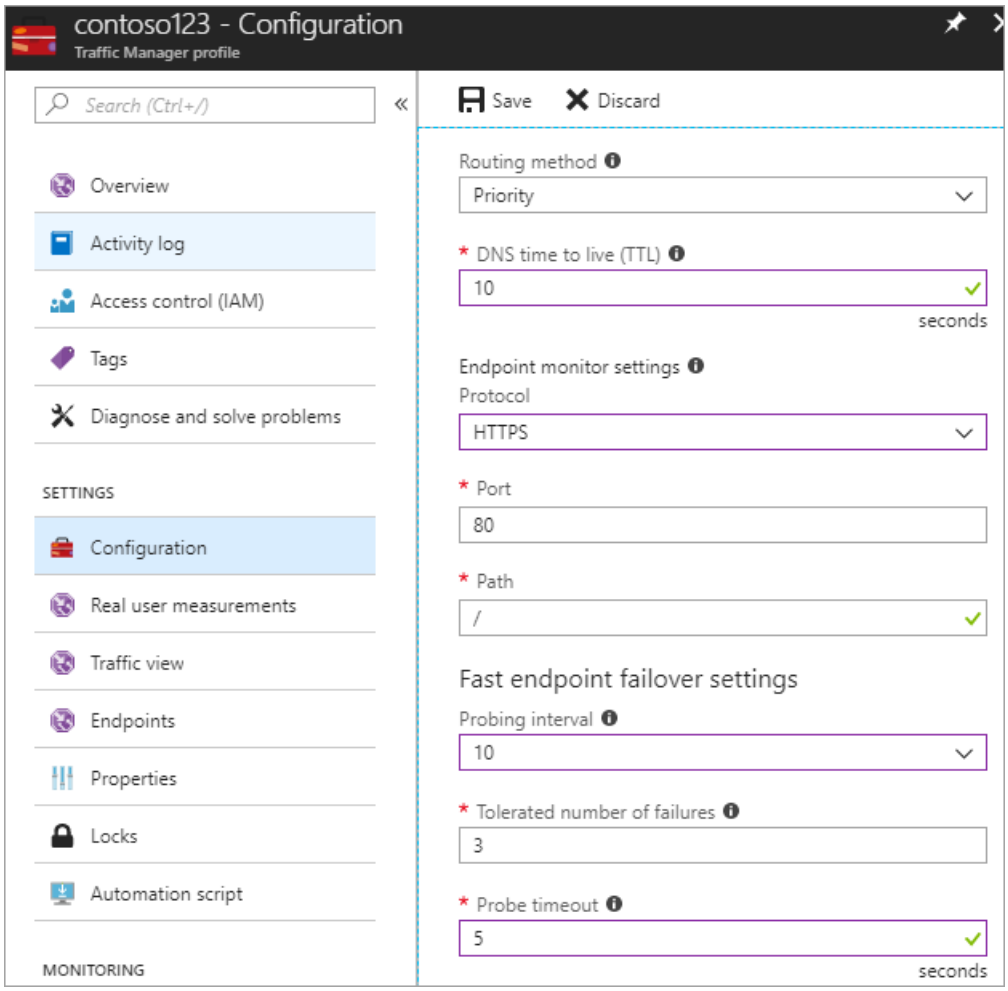


Figure - Set up health check and failover configuration

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.

What is a private Azure DNS zone

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

Azure Private DNS provides a reliable, secure DNS service to manage and resolve domain names in a virtual network without the need to add a custom DNS solution. By using private DNS zones, you can use your own custom domain names rather than the Azure-provided names available today.

The records contained in a private DNS zone are not resolvable from the Internet. DNS resolution against a private DNS zone works only from virtual networks that are linked to it.

You can link a private DNS zone to one or more virtual networks by creating virtual network links. You can also enable auto-registration feature to automatically manage the life cycle of the DNS records for the virtual machines deployed in a virtual network.

Limits

To understand how many private DNS zones you can create in a subscription and how many record sets are supported in a private DNS zone see Azure DNS limits

Restrictions

- Single labeled private DNS zones are not supported. Your private DNS zone must have two or more labels. For example contoso.com has two labels separated by a dot. A private DNS zone can have a maximum 34 labels.
- You can't create zone delegations (NS records) in a private DNS zone. If you intend to use a child domain, you can directly create the domain as a private DNS zone and link it to virtual network without setting up a nameserver delegation from the parent zone.

Next steps

- Learn how to create a private zone in Azure DNS by using Azure PowerShell or Azure CLI.
- Read about some common private zone scenarios that can be realized with private zones in Azure DNS.
- For common questions and answers about private zones in Azure DNS, including specific behavior you can expect for certain kinds of operations, see Private DNS FAQ.

What is a virtual network link?

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

Once you create a private DNS zone in Azure, it is not immediately accessible from any virtual network. You must link it to a virtual network before a VM hosted in that network can access the private DNS zone. To link a private DNS zone with a virtual network, you must create a virtual network link under the private DNS zone. Every private DNS zone has a collection of virtual network link child resources. Each one of these resources represents a connection to a virtual network.

You can link a virtual network to a private DNS zone as a registration virtual network or as a resolution virtual network.

Registration virtual network

When you create a link between a private DNS zone and a virtual network, you have an option to turn on autoregistration of DNS records for virtual machines. If you choose this option, the virtual network becomes a registration virtual network for the private DNS zone. A DNS record is automatically created for the virtual machines that you deploy in the network. DNS records are created for the virtual machines that you have already deployed in the virtual network. From the virtual network perspective, private DNS zone becomes the registration zone for that virtual network. One private DNS zone can have multiple registration virtual networks, however every virtual network can have exactly one registration zone associated with it.

Resolution virtual network

When you create a virtual network link under a private DNS zone and choose not to enable DNS record autoregistration, the virtual network is treated as a resolution only virtual network. DNS records for virtual machines deployed in such networks will not be automatically created in the linked private DNS zone. However, the virtual machines deployed in such a network can successfully query the DNS records from the private DNS zone. These records may be manually created by you or may be populated from other virtual networks that have been linked as registration networks with the private DNS zone. One private DNS zone can have multiple resolution virtual networks and a virtual network can have multiple resolution zones associated to it.

Limits

To understand how many registration and resolution networks, you can link to private DNS zones see Azure DNS Limits

Other considerations

- Virtual networks deployed using classic deployment model are not supported.
- You can create only one link between a private DNS zone and a virtual network.
- Each virtual network link under a private DNS zone must have unique name within the context of the private DNS zone. You can have links with same name in different private DNS zones.
- After creating a virtual network link, check the "Link Status" field of the virtual network link resource. Depending on the size of the virtual network, it can take a few minutes before the link is operation and the Link Status changes to *Completed*.
- When you delete a virtual network, all the virtual network links and auto-registered DNS records

associated with it in different private DNS zones are automatically deleted.

Next steps

- Learn how to link a virtual network to a private DNS zone using Azure portal
- Learn how to create a private zone in Azure DNS by using Azure PowerShell or Azure CLI.
- Read about some common private zone scenarios that can be realized with private zones in Azure DNS.
- For common questions and answers about private zones in Azure DNS, including specific behavior you can expect for certain kinds of operations, see Private DNS FAQ.

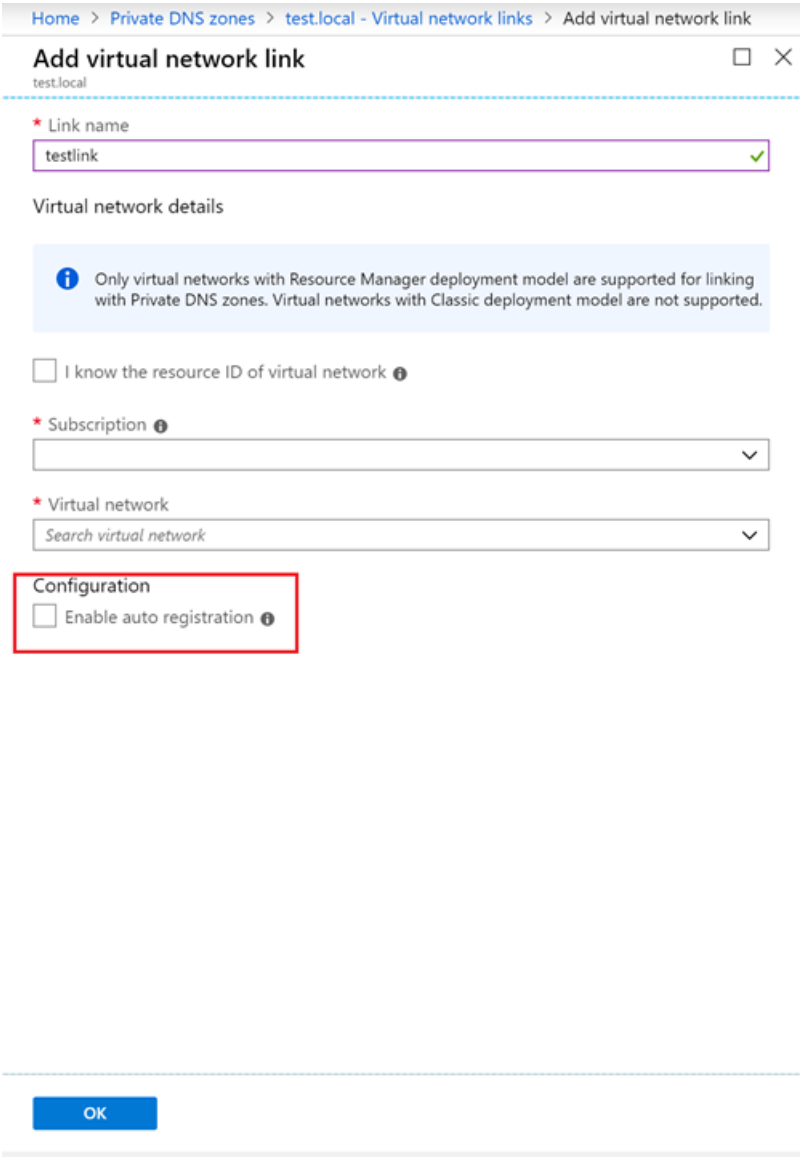
What is the autoregistration feature of Azure DNS private zones

10/4/2019 • 2 minutes to read • Edit Online

The Azure DNS private zones auto registration feature takes the pain out of DNS record management for virtual machines deployed in a virtual network. When you link an virtual network with a private DNS zone and enable auto registration for all the virtual machines, the DNS records for the virtual machines deployed in the virtual network are automatically created in the private DNS zone. In addition to forward look records (A records), reverse lookup records (PTR records) are also automatically created for the virtual machines. If you add more virtual machines to the virtual network, DNS records for these virtual machines are also automatically created in the linked private DNS zone.

When you delete a virtual machine, the DNS records for the virtual machine are automatically deleted from the private DNS zone.

You can enable autoregistration by selecting "Enable auto registration" option while creating a virtual network link.



Restrictions

- Autoregistration works only for virtual machines. For all other resources like internal load balancers etc., you can create DNS records manually in the private DNS zone linked to the virtual network.
- DNS records are created automatically only for the primary virtual machine NIC . If your virtual machines have more than one NIC, you can manually create the DNS records for other network interfaces.
- autoregistration for IPv6 (AAAA records) is not supported.

Next steps

- Learn how to create a private zone in Azure DNS using Azure PowerShell or Azure CLI.
- Read about some common private zone scenarios that can be realized with private zones in Azure DNS.
- For common questions and answers about private zones in Azure DNS, including specific behavior you can expect for certain kinds of operations, see Private DNS FAQ.

Azure DNS Private zones scenarios

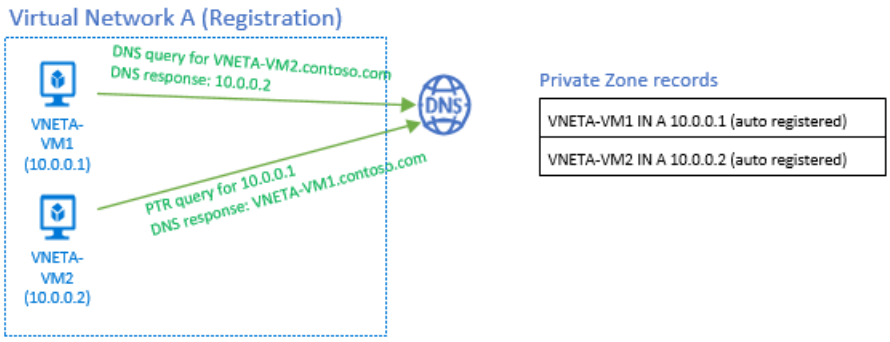
2/1/2020 • 3 minutes to read • Edit Online

Azure DNS Private Zones provide name resolution within a virtual network as well as between virtual networks. In this article, we look at some common scenarios that can be realized using this feature.

Scenario: Name Resolution scoped to a single virtual network

In this scenario, you have a virtual network in Azure that has a number of Azure resources in it, including virtual machines (VMs). You want to resolve the resources from within the virtual network via a specific domain name (DNS zone), and you need the name resolution to be private and not accessible from the internet. Furthermore, for the VMs within the VNET, you need Azure to automatically register them into the DNS zone.

This scenario is depicted below. Virtual Network named "A" contains two VMs (VNETA-VM1 and VNETA-VM2). Each of these have Private IPs associated. Once you create a Private Zone named contoso.com and link this virtual network as a Registration virtual network, Azure DNS will automatically create two A records in the zone as depicted. Now, DNS queries from VNETA-VM1 to resolve VNETA-VM2.contoso.com will receive a DNS response that contains the Private IP of VNETA-VM2. Furthermore, a Reverse DNS query (PTR) for the Private IP of VNETA-VM1 (10.0.0.1) issued from VNETA-VM2 will receive a DNS response that contains the name of VNETA-VM1, as expected.



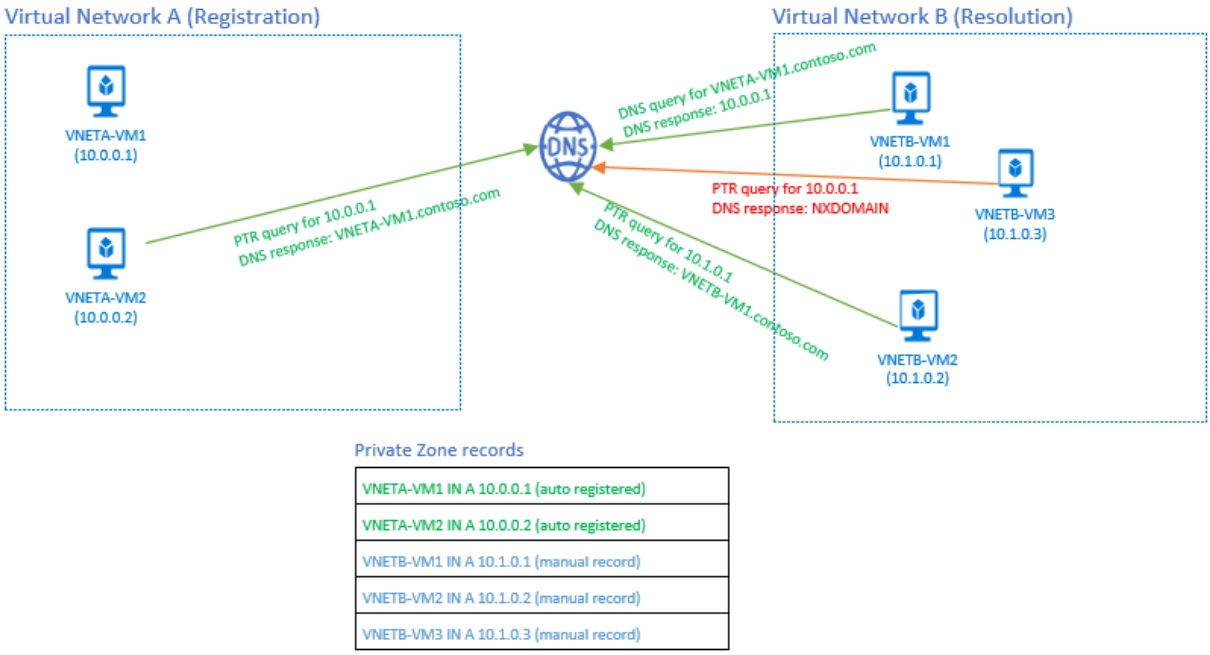
Scenario: Name Resolution across virtual networks

This scenario is the more common case where you need to associate a Private Zone with multiple virtual networks. This scenario can fit architectures such as the Hub-and-Spoke model where there is a central Hub virtual network to which multiple other Spoke virtual networks are connected. The central Hub virtual network can be linked as the Registration virtual network to a private zone, and the Spoke virtual networks can be linked as Resolution virtual networks.

The following diagram shows a simple version of this scenario where there are only two virtual networks - A and B. A is designated as a Registration virtual network and B is designated as a Resolution virtual network. The intent is for both virtual networks to share a common zone contoso.com. When the zone is created and the Resolution and Registration virtual networks are linked to the zone, Azure will automatically register DNS records for the VMs (VNETA-VM1 and VNETA-VM2) from the virtual network A. You can also manually add DNS records into the zone for VMs in the Resolution virtual network B. With this setup, you will observe the following behavior for forward and reverse DNS queries:

- A DNS query from VNETB-VM1 in the Resolution virtual network B, for VNETA-VM1.contoso.com, will receive a DNS response containing the Private IP of VNETA-VM1.

- A Reverse DNS (PTR) query from VNETB-VM2 in the Resolution virtual network B, for 10.1.0.1, will receive a DNS response containing the FQDN VNETB-VM1.contoso.com.
- A Reverse DNS (PTR) query from VNETB-VM3 in the Resolution virtual network B, for 10.0.0.1, will receive NXDOMAIN. The reason is that Reverse DNS queries are only scoped to the same virtual network.



Private Zone records

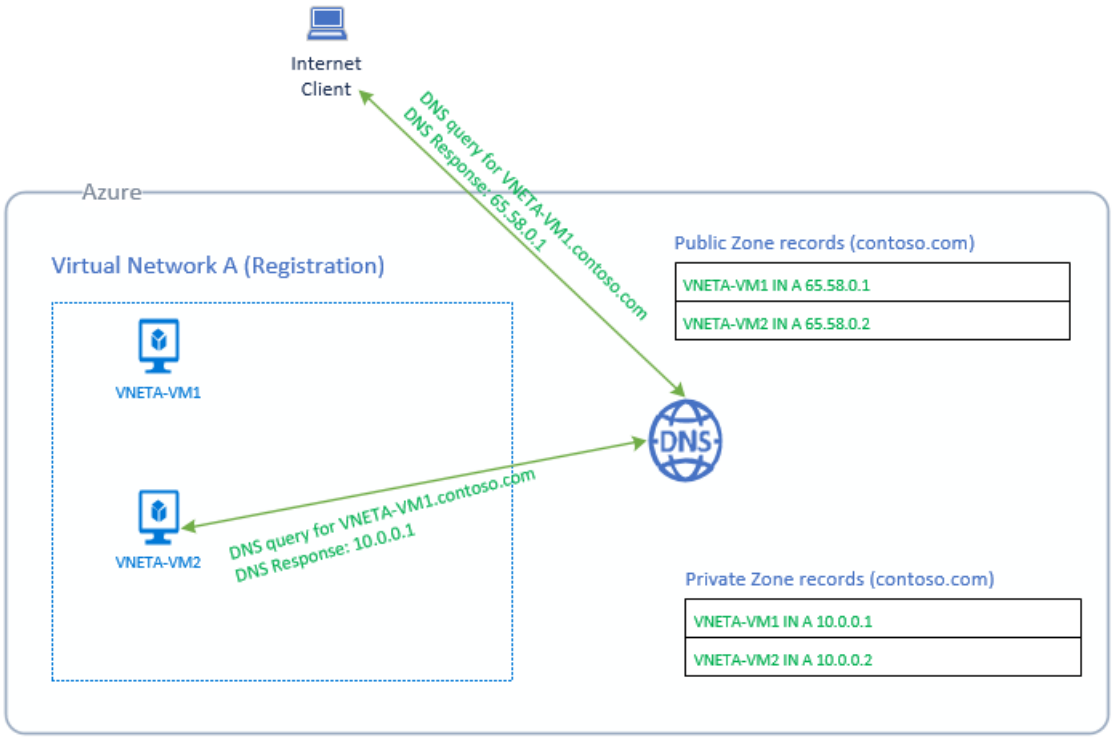
VNETA-VM1 IN A 10.0.0.1 (auto registered)
VNETA-VM2 IN A 10.0.0.2 (auto registered)
VNETB-VM1 IN A 10.1.0.1 (manual record)
VNETB-VM2 IN A 10.1.0.2 (manual record)
VNETB-VM3 IN A 10.1.0.3 (manual record)

Scenario: Split-Horizon functionality

In this scenario, you have a use case where you want to realize different DNS resolution behavior depending on where the client sits (inside of Azure or out on the internet), for the same DNS zone. For example, you may have a private and public version of your application that has different functionality or behavior, but you want to use the same domain name for both versions. This scenario can be realized with Azure DNS by creating a Public DNS zone as well as a Private Zone, with the same name.

The following diagram depicts this scenario. You have a virtual network A that has two VMs (VNETA-VM1 and VNETA-VM2) which have both Private IPs and Public IPs allocated. You create a Public DNS zone called contoso.com and register the Public IPs for these VMs as DNS records within the zone. You also create a Private DNS zone also called contoso.com specifying A as the Registration virtual network. Azure automatically registers the VMs as A records into the Private Zone, pointing to their Private IPs.

Now when an internet client issues a DNS query to look up VNETA-VM1.contoso.com, Azure will return the Public IP record from the public zone. If the same DNS query is issued from another VM (for example: VNETA-VM2) in the same virtual network A, Azure will return the Private IP record from the private zone.



Next steps

To learn more about private DNS zones, see Using Azure DNS for private domains.

Learn how to create a private DNS zone in Azure DNS.

Learn about DNS zones and records by visiting: DNS zones and records overview.

Learn about some of the other key networking capabilities of Azure.

Name resolution for resources in Azure virtual networks

2/18/2020 • 13 minutes to read • Edit Online

Depending on how you use Azure to host IaaS, PaaS, and hybrid solutions, you might need to allow the virtual machines (VMs), and other resources deployed in a virtual network to communicate with each other. Although you can enable communication by using IP addresses, it is much simpler to use names that can be easily remembered, and do not change.

When resources deployed in virtual networks need to resolve domain names to internal IP addresses, they can use one of two methods:

- Azure-provided name resolution
- Name resolution that uses your own DNS server (which might forward queries to the Azure-provided DNS servers)

The type of name resolution you use depends on how your resources need to communicate with each other. The following table illustrates scenarios and corresponding name resolution solutions:

> **NOTE**
> Depending on your scenario, you might want to use Azure DNS private zones. For more information, see Using Azure DNS for private domains.

SCENARIO	SOLUTION	SUFFIX
Name resolution between VMs located in the same virtual network, or Azure Cloud Services role instances in the same cloud service.	Azure DNS private zones or Azure-provided name resolution	Hostname or FQDN
Name resolution between VMs in different virtual networks or role instances in different cloud services.	Azure DNS private zones or, Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server.	FQDN only
Name resolution from an Azure App Service (Web App, Function, or Bot) using virtual network integration to role instances or VMs in the same virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server.	FQDN only
Name resolution from App Service Web Apps to VMs in the same virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server.	FQDN only

SCENARIO	SOLUTION	SUFFIX
Name resolution from App Service Web Apps in one virtual network to VMs in a different virtual network.	Customer-managed DNS servers forwarding queries between virtual networks for resolution by Azure (DNS proxy). See Name resolution using your own DNS server.	FQDN only
Resolution of on-premises computer and service names from VMs or role instances in Azure.	Customer-managed DNS servers (on-premises domain controller, local read-only domain controller, or a DNS secondary synced using zone transfers, for example). See Name resolution using your own DNS server.	FQDN only
Resolution of Azure hostnames from on-premises computers.	Forward queries to a customer-managed DNS proxy server in the corresponding virtual network, the proxy server forwards queries to Azure for resolution. See Name resolution using your own DNS server.	FQDN only
Reverse DNS for internal IPs.	Name resolution using your own DNS server.	Not applicable
Name resolution between VMs or role instances located in different cloud services, not in a virtual network.	Not applicable. Connectivity between VMs and role instances in different cloud services is not supported outside a virtual network.	Not applicable

Azure-provided name resolution

Along with resolution of public DNS names, Azure provides internal name resolution for VMs and role instances that reside within the same virtual network or cloud service. VMs and instances in a cloud service share the same DNS suffix, so the host name alone is sufficient. But in virtual networks deployed using the classic deployment model, different cloud services have different DNS suffixes. In this situation, you need the FQDN to resolve names between different cloud services. In virtual networks deployed using the Azure Resource Manager deployment model, the DNS suffix is consistent across the virtual network, so the FQDN is not needed. DNS names can be assigned to both VMs and network interfaces. Although Azure-provided name resolution does not require any configuration, it is not the appropriate choice for all deployment scenarios, as detailed in the previous table.

> **NOTE**
> When using cloud services web and worker roles, you can also access the internal IP addresses of role instances using the Azure Service Management REST API. For more information, see the Service Management REST API Reference. The address is based on the role name and instance number.

Features

Azure-provided name resolution includes the following features:

- Ease of use. No configuration is required.
- High availability. You don't need to create and manage clusters of your own DNS servers.
- You can use the service in conjunction with your own DNS servers, to resolve both on-premises and Azure host names.
- You can use name resolution between VMs and role instances within the same cloud service, without the need

for an FQDN.

- You can use name resolution between VMs in virtual networks that use the Azure Resource Manager deployment model, without need for an FQDN. Virtual networks in the classic deployment model require an FQDN when you are resolving names in different cloud services.
- You can use host names that best describe your deployments, rather than working with auto-generated names.

Considerations

Points to consider when you are using Azure-provided name resolution:

- The Azure-created DNS suffix cannot be modified.
- You cannot manually register your own records.
- WINS and NetBIOS are not supported. You cannot see your VMs in Windows Explorer.
- Host names must be DNS-compatible. Names must use only 0-9, a-z, and '-', and cannot start or end with a '-'.
- DNS query traffic is throttled for each VM. Throttling shouldn't impact most applications. If request throttling is observed, ensure that client-side caching is enabled. For more information, see DNS client configuration.
- Only VMs in the first 180 cloud services are registered for each virtual network in a classic deployment model. This limit does not apply to virtual networks in Azure Resource Manager.
- The Azure DNS IP address is 168.63.129.16. This is a static IP address and will not change.

DNS client configuration

This section covers client-side caching and client-side retries.

Client-side caching

Not every DNS query needs to be sent across the network. Client-side caching helps reduce latency and improve resilience to network blips, by resolving recurring DNS queries from a local cache. DNS records contain a time-to-live (TTL) mechanism, which allows the cache to store the record for as long as possible without impacting record freshness. Thus, client-side caching is suitable for most situations.

The default Windows DNS client has a DNS cache built-in. Some Linux distributions do not include caching by default. If you find that there isn't a local cache already, add a DNS cache to each Linux VM.

There are a number of different DNS caching packages available (such as dnsmasq). Here's how to install dnsmasq on the most common distributions:

- **Ubuntu (uses resolvconf)**:
  - Install the dnsmasq package with `sudo apt-get install dnsmasq`.
- **SUSE (uses netconf)**:
  - Install the dnsmasq package with `sudo zypper install dnsmasq`.
  - Enable the dnsmasq service with `systemctl enable dnsmasq.service`.
  - Start the dnsmasq service with `systemctl start dnsmasq.service`.
  - Edit **/etc/sysconfig/network/config**, and change *NETCONFIG_DNS_FORWARDER*="" to *dnsmasq*.
  - Update resolv.conf with `netconfig update`, to set the cache as the local DNS resolver.
- **CentOS (uses NetworkManager)**:
  - Install the dnsmasq package with `sudo yum install dnsmasq`.
  - Enable the dnsmasq service with `systemctl enable dnsmasq.service`.
  - Start the dnsmasq service with `systemctl start dnsmasq.service`.
  - Add *prepend domain-name-servers 127.0.0.1;* to **/etc/dhclient-eth0.conf**.
  - Restart the network service with `service network restart`, to set the cache as the local DNS resolver.

> **NOTE**
> The dnsmasq package is only one of many DNS caches available for Linux. Before using it, check its suitability for your particular needs, and check that no other cache is installed.

Client-side retries

DNS is primarily a UDP protocol. Because the UDP protocol doesn't guarantee message delivery, retry logic is handled in the DNS protocol itself. Each DNS client (operating system) can exhibit different retry logic, depending on the creator's preference:

- Windows operating systems retry after one second, and then again after another two seconds, four seconds, and another four seconds.
- The default Linux setup retries after five seconds. We recommend changing the retry specifications to five times, at one-second intervals.

Check the current settings on a Linux VM with `cat /etc/resolv.conf`. Look at the *options* line, for example:

```
options timeout:1 attempts:5
```

The resolv.conf file is usually auto-generated, and should not be edited. The specific steps for adding the *options* line vary by distribution:

- **Ubuntu** (uses resolvconf):
  1. Add the *options* line to **/etc/resolvconf/resolv.conf.d/tail**.
  2. Run `resolvconf -u` to update.
- **SUSE** (uses netconf):
  1. Add *timeout:1 attempts:5* to the **NETCONFIG_DNS_RESOLVER_OPTIONS=""** parameter in **/etc/sysconfig/network/config**.
  2. Run `netconfig update` to update.
- **CentOS** (uses NetworkManager):
  1. Add *echo "options timeout:1 attempts:5"* to **/etc/NetworkManager/dispatcher.d/11-dhclient**.
  2. Update with `service network restart`.

Name resolution that uses your own DNS server

This section covers VMs, role instances, and web apps.

VMs and role instances

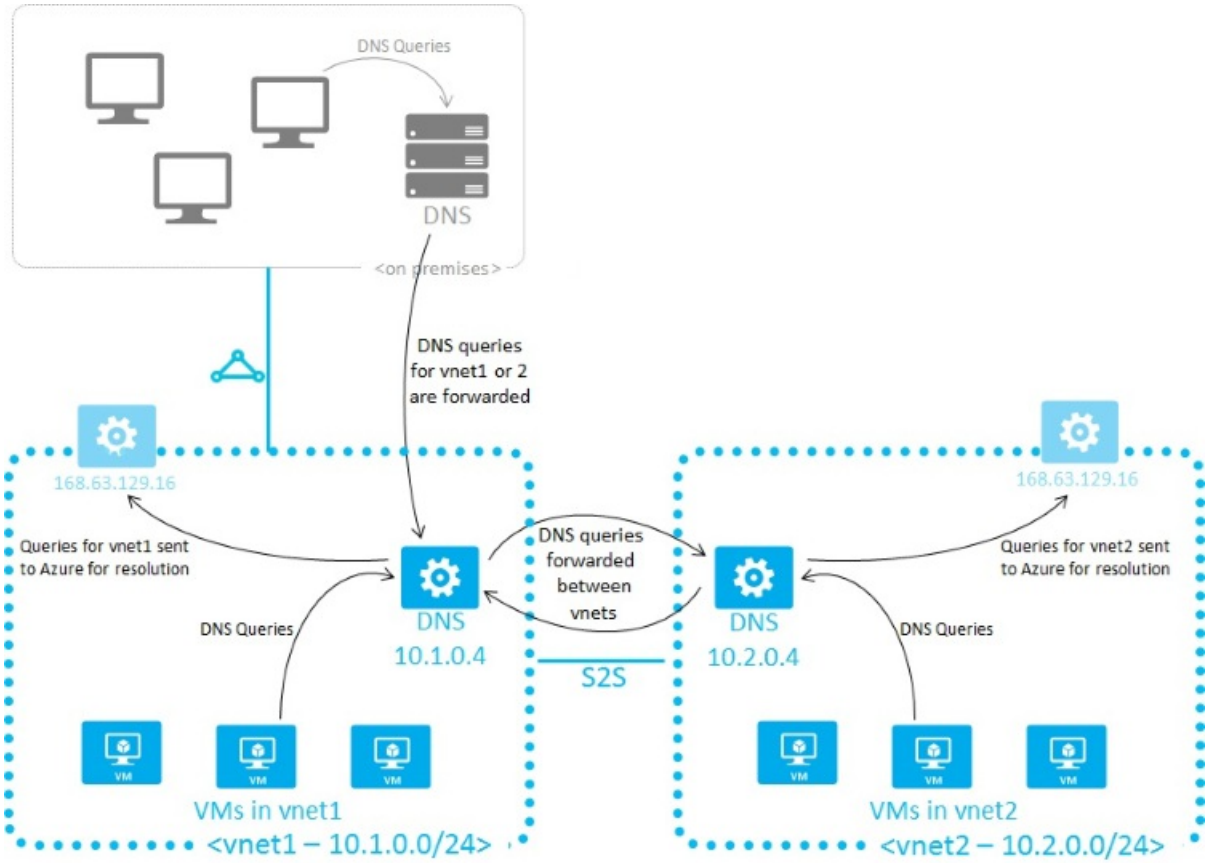
Your name resolution needs might go beyond the features provided by Azure. For example, you might need to use Microsoft Windows Server Active Directory domains, resolve DNS names between virtual networks. To cover these scenarios, Azure provides the ability for you to use your own DNS servers.

DNS servers within a virtual network can forward DNS queries to the recursive resolvers in Azure. This enables you to resolve host names within that virtual network. For example, a domain controller (DC) running in Azure can respond to DNS queries for its domains, and forward all other queries to Azure. Forwarding queries allows VMs to see both your on-premises resources (via the DC) and Azure-provided host names (via the forwarder). Access to the recursive resolvers in Azure is provided via the virtual IP 168.63.129.16.

DNS forwarding also enables DNS resolution between virtual networks, and allows your on-premises machines to resolve Azure-provided host names. In order to resolve a VM's host name, the DNS server VM must reside in the same virtual network, and be configured to forward host name queries to Azure. Because the DNS suffix is different in each virtual network, you can use conditional forwarding rules to send DNS queries to the correct

virtual network for resolution. The following image shows two virtual networks and an on-premises network doing DNS resolution between virtual networks, by using this method. An example DNS forwarder is available in the Azure Quickstart Templates gallery and GitHub.

> **NOTE**
>
> A role instance can perform name resolution of VMs within the same virtual network. It does so by using the FQDN, which consists of the VM's host name and **internal.cloudapp.net** DNS suffix. However, in this case, name resolution is only successful if the role instance has the VM name defined in the Role Schema (.cscfg file).
>
> `<Role name="<role-name>" vmName="<vm-name>">`
>
> Role instances that need to perform name resolution of VMs in another virtual network (FQDN by using the **internal.cloudapp.net** suffix) have to do so by using the method described in this section (custom DNS servers forwarding between the two virtual networks).



When you are using Azure-provided name resolution, Azure Dynamic Host Configuration Protocol (DHCP) provides an internal DNS suffix (**.internal.cloudapp.net**) to each VM. This suffix enables host name resolution because the host name records are in the **internal.cloudapp.net** zone. When you are using your own name resolution solution, this suffix is not supplied to VMs because it interferes with other DNS architectures (like domain-joined scenarios). Instead, Azure provides a non-functioning placeholder (*reddog.microsoft.com*).

If necessary, you can determine the internal DNS suffix by using PowerShell or the API:

- For virtual networks in Azure Resource Manager deployment models, the suffix is available via the network interface REST API, the Get-AzNetworkInterface PowerShell cmdlet, and the az network nic show Azure CLI command.
- In classic deployment models, the suffix is available via the Get Deployment API call or the Get-AzureVM -Debug cmdlet.

If forwarding queries to Azure doesn't suit your needs, you should provide your own DNS solution. Your DNS solution needs to:

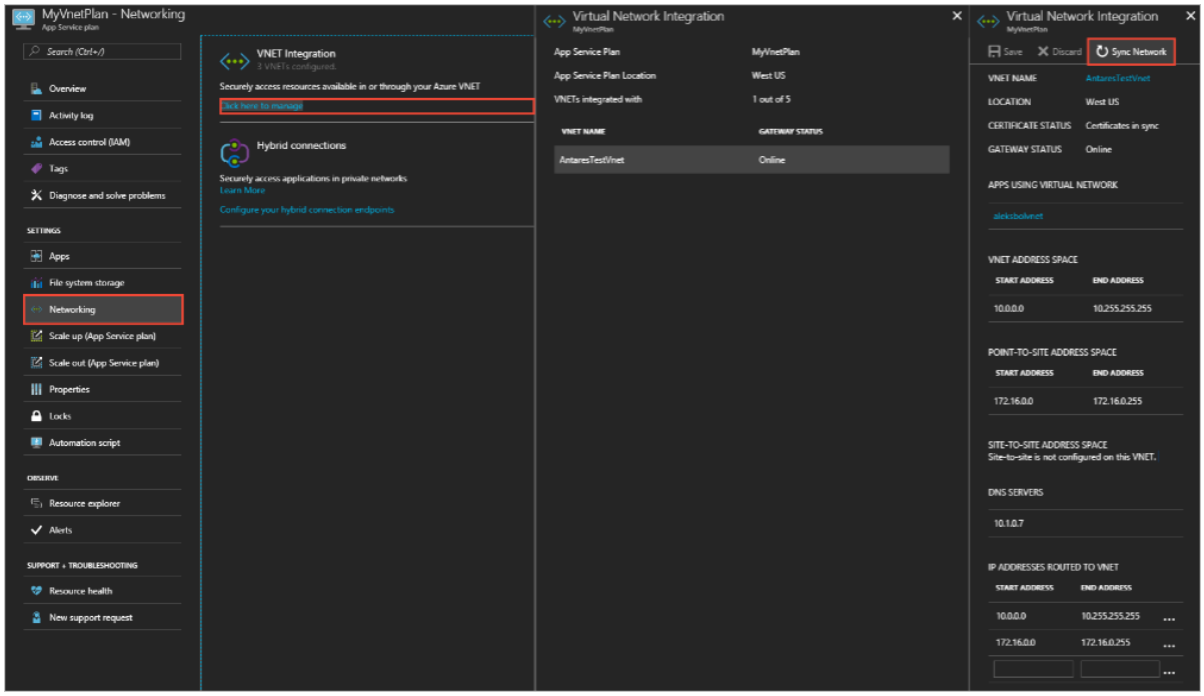
- Provide appropriate host name resolution, via DDNS, for example. If you are using DDNS, you might need to disable DNS record scavenging. Azure DHCP leases are long, and scavenging might remove DNS records prematurely.
- Provide appropriate recursive resolution to allow resolution of external domain names.
- Be accessible (TCP and UDP on port 53) from the clients it serves, and be able to access the internet.
- Be secured against access from the internet, to mitigate threats posed by external agents.

> **NOTE**
>
> For best performance, when you are using Azure VMs as DNS servers, IPv6 should be disabled. A public IP address should be assigned to each DNS server VM.

Web apps

Suppose you need to perform name resolution from your web app built by using App Service, linked to a virtual network, to VMs in the same virtual network. In addition to setting up a custom DNS server that has a DNS forwarder that forwards queries to Azure (virtual IP 168.63.129.16), perform the following steps:

1. Enable virtual network integration for your web app, if not done already, as described in Integrate your app with a virtual network.
2. In the Azure portal, for the App Service plan hosting the web app, select **Sync Network** under **Networking**, **Virtual Network Integration**.



If you need to perform name resolution from your web app built by using App Service, linked to a virtual network, to VMs in a different virtual network, you have to use custom DNS servers on both virtual networks, as follows:

- Set up a DNS server in your target virtual network, on a VM that can also forward queries to the recursive resolver in Azure (virtual IP 168.63.129.16). An example DNS forwarder is available in the Azure Quickstart Templates gallery and GitHub.
- Set up a DNS forwarder in the source virtual network on a VM. Configure this DNS forwarder to forward queries to the DNS server in your target virtual network.
- Configure your source DNS server in your source virtual network's settings.
- Enable virtual network integration for your web app to link to the source virtual network, following the instructions in Integrate your app with a virtual network.
- In the Azure portal, for the App Service plan hosting the web app, select **Sync Network** under **Networking**,

Virtual Network Integration.

Specify DNS servers

When you are using your own DNS servers, Azure provides the ability to specify multiple DNS servers per virtual network. You can also specify multiple DNS servers per network interface (for Azure Resource Manager), or per cloud service (for the classic deployment model). DNS servers specified for a network interface or cloud service get precedence over DNS servers specified for the virtual network.

> **NOTE**
> Network connection properties, such as DNS server IPs, should not be edited directly within VMs. This is because they might get erased during service heal when the virtual network adaptor gets replaced. This applies to both Windows and Linux VMs.

When you are using the Azure Resource Manager deployment model, you can specify DNS servers for a virtual network and a network interface. For details, see Manage a virtual network and Manage a network interface.

> **NOTE**
> If you opt for custom DNS server for your virtual network, you must specify at least one DNS server IP address; otherwise, virtual network will ignore the configuration and use Azure-provided DNS instead.

When you are using the classic deployment model, you can specify DNS servers for the virtual network in the Azure portal or the Network Configuration file. For cloud services, you can specify DNS servers via the Service Configuration file or by using PowerShell, with New-AzureVM.

> **NOTE**
> If you change the DNS settings for a virtual network or virtual machine that is already deployed, for the new DNS settings to take effect, you must perform a DHCP lease renewal on all affected VMs in the virtual network. For VMs running the Windows OS, you can do this by typing `ipconfig /renew` directly in the VM. The steps vary depending on the OS. See the relevant documentation for your OS type.

Next steps

Azure Resource Manager deployment model:

- Manage a virtual network
- Manage a network interface

Classic deployment model:

- Azure Service Configuration Schema
- Virtual Network Configuration Schema
- Configure a Virtual Network by using a network configuration file

Azure Private DNS FAQ

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

The following are frequently asked questions about Azure private DNS.

Does Azure DNS support private domains?

Private domains are supported using the Azure Private DNS zones feature. Private DNS zones are resolvable only from within specified virtual networks. For more information, see the overview.

For information on other internal DNS options in Azure, see Name resolution for VMs and role instances.

Will Azure Private DNS zones work across Azure regions?

Yes. Private Zones is supported for DNS resolution between virtual networks across Azure regions. Private Zones works even without explicitly peering the virtual networks. All the virtual networks must be linked to the private DNS zone.

Is connectivity to the Internet from virtual networks required for private zones?

No. Private zones work along with virtual networks. You use them to manage domains for virtual machines or other resources within and across virtual networks. Internet connectivity isn't required for name resolution.

Can the same private zone be used for several virtual networks for resolution?

Yes. You can link a private DNS zone with thousands of virtual networks. For more information, see Azure DNS Limits

Can a virtual network that belongs to a different subscription be linked to a private zone?

Yes. You must have write operation permission on the virtual networks and the private DNS zone. The write permission can be granted to several RBAC roles. For example, the Classic Network Contributor RBAC role has write permissions to virtual networks and Private DNS zones Contributor role has write permissions on the private DNS zones. For more information on RBAC roles, see Role-based access control.

Will the automatically registered virtual machine DNS records in a private zone be automatically deleted when you delete the virtual machine?

Yes. If you delete a virtual machine within a linked virtual network with autoregistration enabled, the registered records are automatically deleted.

Can an automatically registered virtual machine record in a private zone from a linked virtual network be deleted manually?

Yes. You can overwrite the automatically registered DNS records with a manually created DNS record in the zone. The following question and answer address this topic.

What happens when I try to manually create a new DNS record into a private zone that has the same hostname as an automatically registered existing virtual machine in a linked virtual network?

You try to manually create a new DNS record into a private zone that has the same hostname as an existing, automatically registered virtual machine in a linked virtual network. When you do, the new DNS record overwrites the automatically registered virtual machine record. If you try to delete this manually created DNS record from the zone again, the delete succeeds. The automatic registration happens again as long as the virtual machine still exists and has a private IP attached to it. The DNS record is re-created automatically in the zone.

What happens when we unlink a linked virtual network from a private zone? Will the automatically registered virtual machine records from the virtual network be removed from the zone too?

Yes. To unlink a linked virtual network from a private zone, you update the DNS zone to remove the associated virtual network link. In this process, virtual machine records that were automatically registered are removed from the zone.

What happens when we delete a linked virtual network that's linked to a private zone? Do we have to manually update the private zone to unlink the virtual network as a linked virtual network from the zone?

No. When you delete a linked virtual network without unlinking it from a private zone first, your deletion operation succeeds and the links to the DNS zone are automatically cleared.

Will DNS resolution by using the default FQDN (internal.cloudapp.net) still work even when a private zone (for example, private.contoso.com) is linked to a virtual network?

Yes. Private Zones don't replace the default Azure-provided internal.cloudapp.net zone. Whether you rely on the Azure-provided internal.cloudapp.net or on your own private zone, use the FQDN of the zone you want to resolve against.

Will the DNS suffix on virtual machines within a linked virtual network be changed to that of the private zone?

No. The DNS suffix on the virtual machines in your linked virtual network stays as the default Azure-provided suffix ("*.internal.cloudapp.net"). You can manually change this DNS suffix on your virtual machines to that of the private zone. For guidance on how to change this suffix refer to Use dynamic DNS to register hostnames in your own DNS server

What are the usage limits for Azure DNS Private zones?

Refer to Azure DNS limits for details on the usage limits for Azure DNS private zones.

Why don’t my existing private DNS zones show up in new portal

experience?

If your existing private DNS zone were created using preview API, you must migrate these zones to new resource model. Private DNS zones created using preview API will not show up in new portal experience. See below for instructions on how to migrate to new resource model.

How do I migrate my existing private DNS zones to the new model?

We strongly recommend that you migrate to the new resource model as soon as possible. Legacy resource model will be supported, however, further features will not be developed on top of this model. In future, we intend to deprecate it in favor of new resource model. For guidance on how to migrate your existing private DNS zones to new resource model seemigration guide for Azure DNS private zones.

Next steps

- Learn more about Azure Private DNS

Host load-balanced Azure web apps at the zone apex

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

The DNS protocol prevents the assignment of anything other than an A or AAAA record at the zone apex. An example zone apex is contoso.com. This restriction presents a problem for application owners who have load-balanced applications behind Traffic Manager. It isn't possible to point at the Traffic Manager profile from the zone apex record. As a result, application owners must use a workaround. A redirect at the application layer must redirect from the zone apex to another domain. An example is a redirect from contoso.com to www.contoso.com. This arrangement presents a single point of failure for the redirect function.

With alias records, this problem no longer exists. Now application owners can point their zone apex record to a Traffic Manager profile that has external endpoints. Application owners can point to the same Traffic Manager profile that's used for any other domain within their DNS zone.

For example, contoso.com and www.contoso.com can point to the same Traffic Manager profile. This is the case as long as the Traffic Manager profile has only external endpoints configured.

In this article, you learn how to create an alias record for your domain apex, and configure your Traffic Manager profile end points for your web apps.

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 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 a resource group

Create a resource group to hold all the resources used in this article.

Create App Service plans

Create two Web App Service plans in your resource group using the following table for configuration information. For more information about creating an App Service plan, see Manage an App Service plan in Azure.

NAME	OPERATING SYSTEM	LOCATION	PRICING TIER
ASP-01	Windows	East US	Dev/Test D1-Shared
ASP-02	Windows	Central US	Dev/Test D1-Shared

Create App Services

Create two web apps, one in each App Service plan.

1. On upper left corner of the Azure portal page, select **Create a resource**.
2. Type **Web app** in the search bar and press Enter.
3. Select **Web App**.
4. Select **Create**.
5. Accept the defaults, and use the following table to configure the two web apps:

NAME (MUST BE UNIQUE WITHIN .AZUREWEBSITES.NET)	RESOURCE GROUP	RUNTIME STACK	REGION	APP SERVICE PLAN/LOCATION
App-01	Use existing Select your resource group	.NET Core 2.2	East US	ASP-01(D1)
App-02	Use existing Select your resource group	.NET Core 2.2	Central US	ASP-02(D1)

Gather some details

Now you need to note the IP address and host name for the web apps.

1. Open your resource group and select your first web app (**App-01** in this example).
2. In the left column, select **Properties**.
3. Note the address under **URL**, and under **Outbound IP Addresses** note the first IP address in the list. You'll use this information later when you configure your Traffic Manager end points.
4. Repeat for **App-02**.

Create a Traffic Manager profile

Create a Traffic Manager profile in your resource group. Use the defaults and type a unique name within the trafficmanager.net namespace.

For information about creating a Traffic Manager profile, see Quickstart: Create a Traffic Manager profile for a highly available web application.

Create endpoints

Now you can create the endpoints for the two web apps.

1. Open your resource group and select your Traffic Manager profile.
2. In the left column, select **Endpoints**.
3. Select **Add**.
4. Use the following table to configure the endpoints:

TYPE	NAME	TARGET	LOCATION	CUSTOM HEADER SETTINGS

TYPE	NAME	TARGET	LOCATION	CUSTOM HEADER SETTINGS
External endpoint	End-01	IP address you recorded for App-01	East US	host:<the URL you recorded for App-01> Example: **host:app-01.azurewebsites.net**
External endpoint	End-02	IP address you recorded for App-02	Central US	host:<the URL you recorded for App-02> Example: **host:app-02.azurewebsites.net**

Create DNS zone

You can either use an existing DNS zone for testing, or you can create a new zone. To create and delegate a new DNS zone in Azure, see Tutorial: Host your domain in Azure DNS.

Add a TXT record for custom domain validation

When you add a custom hostname to your web apps, it will look for a specific TXT record to validate your domain.

1. Open your resource group and select the DNS zone.
2. Select **Record set**.
3. Add the record set using the following table. For the value, use the actual web app URL that you previously recorded:

NAME	TYPE	VALUE
@	TXT	App-01.azurewebsites.net

Add a custom domain

Add a custom domain for both web apps.

1. Open your resource group and select your first web app.
2. In the left column, select **Custom domains**.
3. Under **Custom Domains**, select **Add custom domain**.
4. Under **Custom domain**, type your custom domain name. For example, contoso.com.
5. Select **Validate**.

   Your domain should pass validation and show green check marks next to **Hostname availability** and **Domain ownership**.
6. Select **Add custom domain**.
7. To see the new hostname under **Hostnames assigned to site**, refresh your browser. The refresh on the page doesn't always show changes right away.

8. Repeat this procedure for your second web app.

Add the alias record set

Now add an alias record for the zone apex.

1. Open your resource group and select the DNS zone.
2. Select **Record set**.
3. Add the record set using the following table:

NAME	TYPE	ALIAS RECORD SET	ALIAS TYPE	AZURE RESOURCE
@	A	Yes	Azure resource	Traffic Manager - your profile

Test your web apps

Now you can test to make sure you can reach your web app and that it's being load balanced.

1. Open a web browser and browse to your domain. For example, contoso.com. You should see the default web app page.
2. Stop your first web app.
3. Close your web browser, and wait a few minutes.
4. Start your web browser and browse to your domain. You should still see the default web app page.
5. Stop your second web app.
6. Close your web browser, and wait a few minutes.
7. Start your web browser and browse to your domain. You should see Error 403, indicating that the web app is stopped.
8. Start your second web app.
9. Close your web browser, and wait a few minutes.
10. Start your web browser and browse to your domain. You should see the default web app page again.

Next steps

To learn more about alias records, see the following articles:

- Tutorial: Configure an alias record to refer to an Azure public IP address
- Tutorial: Configure an alias record to support apex domain names with Traffic Manager
- DNS FAQ

To learn how to migrate an active DNS name, see Migrate an active DNS name to Azure App Service.

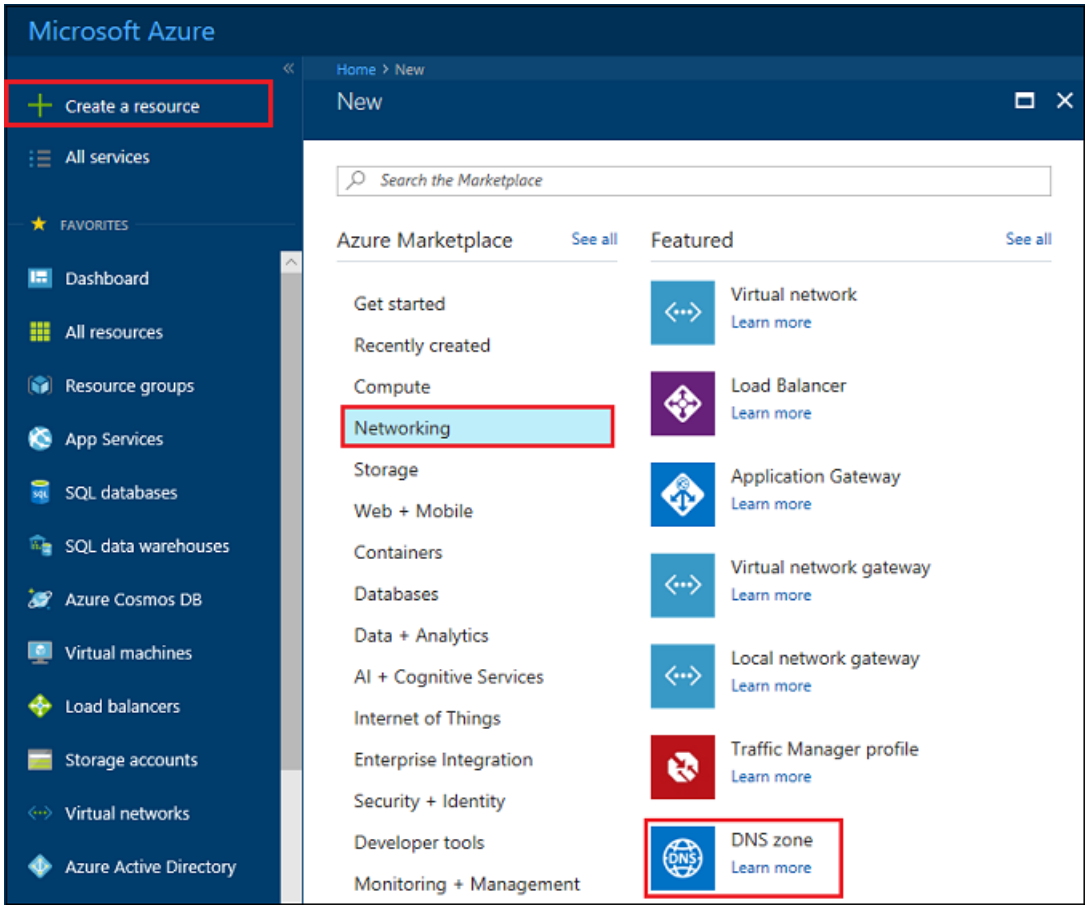
How to manage DNS Zones in the Azure portal

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

This article shows you how to manage your DNS zones by using the Azure portal. You can also manage your DNS zones using the cross-platform Azure CLI or the Azure PowerShell.

Create a DNS zone

1. Sign in to the Azure portal
2. On the Hub menu, navigate to **Create a resource > Networking > DNS zone** to open the **Create DNS zone** blade.



3. On the **Create DNS zone** blade enter the following values, then click **Create**:

SETTING	VALUE	DETAILS
Name	contoso.com	The name of the DNS zone
Subscription	[Your subscription]	Select a subscription to create the DNS zone in.

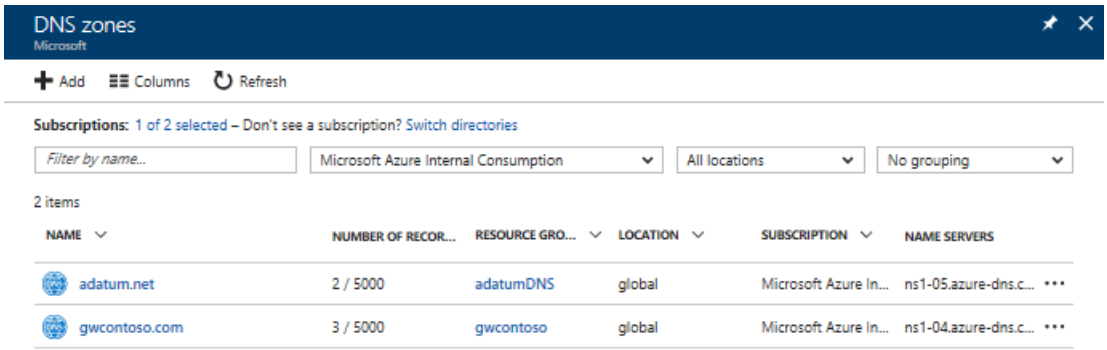
SETTING	VALUE	DETAILS
Resource group	**Create new:** contosoDNSRG	Create a resource group. The resource group name must be unique within the subscription you selected. To learn more about resource groups, read the Resource Manager overview article.
Location	West US	

NOTE

The resource group refers to the location of the resource group, and has no impact on the DNS zone. The DNS zone location is always "global", and is not shown.

List DNS zones

In the Azure portal, navigate to **More services** > **Networking** > **DNS zones**. Each DNS zone is its own resource, and information such as number of record-sets and name servers are viewable from this view. The column **NAME SERVERS** is not in the default view. To add it, click **Columns**, select **Name servers**, and then click **Done**.



Delete a DNS zone

Navigate to a DNS zone in the portal. On the **DNS zone** blade, click **Delete zone**. You are then prompted to confirm you are wanting to delete the DNS zone. Deleting a DNS zone also deletes all records that are contained in the zone.

Next steps

Learn how to work with your DNS Zone and records by visiting Get started with Azure DNS using the Azure portal.

How to manage DNS Zones using PowerShell

2/1/2020 • 7 minutes to read • Edit Online

This article shows you how to manage your DNS zones by using Azure PowerShell. You can also manage your DNS zones using the cross-platform Azure CLI or the Azure portal.

This guide specifically deals with Public DNS zones. For information on using Azure PowerShell to manage Private Zones in Azure DNS, see Get started with Azure DNS Private Zones using Azure PowerShell.

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone.

For example, the domain 'contoso.com' may contain several DNS records, such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site).

When creating a DNS zone in Azure DNS:

- The name of the zone must be unique within the resource group, and the zone must not exist already. Otherwise, the operation fails.
- The same zone name can be reused in a different resource group or a different Azure subscription.
- Where multiple zones share the same name, each instance is assigned different name server addresses. Only one set of addresses can be configured with the domain name registrar.

> **NOTE**
>
> You do not have to own a domain name to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to configure the Azure DNS name servers as the correct name servers for the domain name with the domain name registrar.
>
> For more information, see Delegate a domain to Azure DNS.

Set up Azure PowerShell for Azure DNS

Before you begin

> **IMPORTANT**
>
> Using this Azure feature from PowerShell requires the `AzureRM` module installed. This is an older module only available for Windows PowerShell 5.1 that no longer receives new features. The `Az` and `AzureRM` modules are **not** compatible when installed for the same versions of PowerShell. If you need both versions:
>
> 1. Uninstall the Az module from a PowerShell 5.1 session.
> 2. Install the AzureRM module from a PowerShell 5.1 session.
> 3. Download and install PowerShell Core 6.x or later.
> 4. Install the Az module in a PowerShell Core session.

Verify that you have the following items before beginning your configuration.

- An Azure subscription. If you don't already have an Azure subscription, you can activate your MSDN subscriber benefits or sign up for a free account.
- You need to install the latest version of the Azure Resource Manager PowerShell cmdlets. For more

information, see How to install and configure Azure PowerShell.

In addition, to use Private Zones (Public Preview), you need to ensure you have the below PowerShell modules and versions.

- AzureRM.Dns - version 4.1.0 or above
- AzureRM.Network - version 5.4.0 or above

```
Find-Module -Name AzureRM.Dns
```

```
Find-Module -Name AzureRM.Network
```

The output of the above commands needs to show that the version of AzureRM.Dns is 4.1.0 or higher version, and for AzureRM.Network is 5.4.0 or higher version.

In case your system has earlier versions, you can either install the latest version of Azure PowerShell, or download and install the above modules from the PowerShell Gallery, using the links above next to the Module versions. You can then install them using the below commands. Both the modules are required and are fully backwards compatible.

```
Install-Module -Name AzureRM.Dns -Force
```

```
Install-Module -Name AzureRM.Network -Force
```

Sign in to your Azure account

Open your PowerShell console and connect to your account. For more information, see Sign in with AzureRM.

```
Connect-AzureRmAccount
```

Select the subscription

Check the subscriptions for the account.

```
Get-AzureRmSubscription
```

Choose which of your Azure subscriptions to use.

```
Select-AzureRmSubscription -SubscriptionName "your_subscription_name"
```

Create a resource group

Azure Resource Manager requires that all resource groups specify a location. This location is used as the default location for resources in that resource group. However, because all DNS resources are global, not regional, the choice of resource group location has no impact on Azure DNS.

You can skip this step if you are using an existing resource group.

```
New-AzureRmResourceGroup -Name MyAzureResourceGroup -location "West US"
```

Register resource provider

The Azure DNS service is managed by the Microsoft.Network resource provider. Your Azure subscription must be registered to use this resource provider before you can use Azure DNS. This is a one-time operation for each subscription.

```
Register-AzureRmResourceProvider -ProviderNamespace Microsoft.Network
```

Create a DNS zone

A DNS zone is created by using the `New-AzureRmDnsZone` cmdlet.

The following example creates a DNS zone called *contoso.com* in the resource group called *MyResourceGroup*:

```
New-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup
```

The following example shows how to create a DNS zone with two Azure Resource Manager tags, *project = demo* and *env = test*:

```
New-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup -Tag @{ project="demo";
env="test" }
```

Azure DNS also supports private DNS zones. To learn more about private DNS zones, see Using Azure DNS for private domains. For an example of how to create a private DNS zone, see Get started with Azure DNS private zones using PowerShell.

Get a DNS zone

To retrieve a DNS zone, use the `Get-AzureRmDnsZone` cmdlet. This operation returns a DNS zone object corresponding to an existing zone in Azure DNS. The object contains data about the zone (such as the number of record sets), but does not contain the record sets themselves (see `Get-AzureRmDnsRecordSet`).

```
Get-AzureRmDnsZone -Name contoso.com –ResourceGroupName MyAzureResourceGroup

Name                  : contoso.com
ResourceGroupName     : myresourcegroup
Etag                  : 00000003-0000-0000-8ec2-f4879750d201
Tags                  : {project, env}
NameServers           : {ns1-01.azure-dns.com., ns2-01.azure-dns.net., ns3-01.azure-dns.org.,
                        ns4-01.azure-dns.info.}
NumberOfRecordSets    : 2
MaxNumberOfRecordSets : 5000
```

List DNS zones

By omitting the zone name from `Get-AzureRmDnsZone`, you can enumerate all zones in a resource group. This operation returns an array of zone objects.

```
$zoneList = Get-AzureRmDnsZone -ResourceGroupName MyAzureResourceGroup
```

By omitting both the zone name and the resource group name from `Get-AzureRmDnsZone`, you can enumerate all zones in the Azure subscription.

```
$zoneList = Get-AzureRmDnsZone
```

Update a DNS zone

Changes to a DNS zone resource can be made by using `Set-AzureRmDnsZone`. This cmdlet does not update any of the DNS record sets within the zone (see How to Manage DNS records). It's only used to update properties of the zone resource itself. The writable zone properties are currently limited to the Azure Resource Manager 'tags' for the zone resource.

Use one of the following two ways to update a DNS zone:

Specify the zone using the zone name and resource group

This approach replaces the existing zone tags with the values specified.

```
Set-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup -Tag @{ project="demo";
env="test" }
```

Specify the zone using a $zone object

This approach retrieves the existing zone object, modifies the tags, and then commits the changes. In this way, existing tags can be preserved.

```
# Get the zone object
$zone = Get-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup

# Remove an existing tag
$zone.Tags.Remove("project")

# Add a new tag
$zone.Tags.Add("status","approved")

# Commit changes
Set-AzureRmDnsZone -Zone $zone
```

When using `Set-AzureRmDnsZone` with a $zone object, Etag checks are used to ensure concurrent changes are not overwritten. You can use the optional `-Overwrite` switch to suppress these checks.

Delete a DNS Zone

DNS zones can be deleted using the `Remove-AzureRmDnsZone` cmdlet.

> **NOTE**
>
> Deleting a DNS zone also deletes all DNS records within the zone. This operation cannot be undone. If the DNS zone is in use, services using the zone will fail when the zone is deleted.
>
> To protect against accidental zone deletion, see How to protect DNS zones and records.

Use one of the following two ways to delete a DNS zone:

Specify the zone using the zone name and resource group name

```
Remove-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup
```

Specify the zone using a $zone object

You can specify the zone to be deleted using a `$zone` object returned by `Get-AzureRmDnsZone`.

```
$zone = Get-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup
Remove-AzureRmDnsZone -Zone $zone
```

The zone object can also be piped instead of being passed as a parameter:

```
Get-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyAzureResourceGroup | Remove-AzureRmDnsZone
```

As with `Set-AzureRmDnsZone`, specifying the zone using a `$zone` object enables Etag checks to ensure concurrent changes are not deleted. Use the `-Overwrite` switch to suppress these checks.

Confirmation prompts

The `New-AzureRmDnsZone`, `Set-AzureRmDnsZone`, and `Remove-AzureRmDnsZone` cmdlets all support confirmation prompts.

Both `New-AzureRmDnsZone` and `Set-AzureRmDnsZone` prompt for confirmation if the `$ConfirmPreference` PowerShell preference variable has a value of `Medium` or lower. Due to the potentially high impact of deleting a DNS zone, the `Remove-AzureRmDnsZone` cmdlet prompts for confirmation if the `$ConfirmPreference` PowerShell variable has any value other than `None`.

Since the default value for `$ConfirmPreference` is `High`, only `Remove-AzureRmDnsZone` prompts for confirmation by default.

You can override the current `$ConfirmPreference` setting using the `-Confirm` parameter. If you specify `-Confirm` or `-Confirm:$True`, the cmdlet prompts you for confirmation before it runs. If you specify `-Confirm:$False`, the cmdlet does not prompt you for confirmation.

For more information about `-Confirm` and `$ConfirmPreference`, see About Preference Variables.

Next steps

Learn how to manage record sets and records in your DNS zone.
Learn how to delegate your domain to Azure DNS.
Review the Azure DNS PowerShell reference documentation.

How to manage DNS Zones in Azure DNS using the Azure CLI

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

This guide shows how to manage your DNS zones by using the cross-platform Azure CLI, which is available for Windows, Mac and Linux. You can also manage your DNS zones using Azure PowerShell or the Azure portal.

This guide specifically deals with Public DNS zones. For information on using Azure CLI to manage Private Zones in Azure DNS, see Get started with Azure DNS Private Zones using Azure CLI.

Introduction

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone.

For example, the domain 'contoso.com' may contain several DNS records, such as 'mail.contoso.com' (for a mail server) and 'www.contoso.com' (for a web site).

When creating a DNS zone in Azure DNS:

- The name of the zone must be unique within the resource group, and the zone must not exist already. Otherwise, the operation fails.
- The same zone name can be reused in a different resource group or a different Azure subscription.
- Where multiple zones share the same name, each instance is assigned different name server addresses. Only one set of addresses can be configured with the domain name registrar.

> **NOTE**
>
> You do not have to own a domain name to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to configure the Azure DNS name servers as the correct name servers for the domain name with the domain name registrar.
>
> For more information, see Delegate a domain to Azure DNS.

Set up Azure CLI for Azure DNS

Before you begin

Verify that you have the following items before beginning your configuration.

- An Azure subscription. If you don't already have an Azure subscription, you can activate your MSDN subscriber benefits or sign up for a free account.
- Install the latest version of the Azure CLI, available for Windows, Linux, or MAC. More information is available at Install the Azure CLI.

Sign in to your Azure account

Open a console window and authenticate with your credentials. For more information, see Log in to Azure from the Azure CLI

```
az login
```

Select the subscription

Check the subscriptions for the account.

```
az account list
```

Choose which of your Azure subscriptions to use.

```
az account set --subscription "subscription name"
```

Optional: To install/use Azure DNS Private Zones feature

The Azure DNS Private Zone feature is available via an extension to the Azure CLI. Install the "dns" Azure CLI extension

```
az extension add --name dns
```

Create a resource group

Azure Resource Manager requires that all resource groups specify a location. This is used as the default location for resources in that resource group. However, because all DNS resources are global, not regional, the choice of resource group location has no impact on Azure DNS.

You can skip this step if you are using an existing resource group.

```
az group create --name myresourcegroup --location "West US"
```

Getting help

All Azure CLI commands relating to Azure DNS start with `az network dns`. Help is available for each command using the `--help` option (short form `-h`). For example:

```
az network dns --help
az network dns zone --help
az network dns zone create --help
```

Create a DNS zone

A DNS zone is created using the `az network dns zone create` command. For help, see `az network dns zone create -h`.

The following example creates a DNS zone called *contoso.com* in the resource group called *MyResourceGroup*:

```
az network dns zone create --resource-group MyResourceGroup --name contoso.com
```

To create a DNS zone with tags

The following example shows how to create a DNS zone with two Azure Resource Manager tags, *project* = *demo* and *env* = *test*, by using the `--tags` parameter (short form `-t`):

```
az network dns zone create --resource-group MyResourceGroup --name contoso.com --tags "project=demo"
"env=test"
```

Get a DNS zone

To retrieve a DNS zone, use `az network dns zone show`. For help, see `az network dns zone show --help`.

The following example returns the DNS zone *contoso.com* and its associated data from resource group *MyResourceGroup*.

```
az network dns zone show --resource-group myresourcegroup --name contoso.com
```

The following example is the response.

```
{
  "etag": "00000002-0000-0000-3d4d-64aa3689d201",
  "id": "/subscriptions/147a22e9-2356-4e56-b3de-
1f5842ae4a3b/resourceGroups/myresourcegroup/providers/Microsoft.Network/dnszones/contoso.com",
  "location": "global",
  "maxNumberOfRecordSets": 5000,
  "name": "contoso.com",
  "nameServers": [
    "ns1-04.azure-dns.com.",
    "ns2-04.azure-dns.net.",
    "ns3-04.azure-dns.org.",
    "ns4-04.azure-dns.info."
  ],
  "numberOfRecordSets": 4,
  "resourceGroup": "myresourcegroup",
  "tags": {},
  "type": "Microsoft.Network/dnszones"
}
```

Note that DNS records are not returned by `az network dns zone show`. To list DNS records, use `az network dns record-set list`.

List DNS zones

To enumerate DNS zones, use `az network dns zone list`. For help, see `az network dns zone list --help`.

Specifying the resource group lists only those zones within the resource group:

```
az network dns zone list --resource-group MyResourceGroup
```

Omitting the resource group lists all zones in the subscription:

```
az network dns zone list
```

Update a DNS zone

Changes to a DNS zone resource can be made using `az network dns zone update`. For help, see `az network dns zone update --help`.

This command does not update any of the DNS record sets within the zone (see How to Manage DNS records). It

is only used to update properties of the zone resource itself. These properties are currently limited to the Azure Resource Manager 'tags' for the zone resource.

The following example shows how to update the tags on a DNS zone. The existing tags are replaced by the value specified.

```
az network dns zone update --resource-group myresourcegroup --name contoso.com --set tags.team=support
```

Delete a DNS zone

DNS zones can be deleted using `az network dns zone delete`. For help, see `az network dns zone delete --help`.

> **NOTE**
>
> Deleting a DNS zone also deletes all DNS records within the zone. This operation cannot be undone. If the DNS zone is in use, services using the zone will fail when the zone is deleted.
>
> To protect against accidental zone deletion, see How to protect DNS zones and records.

This command prompts for confirmation. The optional `--yes` switch suppresses this prompt.

The following example shows how to delete the zone *contoso.com* from resource group *MyResourceGroup*.

```
az network dns zone delete --resource-group myresourcegroup --name contoso.com
```

Next steps

Learn how to manage record sets and records in your DNS zone.

Learn how to delegate your domain to Azure DNS.

Manage DNS records and record sets by using the Azure portal

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

This article shows you how to manage record sets and records for your DNS zone by using the Azure portal.

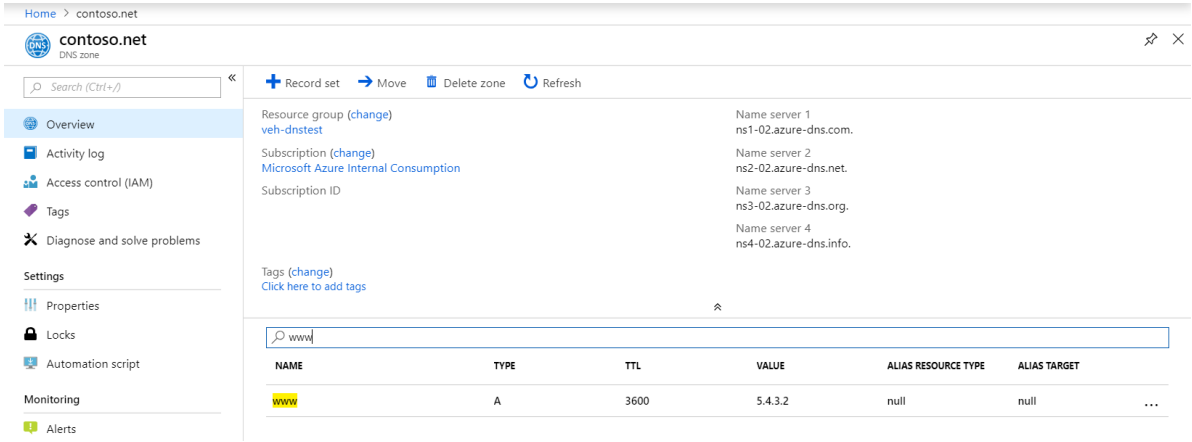
It's important to understand the difference between DNS record sets and individual DNS records. A record set is a collection of records in a zone that have the same name and are the same type. For more information, see Create DNS record sets and records by using the Azure portal.

Create a new record set and record

To create a record set in the Azure portal, see Create DNS records by using the Azure portal.

View a record set

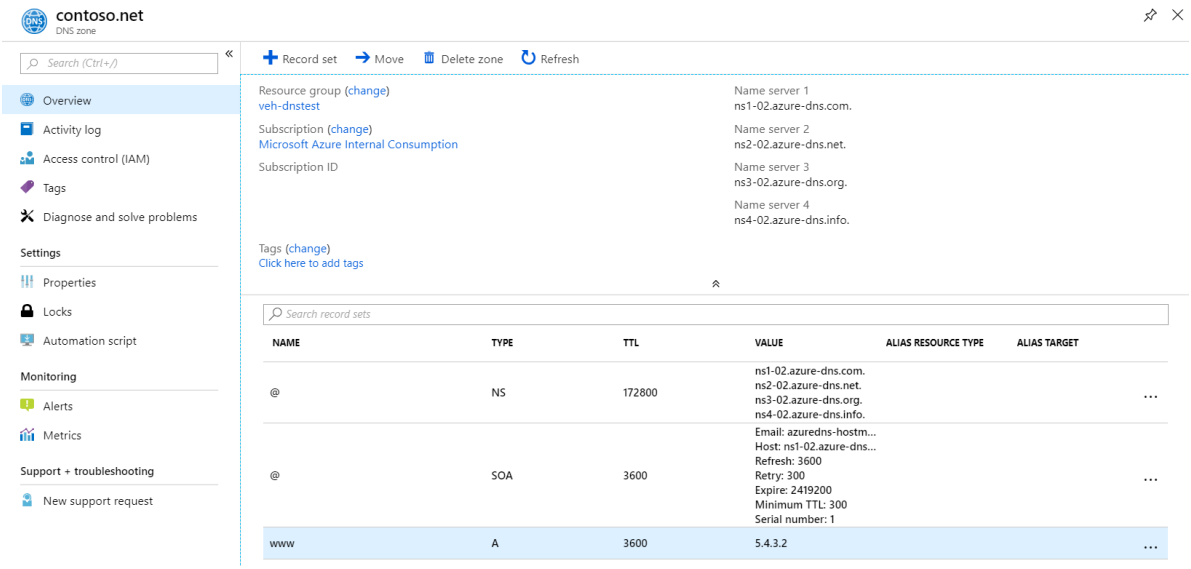
1. In the Azure portal, go to the **DNS zone** blade.
2. Search for the record set and select it. This opens the record set properties.



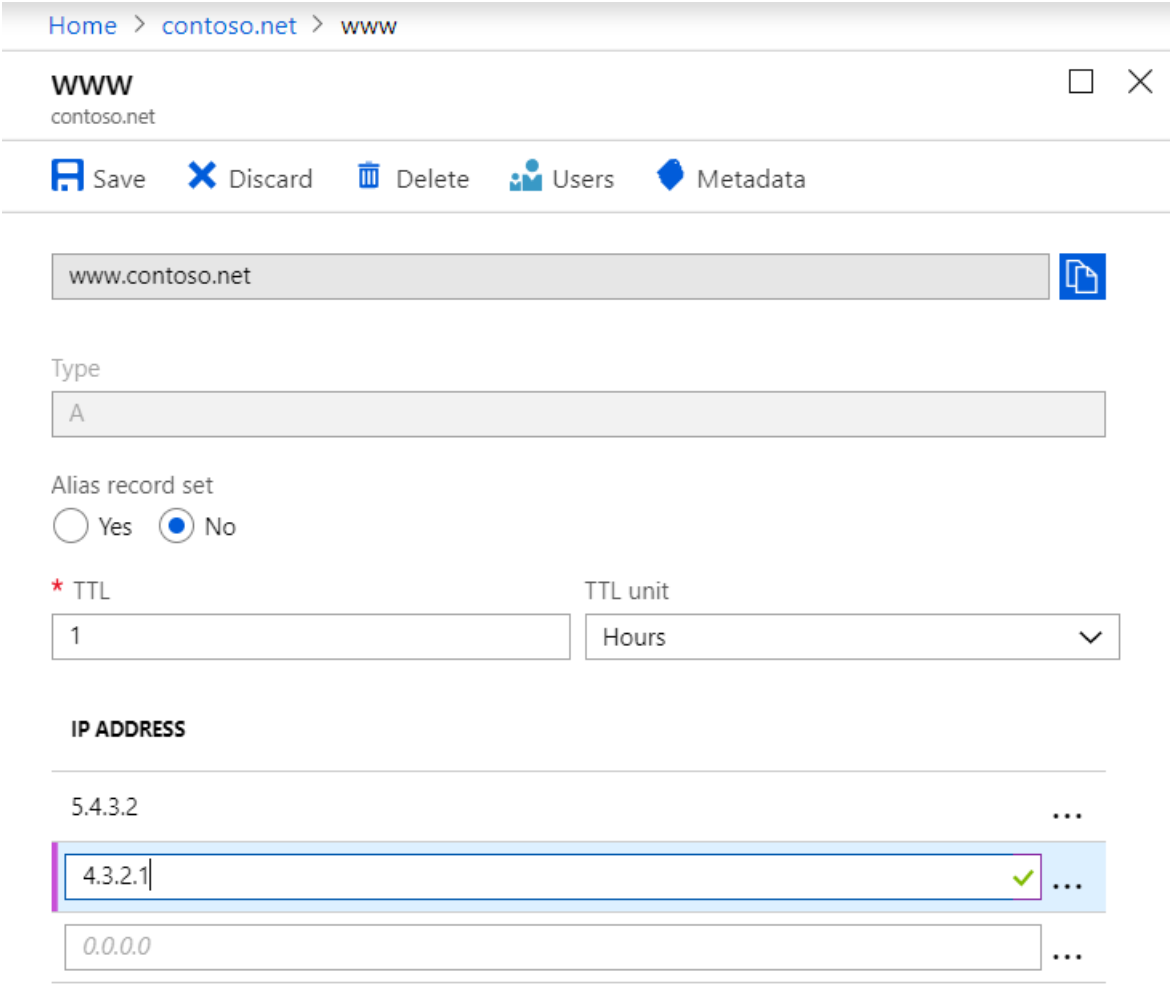
Add a new record to a record set

You can add up to 20 records to any record set. A record set cannot contain two identical records. Empty record sets (with zero records) can be created, but do not appear on the Azure DNS name servers. Record sets of type CNAME can contain one record at most.

1. On the **Record set properties** blade for your DNS zone, click the record set that you want to add a record to.

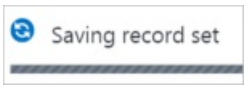


2. Specify the record set properties by filling in the fields.



3. Click **Save** at the top of the blade to save your settings. Then close the blade.

4. In the corner, you will see that the record is saving.

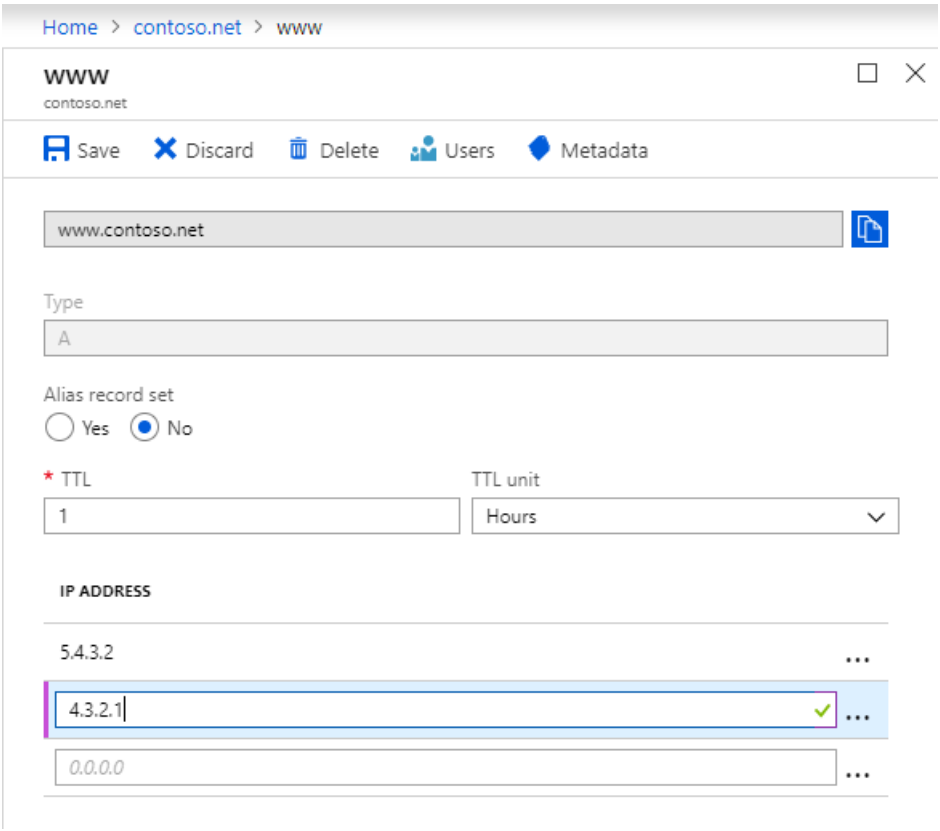


After the record has been saved, the values on the **DNS zone** blade will reflect the new record.

Update a record

When you update a record in an existing record set, the fields you can update depend on the type of record you're working with.

1. On the **Record set properties** blade for your record set, search for the record.
2. Modify the record. When you modify a record, you can change the available settings for the record. In the following example, the **IP address** field is selected, and the IP address is in the process of being modified.



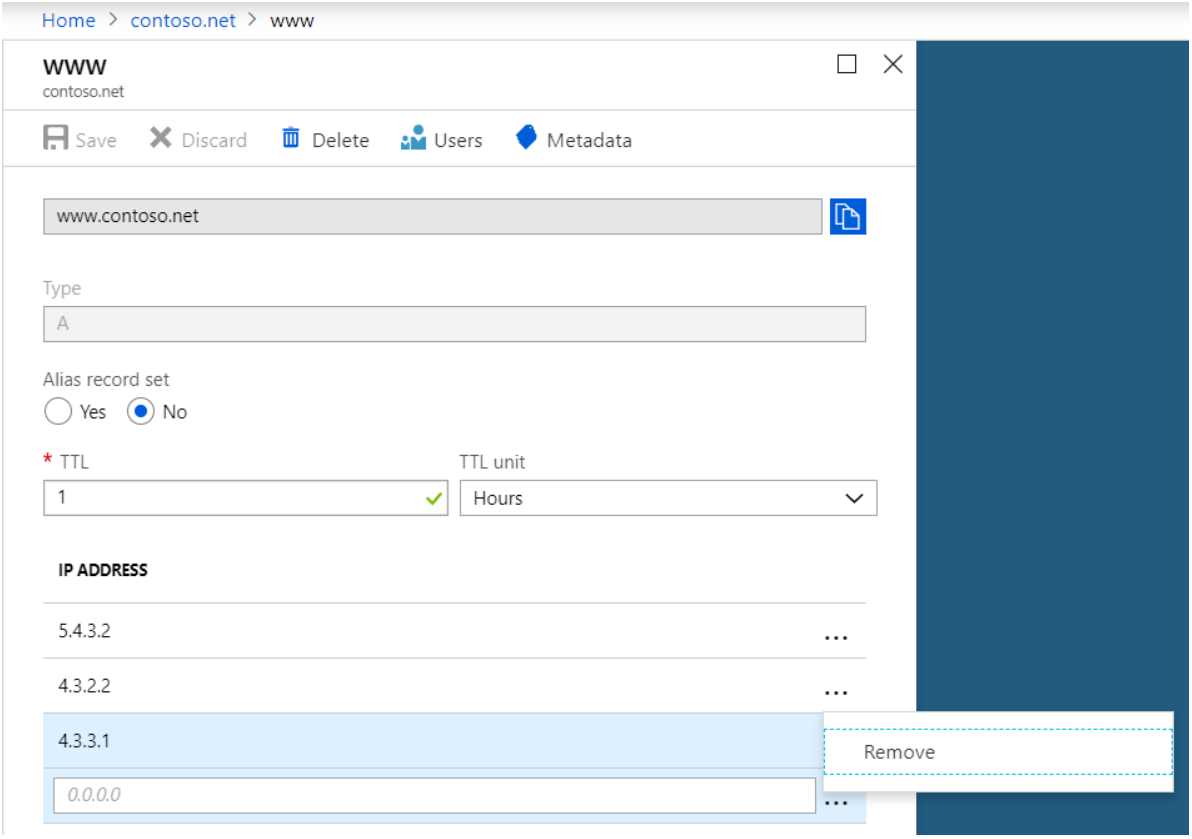
3. Click **Save** at the top of the blade to save your settings. In the upper right corner, you'll see the notification that the record has been saved.

After the record has been saved, the values for the record set on the **DNS zone** blade will reflect the updated record.

Remove a record from a record set

You can use the Azure portal to remove records from a record set. Note that removing the last record from a record set does not delete the record set.

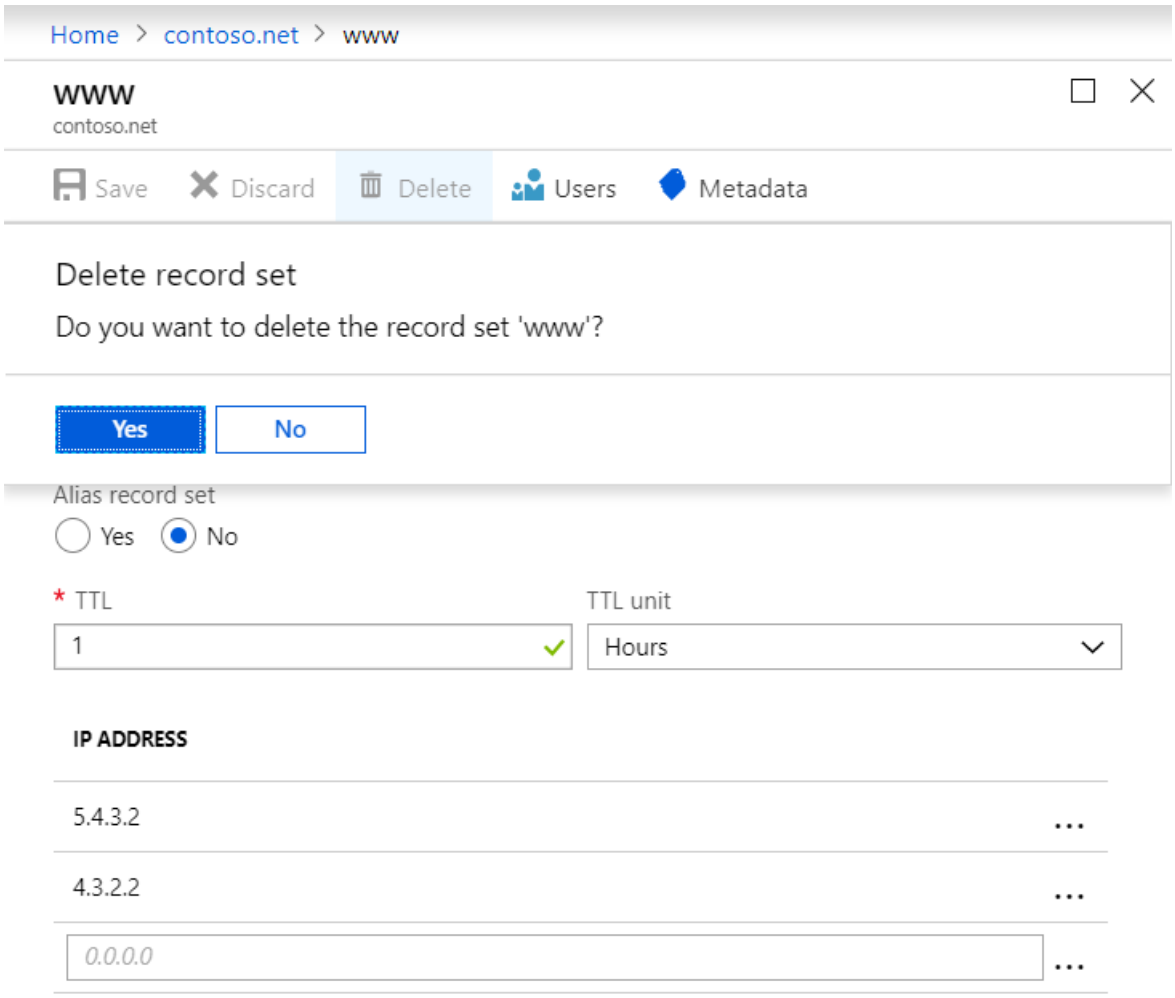
1. On the **Record set properties** blade for your record set, search for the record.
2. Click the record that you want to remove. Then select **Remove**.



3. Click **Save** at the top of the blade to save your settings.
4. After the record has been removed, the values for the record on the **DNS zone** blade will reflect the removal.

Delete a record set

1. On the **Record set properties** blade for your record set, click **Delete**.



2. A message appears asking if you want to delete the record set.
3. Verify that the name matches the record set that you want to delete, and then click **Yes**.
4. On the **DNS zone** blade, verify that the record set is no longer visible.

Work with NS and SOA records

NS and SOA records that are automatically created are managed differently from other record types.

Modify SOA records

You cannot add or remove records from the automatically created SOA record set at the zone apex (name = "@"). However, you can modify any of the parameters within the SOA record (except "Host") and the record set TTL.

Modify NS records at the zone apex

The NS record set at the zone apex is automatically created with each DNS zone. It contains the names of the Azure DNS name servers assigned to the zone.

You can add additional name servers to this NS record set, to support co-hosting domains with more than one DNS provider. You can also modify the TTL and metadata for this record set. However, you cannot remove or modify the pre-populated Azure DNS name servers.

Note that this applies only to the NS record set at the zone apex. Other NS record sets in your zone (as used to delegate child zones) can be modified without constraint.

Delete SOA or NS record sets

You cannot delete the SOA and NS record sets at the zone apex (name = "@") that are created automatically when the zone is created. They are deleted automatically when you delete the zone.

Next steps

- For more information about Azure DNS, see the Azure DNS overview.
- For more information about automating DNS, see Creating DNS zones and record sets using the .NET SDK.
- For more information about reverse DNS records, see Overview of reverse DNS and support in Azure.
- For more information about Azure DNS alias records, see Azure DNS alias records overview.

Manage DNS records and recordsets in Azure DNS using Azure PowerShell

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

This article shows you how to manage DNS records for your DNS zone by using Azure PowerShell. DNS records can also be managed by using the cross-platform Azure CLI or the Azure portal.

The examples in this article assume you have already installed Azure PowerShell, signed in, and created a DNS zone.

> **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.

Introduction

Before creating DNS records in Azure DNS, you first need to understand how Azure DNS organizes DNS records into DNS record sets.

Record names

In Azure DNS, records are specified by using relative names. A *fully qualified* domain name (FQDN) includes the zone name, whereas a *relative* name does not. For example, the relative record name `www` in the zone `contoso.com` gives the fully qualified record name `www.contoso.com`.

An *apex* record is a DNS record at the root (or *apex*) of a DNS zone. For example, in the DNS zone `contoso.com`, an apex record also has the fully qualified name `contoso.com` (this is sometimes called a *naked* domain). By convention, the relative name '@' is used to represent apex records.

Record types

Each DNS record has a name and a type. Records are organized into various types according to the data they contain. The most common type is an 'A' record, which maps a name to an IPv4 address. Another common type is an 'MX' record, which maps a name to a mail server.

Azure DNS supports all common DNS record types: A, AAAA, CAA, CNAME, MX, NS, PTR, SOA, SRV, and TXT. Note that SPF records are represented using TXT records.

Record sets

Sometimes you need to create more than one DNS record with a given name and type. For example, suppose the 'www.contoso.com' web site is hosted on two different IP addresses. The website requires two different A records, one for each IP address. Here is an example of a record set:

```
www.contoso.com.        3600    IN    A    134.170.185.46
www.contoso.com.        3600    IN    A    134.170.188.221
```

Azure DNS manages all DNS records using *record sets*. A record set (also known as a *resource* record set) is the collection of DNS records in a zone that have the same name and are of the same type. Most record sets contain

a single record. However, examples like the one above, in which a record set contains more than one record, are not uncommon.

For example, suppose you have already created an A record 'www' in the zone 'contoso.com', pointing to the IP address '134.170.185.46' (the first record above). To create the second record you would add that record to the existing record set, rather than create an additional record set.

The SOA and CNAME record types are exceptions. The DNS standards don't permit multiple records with the same name for these types, therefore these record sets can only contain a single record.

For more information about DNS records in Azure DNS, see DNS zones and records.

Create a new DNS record

If your new record has the same name and type as an existing record, you need to add it to the existing record set. If your new record has a different name and type to all existing records, you need to create a new record set.

Create 'A' records in a new record set

You create record sets by using the `New-AzDnsRecordSet` cmdlet. When creating a record set, you need to specify the record set name, the zone, the time to live (TTL), the record type, and the records to be created.

The parameters for adding records to a record set vary depending on the type of the record set. For example, when using a record set of type 'A', you need to specify the IP address using the parameter `-IPv4Address`. Other parameters are used for other record types. See Additional record type examples for details.

The following example creates a record set with the relative name 'www' in the DNS Zone 'contoso.com'. The fully-qualified name of the record set is 'www.contoso.com'. The record type is 'A', and the TTL is 3600 seconds. The record set contains a single record, with IP address '1.2.3.4'.

```
New-AzDnsRecordSet -Name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -
Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -IPv4Address "1.2.3.4")
```

To create a record set at the 'apex' of a zone (in this case, 'contoso.com'), use the record set name '@' (excluding quotation marks):

```
New-AzDnsRecordSet -Name "@" -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -Ttl
3600 -DnsRecords (New-AzDnsRecordConfig -IPv4Address "1.2.3.4")
```

If you need to create a record set containing more than one record, first create a local array and add the records, then pass the array to `New-AzDnsRecordSet` as follows:

```
$aRecords = @()
$aRecords += New-AzDnsRecordConfig -IPv4Address "1.2.3.4"
$aRecords += New-AzDnsRecordConfig -IPv4Address "2.3.4.5"
New-AzDnsRecordSet -Name www –ZoneName "contoso.com" -ResourceGroupName MyResourceGroup -Ttl 3600 -RecordType
A -DnsRecords $aRecords
```

Record set metadata can be used to associate application-specific data with each record set, as key-value pairs. The following example shows how to create a record set with two metadata entries, 'dept=finance' and 'environment=production'.

```
New-AzDnsRecordSet -Name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -
Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -IPv4Address "1.2.3.4") -Metadata @{ dept="finance";
environment="production" }
```

Azure DNS also supports 'empty' record sets, which can act as a placeholder to reserve a DNS name before creating DNS records. Empty record sets are visible in the Azure DNS control plane, but do appear on the Azure DNS name servers. The following example creates an empty record set:

```
New-AzDnsRecordSet -Name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -
Ttl 3600 -DnsRecords @()
```

Create records of other types

Having seen in detail how to create 'A' records, the following examples show how to create records of other record types supported by Azure DNS.

In each case, we show how to create a record set containing a single record. The earlier examples for 'A' records can be adapted to create record sets of other types containing multiple records, with metadata, or to create empty record sets.

We do not give an example to create an SOA record set, since SOAs are created and deleted with each DNS zone and cannot be created or deleted separately. However, the SOA can be modified, as shown in a later example.

Create an AAAA record set with a single record

```
New-AzDnsRecordSet -Name "test-aaaa" -RecordType AAAA -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Ipv6Address "2607:f8b0:4009:1803::1005")
```

Create a CAA record set with a single record

```
New-AzDnsRecordSet -Name "test-caa" -RecordType CAA -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Caaflags 0 -CaaTag "issue" -CaaValue
"ca1.contoso.com")
```

Create a CNAME record set with a single record

> **NOTE**
>
> The DNS standards do not permit CNAME records at the apex of a zone (`-Name '@'`), nor do they permit record sets containing more than one record.
>
> For more information, see CNAME records.

```
New-AzDnsRecordSet -Name "test-cname" -RecordType CNAME -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Cname "www.contoso.com")
```

Create an MX record set with a single record

In this example, we use the record set name '@' to create an MX record at the zone apex (in this case, 'contoso.com').

```
New-AzDnsRecordSet -Name "@" -RecordType MX -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -Ttl
3600 -DnsRecords (New-AzDnsRecordConfig -Exchange "mail.contoso.com" -Preference 5)
```

Create an NS record set with a single record

```
New-AzDnsRecordSet -Name "test-ns" -RecordType NS -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Nsdname "ns1.contoso.com")
```

Create a PTR record set with a single record

In this case, 'my-arpa-zone.com' represents the ARPA reverse lookup zone representing your IP range. Each PTR record set in this zone corresponds to an IP address within this IP range. The record name '10' is the last octet of the IP address within this IP range represented by this record.

```
New-AzDnsRecordSet -Name 10 -RecordType PTR -ZoneName "my-arpa-zone.com" -ResourceGroupName "MyResourceGroup"
-Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Ptrdname "myservice.contoso.com")
```

Create an SRV record set with a single record

When creating an SRV record set, specify the *_service* and *_protocol* in the record set name. There is no need to include '@' in the record set name when creating an SRV record set at the zone apex.

```
New-AzDnsRecordSet -Name "_sip._tls" -RecordType SRV -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Priority 0 -Weight 5 -Port 8080 -Target
"sip.contoso.com")
```

Create a TXT record set with a single record

The following example shows how to create a TXT record. For more information about the maximum string length supported in TXT records, see TXT records.

```
New-AzDnsRecordSet -Name "test-txt" -RecordType TXT -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup" -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Value "This is a TXT record")
```

Get a record set

To retrieve an existing record set, use `Get-AzDnsRecordSet`. This cmdlet returns a local object that represents the record set in Azure DNS.

As with `New-AzDnsRecordSet`, the record set name given must be a *relative* name, meaning it must exclude the zone name. You also need to specify the record type, and the zone containing the record set.

The following example shows how to retrieve a record set. In this example, the zone is specified using the `-ZoneName` and `-ResourceGroupName` parameters.

```
$rs = Get-AzDnsRecordSet -Name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup"
```

Alternatively, you can also specify the zone using a zone object, passed using the `-Zone` parameter.

```
$zone = Get-AzDnsZone -Name "contoso.com" -ResourceGroupName "MyResourceGroup"
$rs = Get-AzDnsRecordSet -Name "www" -RecordType A -Zone $zone
```

List record sets

You can also use `Get-AzDnsZone` to list record sets in a zone, by omitting the `-Name` and/or `-RecordType` parameters.

The following example returns all record sets in the zone:

```
$recordsets = Get-AzDnsRecordSet -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"
```

The following example shows how all record sets of a given type can be retrieved by specifying the record type while omitting the record set name:

```
$recordsets = Get-AzDnsRecordSet -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"
```

To retrieve all record sets with a given name, across record types, you need to retrieve all record sets and then filter the results:

```
$recordsets = Get-AzDnsRecordSet -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" | where
{$_.Name.Equals("www")}
```

In all the above examples, the zone can be specified either by using the `-ZoneName` and `-ResourceGroupName` parameters (as shown), or by specifying a zone object:

```
$zone = Get-AzDnsZone -Name "contoso.com" -ResourceGroupName "MyResourceGroup"
$recordsets = Get-AzDnsRecordSet -Zone $zone
```

Add a record to an existing record set

To add a record to an existing record set, follow the following three steps:

1. Get the existing record set

   ```
   $rs = Get-AzDnsRecordSet -Name www –ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -
   RecordType A
   ```

2. Add the new record to the local record set. This is an off-line operation.

   ```
   Add-AzDnsRecordConfig -RecordSet $rs -Ipv4Address "5.6.7.8"
   ```

3. Commit the change back to the Azure DNS service.

   ```
   Set-AzDnsRecordSet -RecordSet $rs
   ```

Using `Set-AzDnsRecordSet` *replaces* the existing record set in Azure DNS (and all records it contains) with the record set specified. Etag checks are used to ensure concurrent changes are not overwritten. You can use the optional `-Overwrite` switch to suppress these checks.

This sequence of operations can also be *piped*, meaning you pass the record set object by using the pipe rather than passing it as a parameter:

```
Get-AzDnsRecordSet -Name "www" –ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -RecordType A |
Add-AzDnsRecordConfig -Ipv4Address "5.6.7.8" | Set-AzDnsRecordSet
```

The examples above show how to add an 'A' record to an existing record set of type 'A'. A similar sequence of operations is used to add records to record sets of other types, substituting the `-Ipv4Address` parameter of

`Add-AzDnsRecordConfig` with other parameters specific to each record type. The parameters for each record type are the same as for the `New-AzDnsRecordConfig` cmdlet, as shown in Additional record type examples above.

Record sets of type 'CNAME' or 'SOA' cannot contain more than one record. This constraint arises from the DNS standards. It is not a limitation of Azure DNS.

Remove a record from an existing record set

The process to remove a record from a record set is similar to the process to add a record to an existing record set:

1. Get the existing record set

   ```
   $rs = Get-AzDnsRecordSet -Name www –ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -
   RecordType A
   ```

2. Remove the record from the local record set object. This is an off-line operation. The record that's being removed must be an exact match with an existing record across all parameters.

   ```
   Remove-AzDnsRecordConfig -RecordSet $rs -Ipv4Address "5.6.7.8"
   ```

3. Commit the change back to the Azure DNS service. Use the optional `-Overwrite` switch to suppress Etag checks for concurrent changes.

   ```
   Set-AzDnsRecordSet -RecordSet $Rs
   ```

Using the above sequence to remove the last record from a record set does not delete the record set, rather it leaves an empty record set. To remove a record set entirely, see Delete a record set.

Similarly to adding records to a record set, the sequence of operations to remove a record set can also be piped:

```
Get-AzDnsRecordSet -Name www –ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -RecordType A |
Remove-AzDnsRecordConfig -Ipv4Address "5.6.7.8" | Set-AzDnsRecordSet
```

Different record types are supported by passing the appropriate type-specific parameters to `Remove-AzDnsRecordSet`. The parameters for each record type are the same as for the `New-AzDnsRecordConfig` cmdlet, as shown in Additional record type examples above.

Modify an existing record set

The steps for modifying an existing record set are similar to the steps you take when adding or removing records from a record set:

1. Retrieve the existing record set by using `Get-AzDnsRecordSet`.
2. Modify the local record set object by:
   - Adding or removing records
   - Changing the parameters of existing records
   - Changing the record set metadata and time to live (TTL)
3. Commit your changes by using the `Set-AzDnsRecordSet` cmdlet. This *replaces* the existing record set in Azure DNS with the record set specified.

When using `Set-AzDnsRecordSet`, Etag checks are used to ensure concurrent changes are not overwritten. You can

use the optional `-Overwrite` switch to suppress these checks.

To update a record in an existing record set

In this example, we change the IP address of an existing 'A' record:

```
$rs = Get-AzDnsRecordSet -name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup"
$rs.Records[0].Ipv4Address = "9.8.7.6"
Set-AzDnsRecordSet -RecordSet $rs
```

To modify an SOA record

You cannot add or remove records from the automatically created SOA record set at the zone apex (`-Name "@"` , including quote marks). However, you can modify any of the parameters within the SOA record (except "Host") and the record set TTL.

The following example shows how to change the *Email* property of the SOA record:

```
$rs = Get-AzDnsRecordSet -Name "@" -RecordType SOA -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup"
$rs.Records[0].Email = "admin.contoso.com"
Set-AzDnsRecordSet -RecordSet $rs
```

To modify NS records at the zone apex

The NS record set at the zone apex is automatically created with each DNS zone. It contains the names of the Azure DNS name servers assigned to the zone.

You can add additional name servers to this NS record set, to support co-hosting domains with more than one DNS provider. You can also modify the TTL and metadata for this record set. However, you cannot remove or modify the pre-populated Azure DNS name servers.

Note that this applies only to the NS record set at the zone apex. Other NS record sets in your zone (as used to delegate child zones) can be modified without constraint.

The following example shows how to add an additional name server to the NS record set at the zone apex:

```
$rs = Get-AzDnsRecordSet -Name "@" -RecordType NS -ZoneName "contoso.com" -ResourceGroupName
"MyResourceGroup"
Add-AzDnsRecordConfig -RecordSet $rs -Nsdname ns1.myotherdnsprovider.com
Set-AzDnsRecordSet -RecordSet $rs
```

To modify record set metadata

Record set metadata can be used to associate application-specific data with each record set, as key-value pairs.

The following example shows how to modify the metadata of an existing record set:

```
# Get the record set
$rs = Get-AzDnsRecordSet -Name www -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"

# Add 'dept=finance' name-value pair
$rs.Metadata.Add('dept', 'finance')

# Remove metadata item named 'environment'
$rs.Metadata.Remove('environment')

# Commit changes
Set-AzDnsRecordSet -RecordSet $rs
```

Delete a record set

Record sets can be deleted by using the `Remove-AzDnsRecordSet` cmdlet. Deleting a record set also deletes all records within the record set.

> **NOTE**
> You cannot delete the SOA and NS record sets at the zone apex (`-Name '@'`). Azure DNS created these automatically when the zone was created, and deletes them automatically when the zone is deleted.

The following example shows how to delete a record set. In this example, the record set name, record set type, zone name, and resource group are each specified explicitly.

```
Remove-AzDnsRecordSet -Name "www" -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"
```

Alternatively, the record set can be specified by name and type, and the zone specified using an object:

```
$zone = Get-AzDnsZone -Name "contoso.com" -ResourceGroupName "MyResourceGroup"
Remove-AzDnsRecordSet -Name "www" -RecordType A -Zone $zone
```

As a third option, the record set itself can be specified using a record set object:

```
$rs = Get-AzDnsRecordSet -Name www -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"
Remove-AzDnsRecordSet -RecordSet $rs
```

When you specify the record set to be deleted by using a record set object, Etag checks are used to ensure concurrent changes are not deleted. You can use the optional `-Overwrite` switch to suppress these checks.

The record set object can also be piped instead of being passed as a parameter:

```
Get-AzDnsRecordSet -Name www -RecordType A -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" |
Remove-AzDnsRecordSet
```

Confirmation prompts

The `New-AzDnsRecordSet` , `Set-AzDnsRecordSet` , and `Remove-AzDnsRecordSet` cmdlets all support confirmation prompts.

Each cmdlet prompts for confirmation if the `$ConfirmPreference` PowerShell preference variable has a value of `Medium` or lower. Since the default value for `$ConfirmPreference` is `High` , these prompts are not given when using the default PowerShell settings.

You can override the current `$ConfirmPreference` setting using the `-Confirm` parameter. If you specify `-Confirm` or `-Confirm:$True` , the cmdlet prompts you for confirmation before it runs. If you specify `-Confirm:$False` , the cmdlet does not prompt you for confirmation.

For more information about `-Confirm` and `$ConfirmPreference` , see About Preference Variables.

Next steps

Learn more about zones and records in Azure DNS.
Learn how to protect your zones and records when using Azure DNS.

Review the Azure DNS PowerShell reference documentation.

Manage DNS records and recordsets in Azure DNS using the Azure CLI

2/1/2020 • 13 minutes to read • Edit Online

This article shows you how to manage DNS records for your DNS zone by using the cross-platform Azure CLI, which is available for Windows, Mac and Linux. You can also manage your DNS records using Azure PowerShell or the Azure portal.

The examples in this article assume you have already installed the Azure CLI, signed in, and created a DNS zone.

Introduction

Before creating DNS records in Azure DNS, you first need to understand how Azure DNS organizes DNS records into DNS record sets.

Record names

In Azure DNS, records are specified by using relative names. A *fully qualified* domain name (FQDN) includes the zone name, whereas a *relative* name does not. For example, the relative record name `www` in the zone `contoso.com` gives the fully qualified record name `www.contoso.com`.

An *apex* record is a DNS record at the root (or *apex*) of a DNS zone. For example, in the DNS zone `contoso.com`, an apex record also has the fully qualified name `contoso.com` (this is sometimes called a *naked* domain). By convention, the relative name '@' is used to represent apex records.

Record types

Each DNS record has a name and a type. Records are organized into various types according to the data they contain. The most common type is an 'A' record, which maps a name to an IPv4 address. Another common type is an 'MX' record, which maps a name to a mail server.

Azure DNS supports all common DNS record types: A, AAAA, CAA, CNAME, MX, NS, PTR, SOA, SRV, and TXT. Note that SPF records are represented using TXT records.

Record sets

Sometimes you need to create more than one DNS record with a given name and type. For example, suppose the 'www.contoso.com' web site is hosted on two different IP addresses. The website requires two different A records, one for each IP address. Here is an example of a record set:

```
www.contoso.com.        3600    IN    A    134.170.185.46
www.contoso.com.        3600    IN    A    134.170.188.221
```

Azure DNS manages all DNS records using *record sets*. A record set (also known as a *resource* record set) is the collection of DNS records in a zone that have the same name and are of the same type. Most record sets contain a single record. However, examples like the one above, in which a record set contains more than one record, are not uncommon.

For example, suppose you have already created an A record 'www' in the zone 'contoso.com', pointing to the IP address '134.170.185.46' (the first record above). To create the second record you would add that record to the existing record set, rather than create an additional record set.

The SOA and CNAME record types are exceptions. The DNS standards don't permit multiple records with the

same name for these types, therefore these record sets can only contain a single record.

For more information about DNS records in Azure DNS, see DNS zones and records.

Create a DNS record

To create a DNS record, use the `az network dns record-set <record-type> add-record` command (where `<record-type>` is the type of record, i.e a, srv, txt, etc.) For help, see `az network dns record-set --help`.

When creating a record, you need to specify the resource group name, zone name, record set name, the record type, and the details of the record being created. The record set name given must be a *relative* name, meaning it must exclude the zone name.

If the record set does not already exist, this command creates it for you. If the record set already exists, this command adds the record you specify to the existing record set.

If a new record set is created, a default time-to-live (TTL) of 3600 is used. For instructions on how to use different TTLs, see Create a DNS record set.

The following example creates an A record called *www* in the zone *contoso.com* in the resource group *MyResourceGroup*. The IP address of the A record is *1.2.3.4*.

```
az network dns record-set a add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name www --ipv4-address 1.2.3.4
```

To create a record set in the apex of the zone (in this case, "contoso.com"), use the record name "@", including the quotation marks:

```
az network dns record-set a add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name "@" --ipv4-address 1.2.3.4
```

Create a DNS record set

In the above examples, the DNS record was either added to an existing record set, or the record set was created *implicitly*. You can also create the record set *explicitly* before adding records to it. Azure DNS supports 'empty' record sets, which can act as a placeholder to reserve a DNS name before creating DNS records. Empty record sets are visible in the Azure DNS control plane, but do not appear on the Azure DNS name servers.

Record sets are created using the `az network dns record-set <record-type> create` command. For help, see `az network dns record-set <record-type> create --help`.

Creating the record set explicitly allows you to specify record set properties such as the Time-To-Live (TTL) and metadata. Record set metadata can be used to associate application-specific data with each record set, as key-value pairs.

The following example creates an empty record set of type 'A' with a 60-second TTL, by using the `--ttl` parameter (short form `-l`):

```
az network dns record-set a create --resource-group myresourcegroup --zone-name contoso.com --name www --ttl 60
```

The following example creates a record set with two metadata entries, "dept=finance" and "environment=production", by using the `--metadata` parameter :

```
az network dns record-set a create --resource-group myresourcegroup --zone-name contoso.com --name www --
metadata "dept=finance" "environment=production"
```

Having created an empty record set, records can be added using `azure network dns record-set <record-type> add-record` as described in Create a DNS record.

Create records of other types

Having seen in detail how to create 'A' records, the following examples show how to create record of other record types supported by Azure DNS.

The parameters used to specify the record data vary depending on the type of the record. For example, for a record of type "A", you specify the IPv4 address with the parameter `--ipv4-address <IPv4 address>`. The parameters for each record type can be listed using `az network dns record-set <record-type> add-record --help`.

In each case, we show how to create a single record. The record is added to the existing record set, or a record set created implicitly. For more information on creating record sets and defining record set parameter explicitly, see Create a DNS record set.

We do not give an example to create an SOA record set, since SOAs are created and deleted with each DNS zone and cannot be created or deleted separately. However, the SOA can be modified, as shown in a later example.

Create an AAAA record

```
az network dns record-set aaaa add-record --resource-group myresourcegroup --zone-name contoso.com --record-
set-name test-aaaa --ipv6-address 2607:f8b0:4009:1803::1005
```

Create an CAA record

```
az network dns record-set caa add-record --resource-group myresourcegroup --zone-name contoso.com --record-
set-name test-caa --flags 0 --tag "issue" --value "ca1.contoso.com"
```

Create a CNAME record

> **NOTE**
>
> The DNS standards do not permit CNAME records at the apex of a zone (`--Name "@"`), nor do they permit record sets containing more than one record.
>
> For more information, see CNAME records.

```
az network dns record-set cname set-record --resource-group myresourcegroup --zone-name contoso.com --record-
set-name test-cname --cname www.contoso.com
```

Create an MX record

In this example, we use the record set name "@" to create the MX record at the zone apex (in this case, "contoso.com").

```
az network dns record-set mx add-record --resource-group myresourcegroup --zone-name contoso.com --record-
set-name "@" --exchange mail.contoso.com --preference 5
```

Create an NS record

```
az network dns record-set ns add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name test-ns --nsdname ns1.contoso.com
```

Create a PTR record

In this case, 'my-arpa-zone.com' represents the ARPA zone representing your IP range. Each PTR record set in this zone corresponds to an IP address within this IP range. The record name '10' is the last octet of the IP address within this IP range represented by this record.

```
az network dns record-set ptr add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name my-arpa.zone.com --ptrdname myservice.contoso.com
```

Create an SRV record

When creating an SRV record set, specify the *_service* and *_protocol* in the record set name. There is no need to include "@" in the record set name when creating an SRV record set at the zone apex.

```
az network dns record-set srv add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name _sip._tls --priority 10 --weight 5 --port 8080 --target sip.contoso.com
```

Create a TXT record

The following example shows how to create a TXT record. For more information about the maximum string length supported in TXT records, see TXT records.

```
az network dns record-set txt add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name test-txt --value "This is a TXT record"
```

Get a record set

To retrieve an existing record set, use `az network dns record-set <record-type> show`. For help, see `az network dns record-set <record-type> show --help`.

As when creating a record or record set, the record set name given must be a *relative* name, meaning it must exclude the zone name. You also need to specify the record type, the zone containing the record set, and the resource group containing the zone.

The following example retrieves the record *www* of type A from zone *contoso.com* in resource group *MyResourceGroup*:

```
az network dns record-set a show --resource-group myresourcegroup --zone-name contoso.com --name www
```

List record sets

You can list all records in a DNS zone by using the `az network dns record-set list` command. For help, see `az network dns record-set list --help`.

This example returns all record sets in the zone *contoso.com*, in resource group *MyResourceGroup*, regardless of name or record type:

```
az network dns record-set list --resource-group myresourcegroup --zone-name contoso.com
```

This example returns all record sets that match the given record type (in this case, 'A' records):

```
az network dns record-set a list --resource-group myresourcegroup --zone-name contoso.com
```

Add a record to an existing record set

You can use `az network dns record-set <record-type> add-record` both to create a record in a new record set, or to add a record to an existing record set.

For more information, see Create a DNS record and Create records of other types above.

Remove a record from an existing record set.

To remove a DNS record from an existing record set, use `az network dns record-set <record-type> remove-record`. For help, see `az network dns record-set <record-type> remove-record -h`.

This command deletes a DNS record from a record set. If the last record in a record set is deleted, the record set itself is also deleted. To keep the empty record set instead, use the `--keep-empty-record-set` option.

You need to specify the record to be deleted and the zone it should be deleted from, using the same parameters as when creating a record using `az network dns record-set <record-type> add-record`. These parameters are described in Create a DNS record and Create records of other types above.

The following example deletes the A record with value '1.2.3.4' from the record set named *www* in the zone *contoso.com*, in the resource group *MyResourceGroup*.

```
az network dns record-set a remove-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name "www" --ipv4-address 1.2.3.4
```

Modify an existing record set

Each record set contains a time-to-live (TTL), metadata, and DNS records. The following sections explain how to modify each of these properties.

To modify an A, AAAA, CAA, MX, NS, PTR, SRV, or TXT record

To modify an existing record of type A, AAAA, CAA, MX, NS, PTR, SRV, or TXT, you should first add a new record and then delete the existing record. For detailed instructions on how to delete and add records, see the earlier sections of this article.

The following example shows how to modify an 'A' record, from IP address 1.2.3.4 to IP address 5.6.7.8:

```
az network dns record-set a add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name www --ipv4-address 5.6.7.8
az network dns record-set a remove-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name www --ipv4-address 1.2.3.4
```

You cannot add, remove, or modify the records in the automatically created NS record set at the zone apex (`--Name "@"`, including quote marks). For this record set, the only changes permitted are to modify the record set TTL and metadata.

To modify a CNAME record

Unlike most other record types, a CNAME record set can only contain a single record. Therefore, you cannot replace the current value by adding a new record and removing the existing record, as for other record types.

Instead, to modify a CNAME record, use `az network dns record-set cname set-record`. For help, see `az network dns record-set cname set-record --help`

The example modifies the CNAME record set *www* in the zone *contoso.com*, in resource group *MyResourceGroup*, to point to 'www.fabrikam.net' instead of its existing value:

```
az network dns record-set cname set-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name test-cname --cname www.fabrikam.net
```

To modify an SOA record

Unlike most other record types, a CNAME record set can only contain a single record. Therefore, you cannot replace the current value by adding a new record and removing the existing record, as for other record types.

Instead, to modify the SOA record, use `az network dns record-set soa update`. For help, see `az network dns record-set soa update --help`.

The following example shows how to set the 'email' property of the SOA record for the zone *contoso.com* in the resource group *MyResourceGroup*:

```
az network dns record-set soa update --resource-group myresourcegroup --zone-name contoso.com --email admin.contoso.com
```

To modify NS records at the zone apex

The NS record set at the zone apex is automatically created with each DNS zone. It contains the names of the Azure DNS name servers assigned to the zone.

You can add additional name servers to this NS record set, to support co-hosting domains with more than one DNS provider. You can also modify the TTL and metadata for this record set. However, you cannot remove or modify the pre-populated Azure DNS name servers.

Note that this applies only to the NS record set at the zone apex. Other NS record sets in your zone (as used to delegate child zones) can be modified without constraint.

The following example shows how to add an additional name server to the NS record set at the zone apex:

```
az network dns record-set ns add-record --resource-group myresourcegroup --zone-name contoso.com --record-set-name "@" --nsdname ns1.myotherdnsprovider.com
```

To modify the TTL of an existing record set

To modify the TTL of an existing record set, use `azure network dns record-set <record-type> update`. For help, see `azure network dns record-set <record-type> update --help`.

The following example shows how to modify a record set TTL, in this case to 60 seconds:

```
az network dns record-set a update --resource-group myresourcegroup --zone-name contoso.com --name www --set ttl=60
```

To modify the metadata of an existing record set

Record set metadata can be used to associate application-specific data with each record set, as key-value pairs. To modify the metadata of an existing record set, use `az network dns record-set <record-type> update`. For help, see `az network dns record-set <record-type> update --help`.

The following example shows how to modify a record set with two metadata entries, "dept=finance" and "environment=production". Note that any existing metadata is *replaced* by the values given.

```
az network dns record-set a update --resource-group myresourcegroup --zone-name contoso.com --name www --set
metadata.dept=finance metadata.environment=production
```

Delete a record set

Record sets can be deleted by using the `az network dns record-set <record-type> delete` command. For help, see `azure network dns record-set <record-type> delete --help`. Deleting a record set also deletes all records within the record set.

> **NOTE**
> You cannot delete the SOA and NS record sets at the zone apex (`--name "@"`). These are created automatically when the zone was created, and are deleted automatically when the zone is deleted.

The following example deletes the record set named *www* of type A from the zone *contoso.com* in resource group *MyResourceGroup*:

```
az network dns record-set a delete --resource-group myresourcegroup --zone-name contoso.com --name www
```

You are prompted to confirm the delete operation. To suppress this prompt, use the `--yes` switch.

Next steps

Learn more about zones and records in Azure DNS.
Learn how to protect your zones and records when using Azure DNS.

Host reverse DNS lookup zones in Azure DNS

2/1/2020 • 7 minutes to read • Edit Online

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.

This article explains how to host the reverse DNS lookup zones for your assigned IP ranges in Azure DNS. The IP ranges represented by the reverse lookup zones must be assigned to your organization, typically by your ISP.

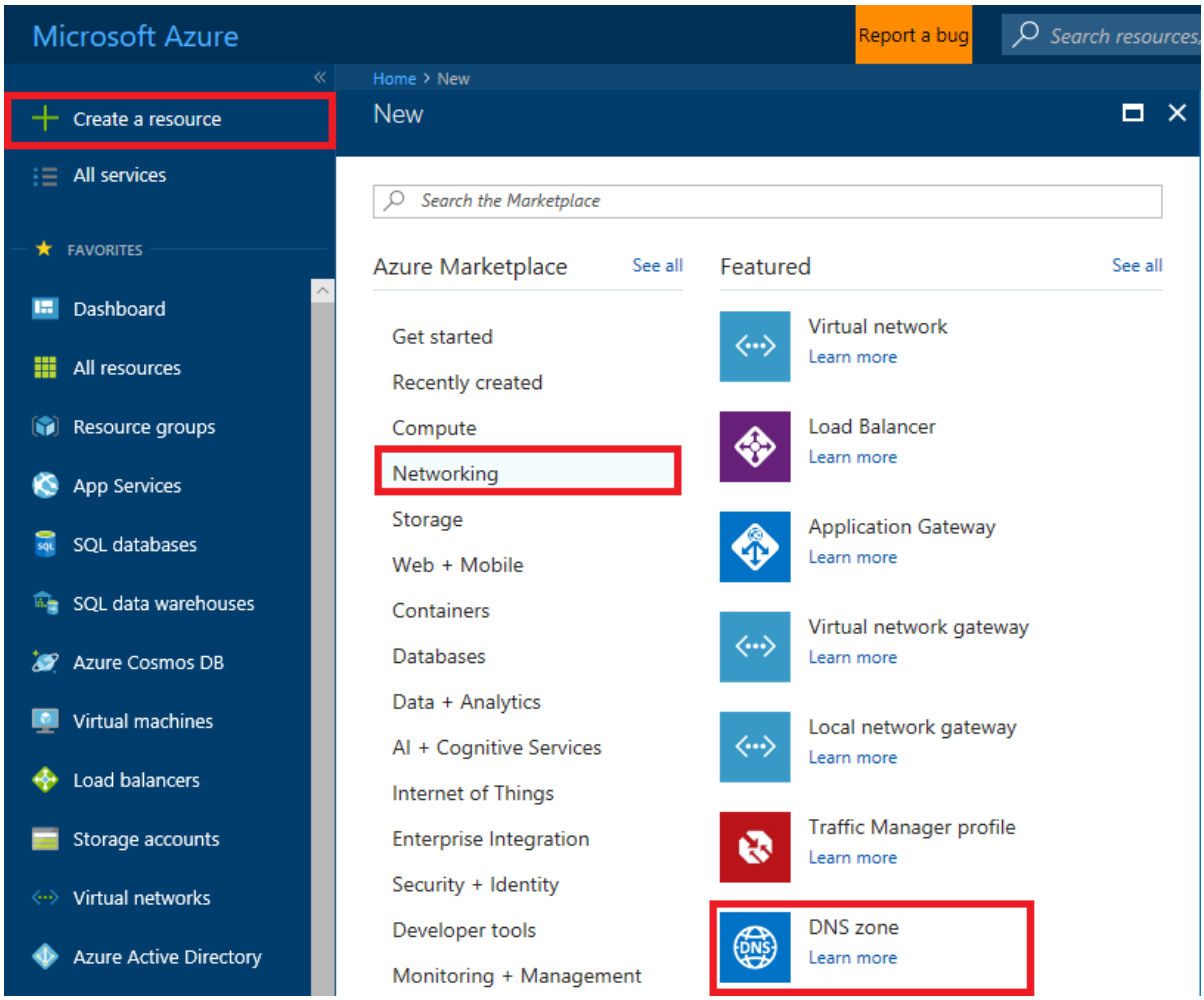
To configure reverse DNS for an Azure-owned IP address that's assigned to your Azure service, see Configure reverse DNS for services hosted in Azure.

Before you read this article, you should be familiar with the overview of reverse DNS and support in Azure.

This article walks you through the steps to create your first reverse lookup DNS zone and record by using the Azure portal, Azure PowerShell, Azure classic CLI, or Azure CLI.

Create a reverse lookup DNS zone

1. Sign in to the Azure portal.
2. On the **Hub** menu, select **New** > **Networking**, and then select **DNS zone**.



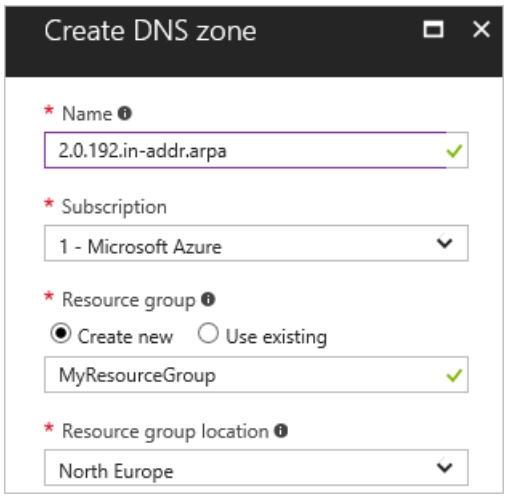
3. In the **Create DNS zone** pane, name your DNS zone. The name of the zone is crafted differently for IPv4 and IPv6 prefixes. Use the instructions for IPv4 or IPv6 to name your zone. When you're finished, select **Create** to create the zone.

IPv4

The name of an IPv4 reverse lookup zone is based on the IP range that it represents. It should be in the following format: `<IPv4 network prefix in reverse order>.in-addr.arpa`. For examples, see Overview of reverse DNS and support in Azure.

> **NOTE**
>
> When you're creating classless reverse DNS lookup zones in Azure DNS, you must use a hyphen (`-`) rather than a forward slash (`/`) in the zone name.
>
> For example, for the IP range 192.0.2.128/26, you must use `128-26.2.0.192.in-addr.arpa` as the zone name instead of `128/26.2.0.192.in-addr.arpa`.
>
> Although the DNS standards support both methods, Azure DNS doesn't support DNS zone names that contain for forward slash (`/`) character.

The following example shows how to create a Class C reverse DNS zone named `2.0.192.in-addr.arpa` in Azure DNS via the Azure portal:



Resource group location defines the location for the resource group. It has no impact on the DNS zone. The DNS zone location is always "global," and is not shown.

The following examples show how to complete this task by using Azure PowerShell and Azure CLI.

PowerShell

```
New-AzDnsZone -Name 2.0.192.in-addr.arpa -ResourceGroupName MyResourceGroup
```

Azure classic CLI

```
azure network dns zone create MyResourceGroup 2.0.192.in-addr.arpa
```

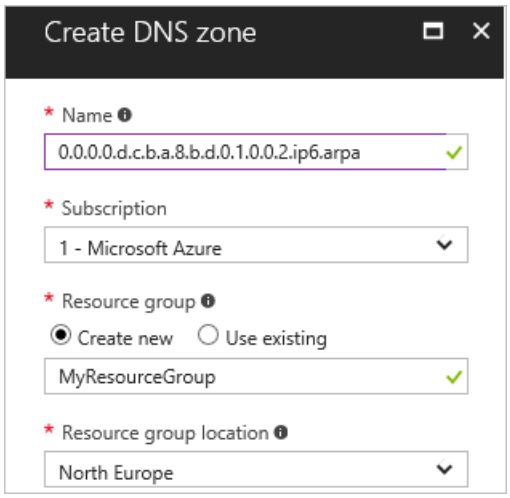
Azure CLI

```
az network dns zone create -g MyResourceGroup -n 2.0.192.in-addr.arpa
```

IPv6

The name of an IPv6 reverse lookup zone should be in the following form: `<IPv6 network prefix in reverse order>.ip6.arpa`. For examples, see Overview of reverse DNS and support in Azure.

The following example shows how to create an IPv6 reverse DNS lookup zone named `0.0.0.0.d.c.b.a.8.b.d.0.1.0.0.2.ip6.arpa` in Azure DNS via the Azure portal:



Resource group location defines the location for the resource group. It has no impact on the DNS zone. The DNS zone location is always "global," and is not shown.

The following examples show how to complete this task by using Azure PowerShell and Azure CLI.

PowerShell

```
New-AzDnsZone -Name 0.0.0.0.d.c.b.a.8.b.d.0.1.0.0.2.ip6.arpa -ResourceGroupName MyResourceGroup
```

Azure classic CLI

```
azure network dns zone create MyResourceGroup 0.0.0.0.d.c.b.a.8.b.d.0.1.0.0.2.ip6.arpa
```

Azure CLI

```
az network dns zone create -g MyResourceGroup -n 0.0.0.0.d.c.b.a.8.b.d.0.1.0.0.2.ip6.arpa
```

Delegate a reverse DNS lookup zone

Now that you've created your reverse DNS lookup zone, you must ensure that the zone is delegated from the parent zone. DNS delegation enables the DNS name resolution process to find the name servers that host your reverse DNS lookup zone. Those name servers can then answer DNS reverse queries for the IP addresses in your address range.

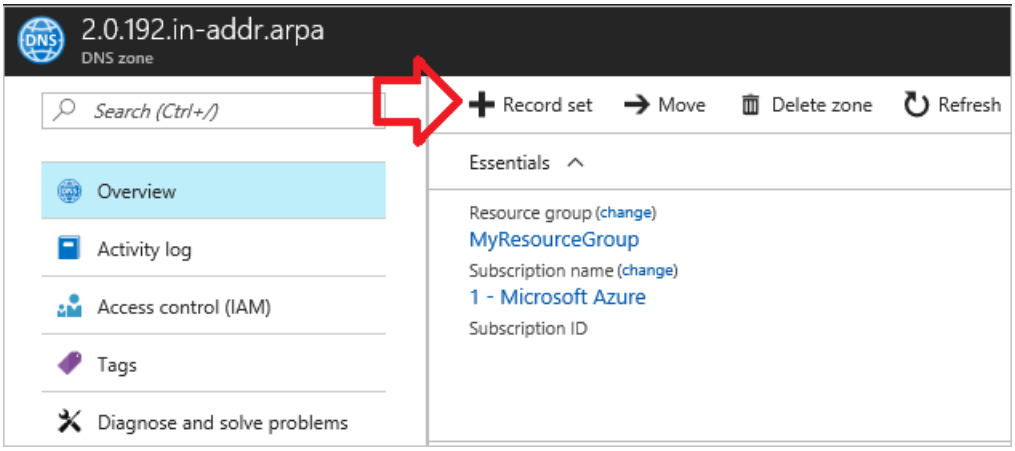
For forward lookup zones, the process of delegating a DNS zone is described in Delegate your domain to Azure DNS. Delegation for reverse lookup zones works the same way. The only difference is that you need to configure the name servers with the ISP that provided your IP range, rather than your domain name registrar.

Create a DNS PTR record

IPv4

The following example walks you through the process of creating a PTR record in a reverse DNS zone in Azure DNS. For other record types and to modify existing records, see Manage DNS records and record sets by using the Azure portal.

1. At the top of the **DNS zone** pane, select **+ Record set** to open the **Add record set** pane.



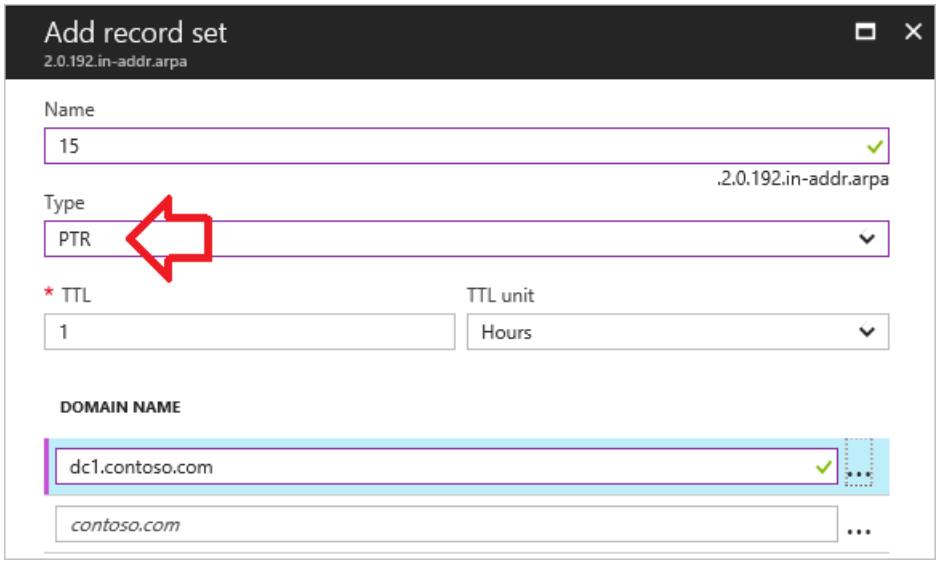
2. The name of the record set for a PTR record needs to be the rest of the IPv4 address in reverse order.

   In this example, the first three octets are already populated as part of the zone name (.2.0.192). Therefore, only the last octet is supplied in the **Name** box. For example, you might name your record set **15** for a resource whose IP address is 192.0.2.15.

3. For **Type**, select **PTR**.

4. For **DOMAIN NAME**, enter the fully qualified domain name (FQDN) of the resource that uses the IP.

5. Select **OK** at the bottom of the pane to create the DNS record.



The following examples show how to complete this task by using PowerShell or Azure CLI.

PowerShell

```
New-AzDnsRecordSet -Name 15 -RecordType PTR -ZoneName 2.0.192.in-addr.arpa -ResourceGroupName MyResourceGroup
-Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -Ptrdname "dc1.contoso.com")
```

Azure classic CLI

```
azure network dns record-set add-record MyResourceGroup 2.0.192.in-addr.arpa 15 PTR --ptrdname dc1.contoso.com
```

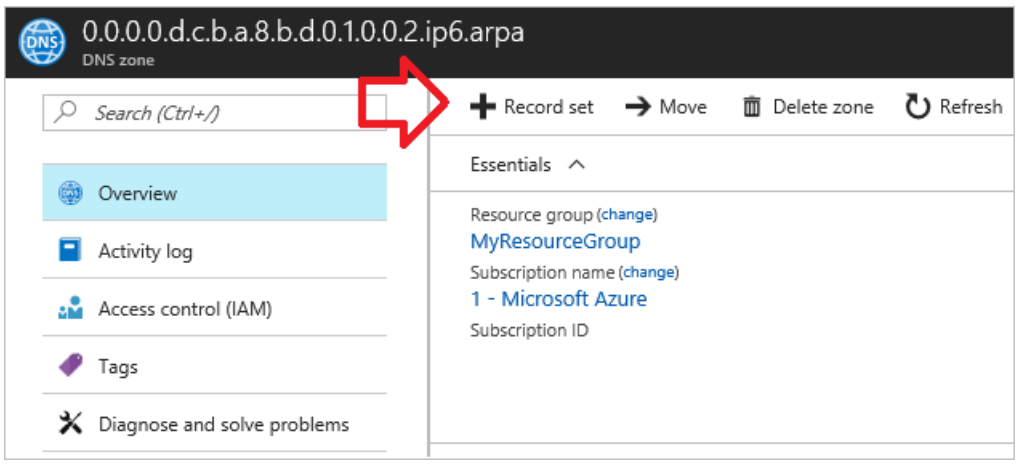
Azure CLI

```
 az network dns record-set ptr add-record -g MyResourceGroup -z 2.0.192.in-addr.arpa -n 15 --ptrdname
dc1.contoso.com
```

IPv6

The following example walks you through the process of creating new PTR record. For other record types and to modify existing records, see Manage DNS records and record sets by using the Azure portal.

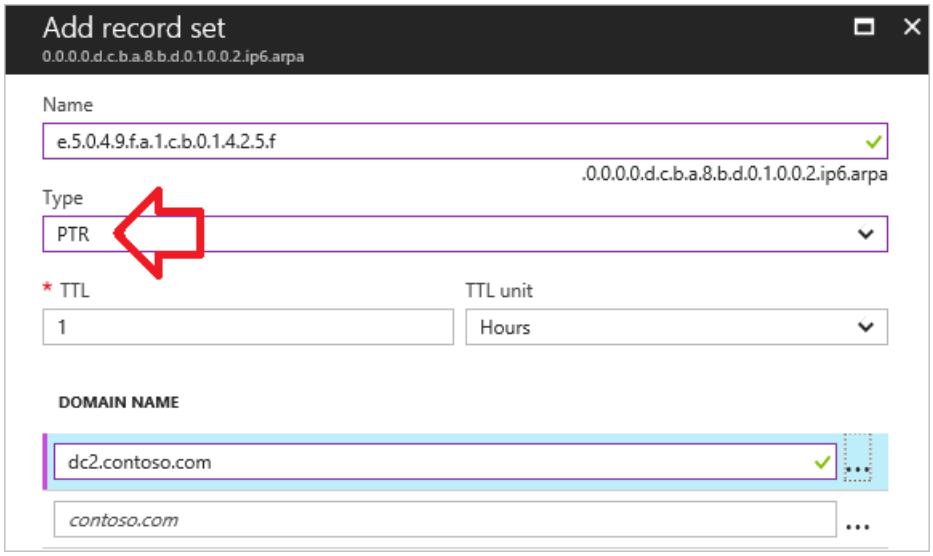
1. At the top of the **DNS zone** pane, select **+ Record set** to open the **Add record set** pane.



2. The name of the record set for a PTR record needs to be the rest of the IPv6 address in reverse order. It must not include any zero compression.

In this example, the first 64 bits of the IPv6 are already populated as part of the zone name (0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa). Therefore, only the last 64 bits are supplied in the **Name** box. The last 64 bits of the IP address are entered in reverse order, with a period as the delimiter between each hexadecimal number. For example, you might name your record set **e.5.0.4.9.f.a.1.c.b.0.1.4.2.5.f** for a resource whose IP address is 2001:0db8:abdc:0000:f524:10bc:1af9:405e.

3. For **Type**, select **PTR**.
4. For **DOMAIN NAME**, enter the FQDN of the resource that uses the IP.
5. Select **OK** at the bottom of the pane to create the DNS record.



The following examples show how to complete this task by using PowerShell or Azure CLI.

PowerShell

```
New-AzDnsRecordSet -Name "e.5.0.4.9.f.a.1.c.b.0.1.4.2.5.f" -RecordType PTR -ZoneName
0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa -ResourceGroupName MyResourceGroup -Ttl 3600 -DnsRecords (New-
AzDnsRecordConfig -Ptrdname "dc2.contoso.com")
```

Azure classic CLI

```
azure network dns record-set add-record MyResourceGroup 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa
e.5.0.4.9.f.a.1.c.b.0.1.4.2.5.f PTR --ptrdname dc2.contoso.com
```

Azure CLI

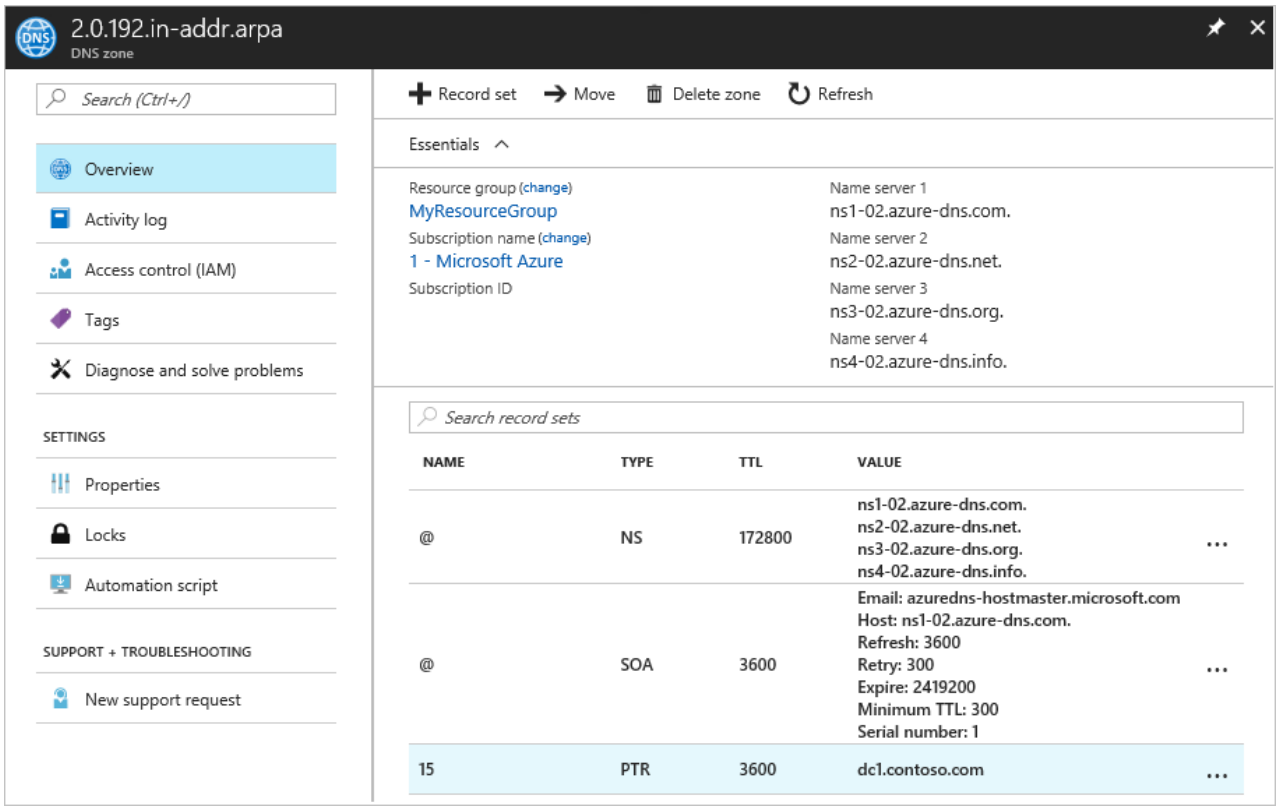
```
 az network dns record-set ptr add-record -g MyResourceGroup -z 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa -n
e.5.0.4.9.f.a.1.c.b.0.1.4.2.5.f --ptrdname dc2.contoso.com
```

View records

To view the records that you created, browse to your DNS zone in the Azure portal. In the lower part of the **DNS zone** pane, you can see the records for the DNS zone. You should see the default NS and SOA records, plus any new records that you've created. The NS and SOA records are created in every zone.

IPv4

The **DNS zone** pane shows the IPv4 PTR records:



The following examples show how to view the PTR records by using PowerShell or Azure CLI.

PowerShell

```
Get-AzDnsRecordSet -ZoneName 2.0.192.in-addr.arpa -ResourceGroupName MyResourceGroup
```

Azure classic CLI

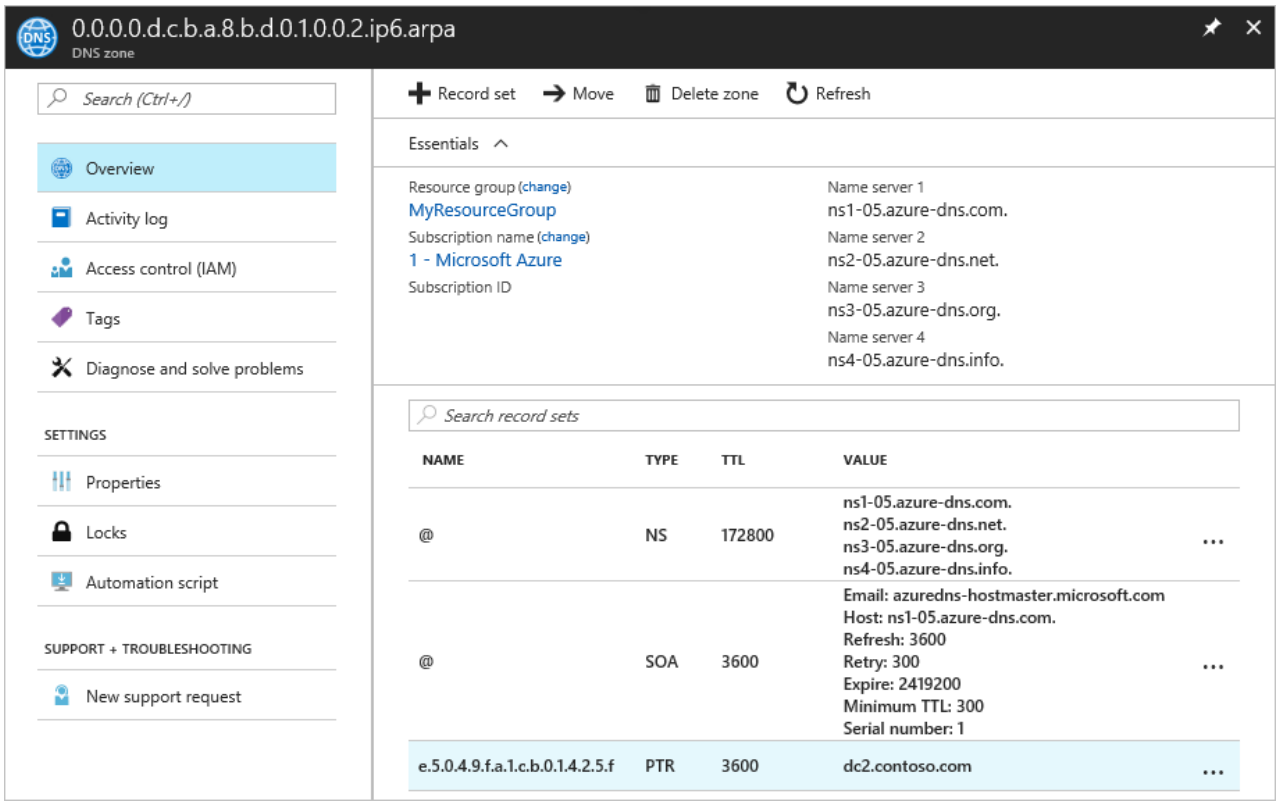
```
azure network dns record-set list MyResourceGroup 2.0.192.in-addr.arpa
```

Azure CLI

```
azure network dns record-set list -g MyResourceGroup -z 2.0.192.in-addr.arpa
```

IPv6

The **DNS zone** pane shows the IPv6 PTR records:



The following examples show how to view the records by using PowerShell or Azure CLI.

PowerShell

```
Get-AzDnsRecordSet -ZoneName 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa -ResourceGroupName MyResourceGroup
```

Azure classic CLI

```
azure network dns record-set list MyResourceGroup 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa
```

Azure CLI

```
azure network dns record-set list -g MyResourceGroup -z 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa
```

FAQ

Can I host reverse DNS lookup zones for my ISP-assigned IP blocks on Azure DNS?

Yes. Hosting the reverse lookup (ARPA) zones for your own IP ranges in Azure DNS is fully supported.

Create the reverse lookup zone in Azure DNS as explained in this article, and then work with your ISP to delegate the zone. You can then manage the PTR records for each reverse lookup in the same way as other record types.

How much does hosting my reverse DNS lookup zone cost?

Hosting the reverse DNS lookup zone for your ISP-assigned IP block in Azure DNS is charged at standard Azure DNS rates.

Can I host reverse DNS lookup zones for both IPv4 and IPv6 addresses in Azure DNS?

Yes. This article explains how to create both IPv4 and IPv6 reverse DNS lookup zones in Azure DNS.

Can I import an existing reverse DNS lookup zone?

Yes. You can use Azure CLI to import existing DNS zones into Azure DNS. This method works for both forward lookup zones and reverse lookup zones.

For more information, see Import and export a DNS zone file using Azure CLI.

Next steps

For more information on reverse DNS, see reverse DNS lookup on Wikipedia.
Learn how to manage reverse DNS records for your Azure services.

Configure reverse DNS for services hosted in Azure

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

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.

This article explains how to configure reverse DNS lookups for services hosted in Azure.

Services in Azure use IP addresses assigned by Azure and owned by Microsoft. These reverse DNS records (PTR records) must be created in the corresponding Microsoft-owned reverse DNS lookup zones. This article explains how to do this.

This scenario should not be confused with the ability to host the reverse DNS lookup zones for your assigned IP ranges in Azure DNS. In this case, the IP ranges represented by the reverse lookup zone must be assigned to your organization, typically by your ISP.

Before reading this article, you should be familiar with this Overview of reverse DNS and support in Azure.

In Azure DNS, compute resources (such as virtual machines, virtual machine scale sets, or Service Fabric clusters) are exposed via a PublicIpAddress resource. Reverse DNS lookups are configured using the 'ReverseFqdn' property of the PublicIpAddress.

Reverse DNS is not currently supported for the Azure App Service.

Validation of reverse DNS records

A third party should not be able to create reverse DNS records for their Azure service mapping to your DNS domains. To prevent this, Azure only allows the creation of a reverse DNS record where domain name specified in the reverse DNS record is the same as, or resolves to, the DNS name or IP address of a PublicIpAddress or Cloud Service in the same Azure subscription.

This validation is only performed when the reverse DNS record is set or modified. Periodic re-validation is not performed.

For example: suppose the PublicIpAddress resource has the DNS name contosoapp1.northus.cloudapp.azure.com and IP address 23.96.52.53. The ReverseFqdn for the PublicIpAddress can be specified as:

- The DNS name for the PublicIpAddress, contosoapp1.northus.cloudapp.azure.com
- The DNS name for a different PublicIpAddress in the same subscription, such as contosoapp2.westus.cloudapp.azure.com
- A vanity DNS name, such as app1.contoso.com, so long as this name is *first* configured as a CNAME to contosoapp1.northus.cloudapp.azure.com, or to a different PublicIpAddress in the same subscription.
- A vanity DNS name, such as app1.contoso.com, so long as this name is *first* configured as an A record to the IP address 23.96.52.53, or to the IP address of a different PublicIpAddress in the same subscription.

The same constraints apply to reverse DNS for Cloud Services.

Reverse DNS for PublicIpAddress resources

This section provides detailed instructions for how to configure reverse DNS for PublicIpAddress resources in the Resource Manager deployment model, using either Azure PowerShell, Azure classic CLI, or Azure CLI. Configuring reverse DNS for PublicIpAddress resources is not currently supported via the Azure portal.

Azure currently supports reverse DNS only for IPv4 PublicIpAddress resources. It is not supported for IPv6.

Add reverse DNS to an existing PublicIpAddresses

PowerShell

To update reverse DNS to an existing PublicIpAddress:

```
$pip = Get-AzPublicIpAddress -Name "PublicIp" -ResourceGroupName "MyResourceGroup"
$pip.DnsSettings.ReverseFqdn = "contosoapp1.westus.cloudapp.azure.com."
Set-AzPublicIpAddress -PublicIpAddress $pip
```

To add reverse DNS to an existing PublicIpAddress that doesn't already have a DNS name, you must also specify a DNS name:

```
$pip = Get-AzPublicIpAddress -Name "PublicIp" -ResourceGroupName "MyResourceGroup"
$pip.DnsSettings = New-Object -TypeName
"Microsoft.Azure.Commands.Network.Models.PSPublicIpAddressDnsSettings"
$pip.DnsSettings.DomainNameLabel = "contosoapp1"
$pip.DnsSettings.ReverseFqdn = "contosoapp1.westus.cloudapp.azure.com."
Set-AzPublicIpAddress -PublicIpAddress $pip
```

Azure classic CLI

To add reverse DNS to an existing PublicIpAddress:

```
azure network public-ip set -n PublicIp -g MyResourceGroup -f contosoapp1.westus.cloudapp.azure.com.
```

To add reverse DNS to an existing PublicIpAddress that doesn't already have a DNS name, you must also specify a DNS name:

```
azure network public-ip set -n PublicIp -g MyResourceGroup -d contosoapp1 -f
contosoapp1.westus.cloudapp.azure.com.
```

Azure CLI

To add reverse DNS to an existing PublicIpAddress:

```
az network public-ip update --resource-group MyResourceGroup --name PublicIp --reverse-fqdn
contosoapp1.westus.cloudapp.azure.com.
```

To add reverse DNS to an existing PublicIpAddress that doesn't already have a DNS name, you must also specify a DNS name:

```
az network public-ip update --resource-group MyResourceGroup --name PublicIp --reverse-fqdn
contosoapp1.westus.cloudapp.azure.com --dns-name contosoapp1
```

Create a Public IP Address with reverse DNS

To create a new PublicIpAddress with the reverse DNS property already specified:

PowerShell

```
New-AzPublicIpAddress -Name "PublicIp" -ResourceGroupName "MyResourceGroup" -Location "WestUS" -
AllocationMethod Dynamic -DomainNameLabel "contosoapp2" -ReverseFqdn "contosoapp2.westus.cloudapp.azure.com."
```

Azure classic CLI

```
azure network public-ip create -n PublicIp -g MyResourceGroup -l westus -d contosoapp3 -f
contosoapp3.westus.cloudapp.azure.com.
```

Azure CLI

```
az network public-ip create --name PublicIp --resource-group MyResourceGroup --location westcentralus --dns-
name contosoapp1 --reverse-fqdn contosoapp1.westcentralus.cloudapp.azure.com
```

View reverse DNS for an existing PublicIpAddress

To view the configured value for an existing PublicIpAddress:

PowerShell

```
Get-AzPublicIpAddress -Name "PublicIp" -ResourceGroupName "MyResourceGroup"
```

Azure classic CLI

```
azure network public-ip show -n PublicIp -g MyResourceGroup
```

Azure CLI

```
az network public-ip show --name PublicIp --resource-group MyResourceGroup
```

Remove reverse DNS from existing Public IP Addresses

To remove a reverse DNS property from an existing PublicIpAddress:

PowerShell

```
$pip = Get-AzPublicIpAddress -Name "PublicIp" -ResourceGroupName "MyResourceGroup"
$pip.DnsSettings.ReverseFqdn = ""
Set-AzPublicIpAddress -PublicIpAddress $pip
```

Azure classic CLI

```
azure network public-ip set -n PublicIp -g MyResourceGroup –f ""
```

Azure CLI

```
az network public-ip update --resource-group MyResourceGroup --name PublicIp --reverse-fqdn ""
```

Configure reverse DNS for Cloud Services

This section provides detailed instructions for how to configure reverse DNS for Cloud Services in the Classic deployment model, using Azure PowerShell. Configuring reverse DNS for Cloud Services is not supported via the Azure portal, Azure classic CLI, or Azure CLI.

Add reverse DNS to existing Cloud Services

To add a reverse DNS record to an existing Cloud Service:

```
Set-AzureService –ServiceName "contosoapp1" –Description "App1 with Reverse DNS" –ReverseDnsFqdn
"contosoapp1.cloudapp.net."
```

Create a Cloud Service with reverse DNS

To create a new Cloud Service with the reverse DNS property already specified:

```
New-AzureService –ServiceName "contosoapp1" –Location "West US" –Description "App1 with Reverse DNS" –
ReverseDnsFqdn "contosoapp1.cloudapp.net."
```

View reverse DNS for existing Cloud Services

To view the reverse DNS property for an existing Cloud Service:

```
Get-AzureService "contosoapp1"
```

Remove reverse DNS from existing Cloud Services

To remove a reverse DNS property from an existing Cloud Service:

```
Set-AzureService –ServiceName "contosoapp1" –Description "App1 with Reverse DNS" –ReverseDnsFqdn ""
```

FAQ

How much do reverse DNS records cost?

They're free! There is no additional cost for reverse DNS records or queries.

Will my reverse DNS records resolve from the internet?

Yes. Once you set the reverse DNS property for your Azure service, Azure manages all the DNS delegations and DNS zones required to ensure that reverse DNS record resolves for all Internet users.

Are default reverse DNS records created for my Azure services?

No. Reverse DNS is an opt-in feature. No default reverse DNS records are created if you choose not to configure them.

What is the format for the fully-qualified domain name (FQDN)?

FQDNs are specified in forward order, and must be terminated by a dot (for example, "app1.contoso.com.").

What happens if the validation check for the reverse DNS I've specified fails?

Where the reverse DNS validation check fails, the operation to configure the reverse DNS record fails. Correct the reverse DNS value as required, and retry.

Can I configure reverse DNS for Azure App Service?

No. Reverse DNS is not supported for the Azure App Service.

Can I configure multiple reverse DNS records for my Azure service?

No. Azure supports a single reverse DNS record for each Azure Cloud Service or PublicIpAddress.

Can I configure reverse DNS for IPv6 PublicIpAddress resources?

No. Azure currently supports reverse DNS only for IPv4 PublicIpAddress resources and Cloud Services.

Can I send emails to external domains from my Azure Compute services?

The technical ability to send email directly from an Azure deployment depends on the subscription type. Regardless of subscription type, Microsoft recommends using trusted mail relay services to send outgoing mail.

For further details, see Enhanced Azure Security for sending Emails – November 2017 Update.

Next steps

For more information on reverse DNS, see reverse DNS lookup on Wikipedia.
Learn how to host the reverse lookup zone for your ISP-assigned IP range in Azure DNS.

Import and export a DNS zone file using the Azure CLI

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

This article walks you through how to import and export DNS zone files for Azure DNS using the Azure CLI.

Introduction to DNS zone migration

A DNS zone file is a text file that contains details of every Domain Name System (DNS) record in the zone. It follows a standard format, making it suitable for transferring DNS records between DNS systems. Using a zone file is a quick, reliable, and convenient way to transfer a DNS zone into or out of Azure DNS.

Azure DNS supports importing and exporting zone files by using the Azure command-line interface (CLI). Zone file import is **not** currently supported via Azure PowerShell or the Azure portal.

The Azure CLI is a cross-platform command-line tool used for managing Azure services. It is available for the Windows, Mac, and Linux platforms from the Azure downloads page. Cross-platform support is important for importing and exporting zone files, because the most common name server software, BIND, typically runs on Linux.

Obtain your existing DNS zone file

Before you import a DNS zone file into Azure DNS, you need to obtain a copy of the zone file. The source of this file depends on where the DNS zone is currently hosted.

- If your DNS zone is hosted by a partner service (such as a domain registrar, dedicated DNS hosting provider, or alternative cloud provider), that service should provide the ability to download the DNS zone file.
- If your DNS zone is hosted on Windows DNS, the default folder for the zone files is **%systemroot%\system32\dns**. The full path to each zone file also shows on the **General** tab of the DNS console.
- If your DNS zone is hosted by using BIND, the location of the zone file for each zone is specified in the BIND configuration file **named.conf**.

Import a DNS zone file into Azure DNS

Importing a zone file creates a new zone in Azure DNS if one does not already exist. If the zone already exists, the record sets in the zone file must be merged with the existing record sets.

Merge behavior

- By default, existing and new record sets are merged. Identical records within a merged record set are de-duplicated.
- When record sets are merged, the time to live (TTL) of preexisting record sets is used.
- Start of Authority (SOA) parameters (except `host`) are always taken from the imported zone file. Similarly, for the name server record set at the zone apex, the TTL is always taken from the imported zone file.
- An imported CNAME record does not replace an existing CNAME record with the same name.
- When a conflict arises between a CNAME record and another record of the same name but different type (regardless of which is existing or new), the existing record is retained.

Additional information about importing

The following notes provide additional technical details about the zone import process.

- The `$TTL` directive is optional, and it is supported. When no `$TTL` directive is given, records without an explicit TTL are imported set to a default TTL of 3600 seconds. When two records in the same record set specify different TTLs, the lower value is used.
- The `$ORIGIN` directive is optional, and it is supported. When no `$ORIGIN` is set, the default value used is the zone name as specified on the command line (plus the terminating ".").
- The `$INCLUDE` and `$GENERATE` directives are not supported.
- These record types are supported: A, AAAA, CAA, CNAME, MX, NS, SOA, SRV, and TXT.
- The SOA record is created automatically by Azure DNS when a zone is created. When you import a zone file, all SOA parameters are taken from the zone file *except* the `host` parameter. This parameter uses the value provided by Azure DNS. This is because this parameter must refer to the primary name server provided by Azure DNS.
- The name server record set at the zone apex is also created automatically by Azure DNS when the zone is created. Only the TTL of this record set is imported. These records contain the name server names provided by Azure DNS. The record data is not overwritten by the values contained in the imported zone file.
- During Public Preview, Azure DNS supports only single-string TXT records. Multistring TXT records are be concatenated and truncated to 255 characters.

CLI format and values

The format of the Azure CLI command to import a DNS zone is:

```
az network dns zone import -g <resource group> -n <zone name> -f <zone file name>
```

Values:

- `<resource group>` is the name of the resource group for the zone in Azure DNS.
- `<zone name>` is the name of the zone.
- `<zone file name>` is the path/name of the zone file to be imported.

If a zone with this name does not exist in the resource group, it is created for you. If the zone already exists, the imported record sets are merged with existing record sets.

Step 1. Import a zone file

To import a zone file for the zone **contoso.com**.

1. If you don't have one already, you need to create a Resource Manager resource group.

   ```
   az group create --group myresourcegroup -l westeurope
   ```

2. To import the zone **contoso.com** from the file **contoso.com.txt** into a new DNS zone in the resource group **myresourcegroup**, you will run the command `az network dns zone import`.
   This command loads the zone file and parses it. The command executes a series of commands on the Azure DNS service to create the zone and all the record sets in the zone. The command reports progress in the console window, along with any errors or warnings. Because record sets are created in series, it may take a few minutes to import a large zone file.

   ```
   az network dns zone import -g myresourcegroup -n contoso.com -f contoso.com.txt
   ```

Step 2. Verify the zone

To verify the DNS zone after you import the file, you can use any one of the following methods:

- You can list the records by using the following Azure CLI command:

```
az network dns record-set list -g myresourcegroup -z contoso.com
```

- You can list the records by using the Azure CLI command `az network dns record-set ns list`.

- You can use `nslookup` to verify name resolution for the records. Because the zone isn't delegated yet, you need to specify the correct Azure DNS name servers explicitly. The following sample shows how to retrieve the name server names assigned to the zone. This also shows how to query the "www" record by using `nslookup`.

```
az network dns record-set ns list -g myresourcegroup -z contoso.com  --output json
```

```
[
  {
   .......
   "name": "@",
    "nsRecords": [
      {
        "additionalProperties": {},
        "nsdname": "ns1-03.azure-dns.com."
      },
      {
        "additionalProperties": {},
        "nsdname": "ns2-03.azure-dns.net."
      },
      {
        "additionalProperties": {},
        "nsdname": "ns3-03.azure-dns.org."
      },
      {
        "additionalProperties": {},
        "nsdname": "ns4-03.azure-dns.info."
      }
    ],
    "resourceGroup": "myresourcegroup",
    "ttl": 86400,
    "type": "Microsoft.Network/dnszones/NS"
  }
]
```

```
nslookup www.contoso.com ns1-03.azure-dns.com

    Server: ns1-01.azure-dns.com
    Address:  40.90.4.1

    Name:www.contoso.com
    Addresses:  134.170.185.46
    134.170.188.221
```

Step 3. Update DNS delegation

After you have verified that the zone has been imported correctly, you need to update the DNS delegation to point to the Azure DNS name servers. For more information, see the article Update the DNS delegation.

Export a DNS zone file from Azure DNS

The format of the Azure CLI command to export a DNS zone is:

```
az network dns zone export -g <resource group> -n <zone name> -f <zone file name>
```

Values:

- `<resource group>` is the name of the resource group for the zone in Azure DNS.
- `<zone name>` is the name of the zone.
- `<zone file name>` is the path/name of the zone file to be exported.

As with the zone import, you first need to sign in, choose your subscription, and configure the Azure CLI to use Resource Manager mode.

To export a zone file

To export the existing Azure DNS zone **contoso.com** in resource group **myresourcegroup** to the file **contoso.com.txt** (in the current folder), run `azure network dns zone export`. This command calls the Azure DNS service to enumerate record sets in the zone and export the results to a BIND-compatible zone file.

```
az network dns zone export -g myresourcegroup -n contoso.com -f contoso.com.txt
```

Next steps

- Learn how to manage record sets and records in your DNS zone.
- Learn how to delegate your domain to Azure DNS.

Delegate an Azure DNS subdomain

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

You can use the Azure portal to delegate a DNS subdomain. For example, if you own the contoso.com domain, you can delegate a subdomain called *engineering* to another, separate zone that you can administer separately from the contoso.com zone.

If you prefer, you can delegate a subdomain using Azure PowerShell.

Prerequisites

To delegate an Azure DNS subdomain, you must first delegate your public domain to Azure DNS. See Delegate a domain to Azure DNS for instructions on how to configure your name servers for delegation. Once your domain is delegated to your Azure DNS zone, you can delegate your subdomain.

> **NOTE**
> Contoso.com is used as an example throughout this article. Substitute your own domain name for contoso.com.

Create a zone for your subdomain

First, create the zone for the **engineering** subdomain.

1. From the Azure portal, select **Create a resource**.
2. In the search box, type **DNS**, and select **DNS zone**.
3. Select **Create**.
4. In the **Create DNS zone** pane, type **engineering.contoso.com** in the **Name** text box.
5. Select the resource group for your zone. You might want to use the same resource group as the parent zone to keep similar resources together.
6. Click **Create**.
7. After the deployment succeeds, go to the new zone.

Note the name servers

Next, note the four name servers for the engineering subdomain.

On the **engineering** zone pane, note the four name servers for the zone. You will use these name servers later.

Create a test record

Create an **A** record to use for testing. For example, create a **www** A record and configure it with a **10.10.10.10** IP address.

Create an NS record

Next, create a name server (NS) record for the **engineering** zone.

1. Navigate to the zone for the parent domain.
2. Select **+ Record set**.
3. On the **Add record set** pane, type **engineering** in the **Name** text box.

4. For **Type**, select **NS**.
5. Under **Name server**, enter the four name servers that you recorded previously from the **engineering** zone.
6. Click **OK**.

Test the delegation

Use nslookup to test the delegation.

1. Open a PowerShell window.
2. At command prompt, type `nslookup www.engineering.contoso.com.`
3. You should receive a non-authoritative answer showing the address **10.10.10.10**.

Next steps

Learn how to configure reverse DNS for services hosted in Azure.

Delegate an Azure DNS subdomain using Azure PowerShell

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

You can use Azure PowerShell to delegate a DNS subdomain. For example, if you own the contoso.com domain, you can delegate a subdomain called *engineering* to another, separate zone that you can administer separately from the contoso.com zone.

If you prefer, you can delegate a subdomain using the Azure Portal.

> **NOTE**
> Contoso.com is used as an example throughout this article. Substitute your own domain name for contoso.com.

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

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.

Prerequisites

To delegate an Azure DNS subdomain, you must first delegate your public domain to Azure DNS. See Delegate a domain to Azure DNS for instructions on how to configure your name servers for delegation. Once your domain is

delegated to your Azure DNS zone, you can delegate your subdomain.

Create a zone for your subdomain

First, create the zone for the **engineering** subdomain.

```
New-AzDnsZone -ResourceGroupName <resource group name> -Name engineering.contoso.com
```

Note the name servers

Next, note the four name servers for the engineering subdomain.

```
Get-AzDnsRecordSet -ZoneName engineering.contoso.com -ResourceGroupName <resource group name> -RecordType NS
```

Create a test record

Create an **A** record in the engineering zone to use for testing.

```
New-AzDnsRecordSet -ZoneName engineering.contoso.com -ResourceGroupName <resource group name> -Name www -
RecordType A -ttl 3600 -DnsRecords (New-AzDnsRecordConfig -IPv4Address 10.10.10.10)
```

.

Create an NS record

Next, create a name server (NS) record for the **engineering** zone in the contoso.com zone.

```
$Records = @()
$Records += New-AzDnsRecordConfig -Nsdname <name server 1 noted previously>
$Records += New-AzDnsRecordConfig -Nsdname <name server 2 noted previously>
$Records += New-AzDnsRecordConfig -Nsdname <name server 3 noted previously>
$Records += New-AzDnsRecordConfig -Nsdname <name server 4 noted previously>
$RecordSet = New-AzDnsRecordSet -Name engineering -RecordType NS -ResourceGroupName <resource group name> -TTL
3600 -ZoneName contoso.com -DnsRecords $Records
```

Test the delegation

Use nslookup to test the delegation.

1. Open a PowerShell window.
2. At command prompt, type `nslookup www.engineering.contoso.com.`
3. You should receive a non-authoritative answer showing the address **10.10.10.10**.

Next steps

Learn how to configure reverse DNS for services hosted in Azure.

How Azure DNS works with other Azure services

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

Azure DNS is a hosted DNS management and name resolution service. You can use it to create public DNS names for other applications and services that you deploy in Azure. Creating a name for an Azure service in your custom domain is simple. You just add a record of the correct type for your service.

- For dynamically allocated IP addresses, you can create a DNS CNAME record that maps to the DNS name that Azure created for your service. DNS standards prevent you from using a CNAME record for the zone apex. You can use an alias record instead. For more information, see Tutorial: Configure an alias record to refer to an Azure Public IP address.
- For statically allocated IP addresses, you can create a DNS A record by using any name, which includes a *naked domain* name at the zone apex.

The following table outlines the supported record types you can use for various Azure services. As the table shows, Azure DNS supports only DNS records for Internet-facing network resources. Azure DNS can't be used for name resolution of internal, private addresses.

AZURE SERVICE	NETWORK INTERFACE	DESCRIPTION
Azure Application Gateway	Front-end public IP	You can create a DNS A or CNAME record.
Azure Load Balancer	Front-end public IP	You can create a DNS A or CNAME record. Load Balancer can have an IPv6 public IP address that's dynamically assigned. Create a CNAME record for an IPv6 address.
Azure Traffic Manager	Public name	You can create an alias record that maps to the trafficmanager.net name assigned to your Traffic Manager profile. For more information, see Tutorial: Configure an alias record to support apex domain names with Traffic Manager.
Azure Cloud Services	Public IP	For statically allocated IP addresses, you can create a DNS A record. For dynamically allocated IP addresses, you must create a CNAME record that maps to the *cloudapp.net* name.
Azure App Service	External IP	For external IP addresses, you can create a DNS A record. Otherwise, you must create a CNAME record that maps to the azurewebsites.net name. For more information, see Map a custom domain name to an Azure app.

AZURE SERVICE	NETWORK INTERFACE	DESCRIPTION
Azure Resource Manager VMs	Public IP	Resource Manager VMs can have public IP addresses. A VM with a public IP address also can be behind a load balancer. You can create a DNS A, CNAME, or alias record for the public address. You can use this custom name to bypass the VIP on the load balancer.
Classic VMs	Public IP	Classic VMs created by using PowerShell or CLI can be configured with a dynamic or static (reserved) virtual address. You can create a DNS CNAME or an A record, respectively.

How to protect DNS zones and records

2/25/2020 • 9 minutes to read • Edit Online

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.

DNS zones and records are critical resources. Deleting a DNS zone or a single DNS record can result in a service outage. It's important that DNS zones and records are protected against unauthorized or accidental changes.

This article explains how Azure DNS enables you to protect your private DNS zones and records against such changes. We apply two powerful securities features provided by Azure Resource Manager: role-based access control and resource locks.

Role-based access control

Azure Role-Based Access Control (RBAC) enables fine-grained access management for Azure users, groups, and resources. With RBAC, you can grant the level of access that users need. For more information about how RBAC helps you manage access, see What is Role-Based Access Control.

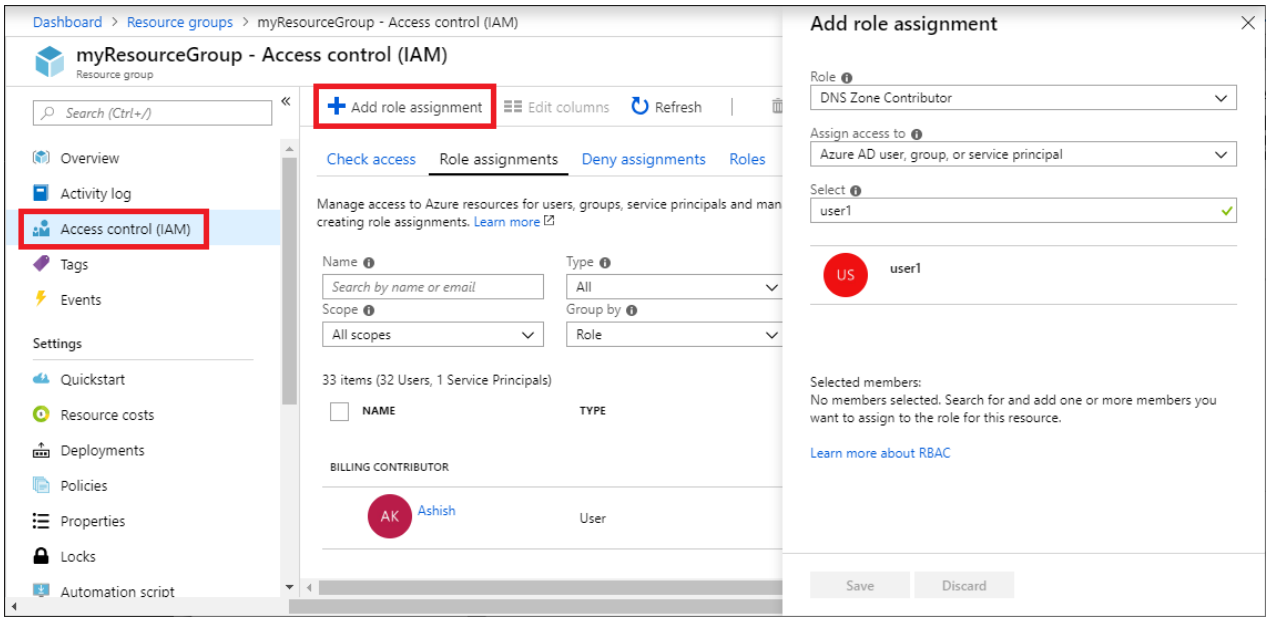
The DNS Zone Contributor role

The DNS Zone Contributor role is a built-in role for managing private DNS resources. This role applied to a user or group enables them to manage DNS resources.

The resource group *myResourceGroup* contains five zones for Contoso Corporation. Granting the DNS administrator DNS Zone Contributor permissions to that resource group, enables full control over those DNS zones. It avoids granting unnecessary permissions. The DNS administrator can't create or stop virtual machines.

The simplest way to assign RBAC permissions is via the Azure portal.

Open **Access control (IAM)** for the resource group, then select **Add**, then select the **DNS Zone Contributor** role. Select the required users or groups to grant permissions.



Permissions can also be granted using Azure PowerShell:

```
# Grant 'DNS Zone Contributor' permissions to all zones in a resource group

$usr = "<user email address>"
$rol = "DNS Zone Contributor"
$rsg = "<resource group name>"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -ResourceGroupName $rsg
```

The equivalent command is also available via the Azure CLI:

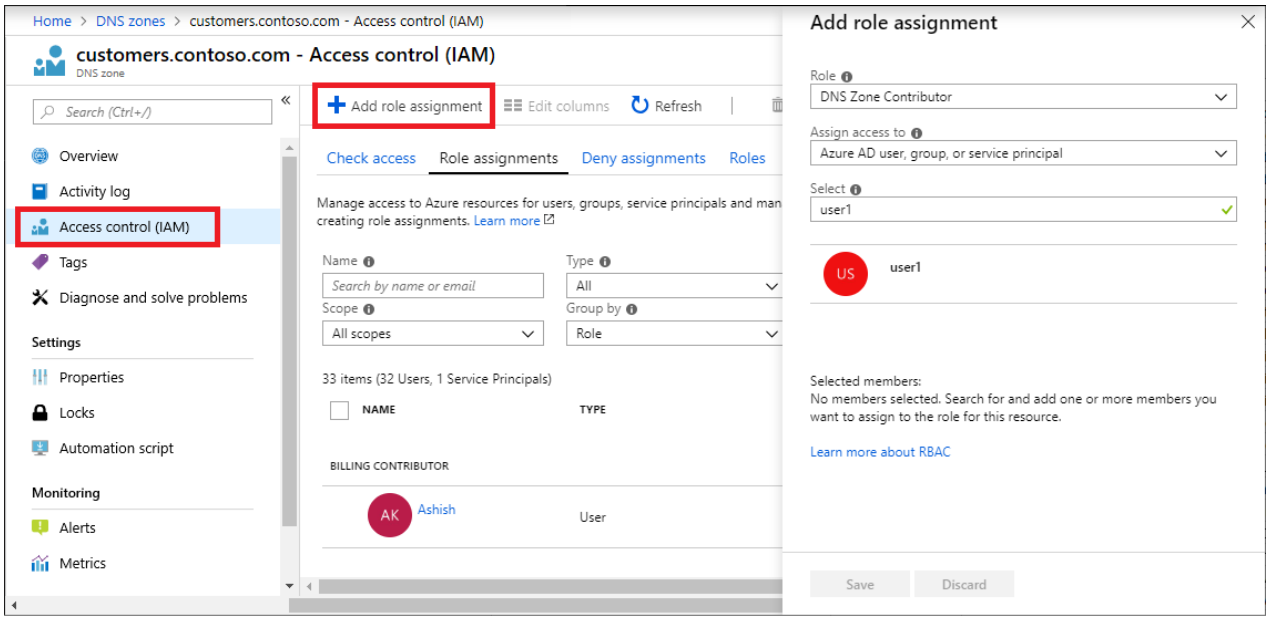
```
# Grant 'DNS Zone Contributor' permissions to all zones in a resource group

az role assignment create \
--assignee "<user email address>" \
--role "DNS Zone Contributor" \
--resource-group "<resource group name>"
```

Zone level RBAC

Azure RBAC rules can be applied to a subscription, a resource group or to an individual resource. That resource can be an individual DNS zone, or an individual record set.

For example, the resource group *myResourceGroup* contains the zone *contoso.com* and a subzone *customers.contoso.com*. CNAME records are created for each customer account. The administrator account used to manage CNAME records is assigned permissions to create records in the *customers.contoso.com* zone. The account can manage *customers.contoso.com* only.

Zone-level RBAC permissions can be granted via the Azure portal. Open **Access control (IAM)** for the zone, select **Add**, then select the **DNS Zone Contributor** role and select the required users or groups to grant permissions.



Permissions can also be granted using Azure PowerShell:

```
# Grant 'DNS Zone Contributor' permissions to a specific zone

$usr = "<user email address>"
$rol = "DNS Zone Contributor"
$rsg = "<resource group name>"
$zon = "<zone name>"
$typ = "Microsoft.Network/DNSZones"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -ResourceGroupName $rsg -ResourceName $zon -
ResourceType $typ
```

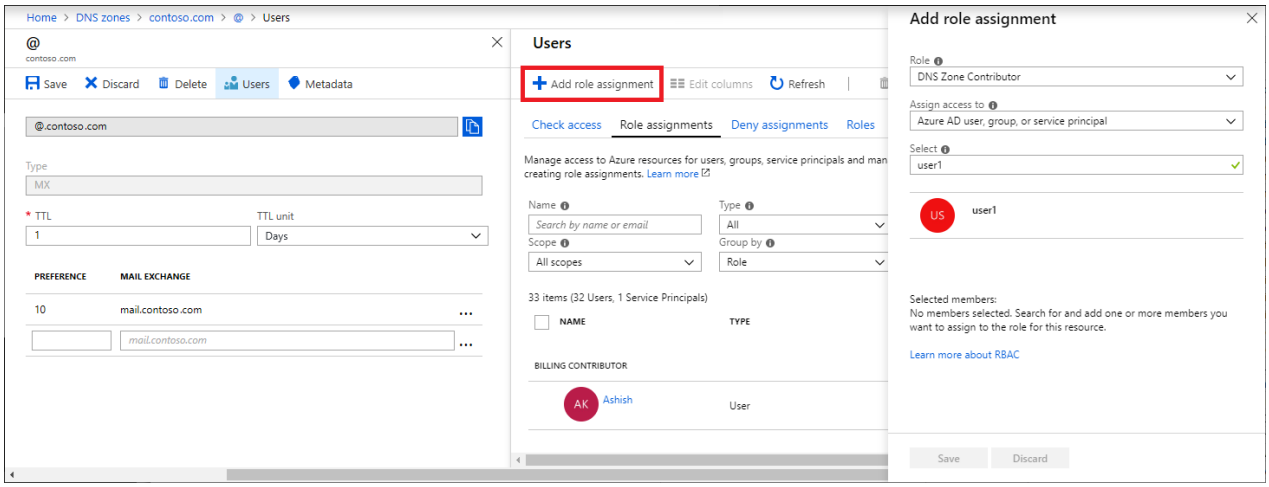
The equivalent command is also available via the Azure CLI:

```
# Grant 'DNS Zone Contributor' permissions to a specific zone

az role assignment create \
--assignee <user email address> \
--role "DNS Zone Contributor" \
--scope "/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/DnsZones/<zone name>/"
```

Record set level RBAC

Permissions are applied at the record set level. The user is granted control to entries they need and are unable to make any other changes.

Record-set level RBAC permissions can be configured via the Azure portal, using the **Access Control (IAM)** button in the record set page:



Record-set level RBAC permissions can also be granted using Azure PowerShell:

```
# Grant permissions to a specific record set

$usr = "<user email address>"
$rol = "DNS Zone Contributor"
$sco =
"/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/dnszones/<zone name>/<record type>/<record name>"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -Scope $sco
```

The equivalent command is also available via the Azure CLI:

```
# Grant permissions to a specific record set

az role assignment create \
--assignee "<user email address>" \
--role "DNS Zone Contributor" \
--scope "/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/dnszones/<zone name>/<record type>/<record name>"
```

Custom roles

The built-in DNS Zone Contributor role enables full control over a DNS resource. It's possible to build your own custom Azure roles to provide finer-grained control.

The account that is used to manage CNAMEs is granted permission to manage CNAME records only. The account is unable to modify records of other types. The account is unable to do zone-level operations such as zone delete.

The following example shows a custom role definition for managing CNAME records only:

```
{
    "Name": "DNS CNAME Contributor",
    "Id": "",
    "IsCustom": true,
    "Description": "Can manage DNS CNAME records only.",
    "Actions": [
        "Microsoft.Network/dnsZones/CNAME/*",
        "Microsoft.Network/dnsZones/read",
        "Microsoft.Authorization/*/read",
        "Microsoft.Insights/alertRules/*",
        "Microsoft.ResourceHealth/availabilityStatuses/read",
        "Microsoft.Resources/deployments/*",
        "Microsoft.Resources/subscriptions/resourceGroups/read",
        "Microsoft.Support/*"
    ],
    "NotActions": [
    ],
    "AssignableScopes": [
        "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e"
    ]
}
```

The Actions property defines the following DNS-specific permissions:

- `Microsoft.Network/dnsZones/CNAME/*` grants full control over CNAME records
- `Microsoft.Network/dnsZones/read` grants permission to read DNS zones, but not to modify them, enabling you to see the zone in which the CNAME is being created.

The remaining Actions are copied from the DNS Zone Contributor built-in role.

> **NOTE**
> Using a custom RBAC role to prevent deleting record sets while still allowing them to be updated is not an effective control. It prevents record sets from being deleted, but it does not prevent them from being modified. Permitted modifications include adding and removing records from the record set, including removing all records to leave an empty record set. This has the same effect as deleting the record set from a DNS resolution viewpoint.

Custom role definitions can't currently be defined via the Azure portal. A custom role based on this role definition can be created using Azure PowerShell:

```
# Create new role definition based on input file
New-AzRoleDefinition -InputFile <file path>
```

It can also be created via the Azure CLI:

```
# Create new role definition based on input file
az role create -inputfile <file path>
```

The role can then be assigned in the same way as built-in roles, as described earlier in this article.

For more information on how to create, manage, and assign custom roles, see Custom Roles in Azure RBAC.

Resource locks

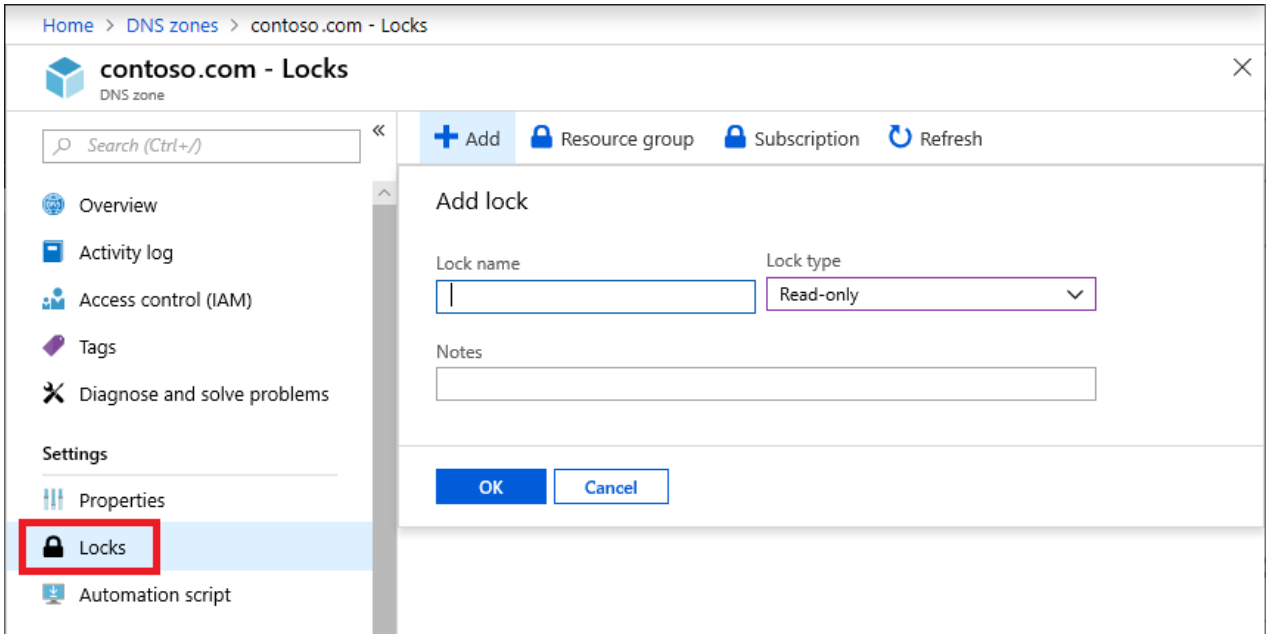
Azure Resource Manager supports another type of security control, the ability to lock resources. Resource locks are applied to the resource, and are effective across all users and roles. For more information, see Lock resources with Azure Resource Manager.

There are two types of resource lock: **CanNotDelete** and **ReadOnly**. These lock types can be applied either to a Private DNS zone, or to an individual record set. The following sections describe several common scenarios, and how to support them using resource locks.

Protecting against all changes

To prevent changes being made, apply a ReadOnly lock to the zone. This lock prevents new record sets from being created, and existing record sets from being modified or deleted.

Zone level resource locks can be created via the Azure portal. From the DNS zone page, select **Locks**, then select **+Add**:



Zone-level resource locks can also be created via Azure PowerShell:

```
# Lock a DNS zone

$lvl = "<lock level>"
$lnm = "<lock name>"
$rsc = "<zone name>"
$rty = "Microsoft.Network/DNSZones"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rsc -ResourceType $rty -ResourceGroupName
$rsg
```

The equivalent command is also available via the Azure CLI:

```
# Lock a DNS zone

az lock create \
--lock-type "<lock level>" \
--name "<lock name>" \
--resource-name "<zone name>" \
--namespace "Microsoft.Network" \
--resource-type "DnsZones" \
--resource-group "<resource group name>"
```

Protecting individual records

To prevent an existing DNS record set against modification, apply a ReadOnly lock to the record set.

> **NOTE**
> Applying a CanNotDelete lock to a record set is not an effective control. It prevents the record set from being deleted, but it does not prevent it from being modified. Permitted modifications include adding and removing records from the record set, including removing all records to leave an empty record set. This has the same effect as deleting the record set from a DNS resolution viewpoint.

Record set level resource locks can currently only be configured using Azure PowerShell. They aren't supported in the Azure portal or Azure CLI.

```
# Lock a DNS record set

$lvl = "<lock level>"
$lnm = "<lock name>"
$rsc = "<zone name>/<record set name>"
$rty = "Microsoft.Network/DNSZones/<record type>"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rsc -ResourceType $rty -ResourceGroupName
$rsg
```

Protecting against zone deletion

When a zone is deleted in Azure DNS, all record sets in the zone are deleted. This operation can't be undone. Accidentally deleting a critical zone has the potential to have a significant business impact. It's important to protect against accidental zone deletion.

Applying a CanNotDelete lock to a zone prevents the zone from being deleted. Locks are inherited by child resources. A lock prevents any record sets in the zone from being deleted. As described in the note above, it's ineffective since records can still be removed from the existing record sets.

As an alternative, apply a CanNotDelete lock to a record set in the zone, such as the SOA record set. The zone isn't deleted without also deleting the record sets. This lock protects against zone deletion, while still allowing record sets within the zone to be modified freely. If an attempt is made to delete the zone, Azure Resource Manager detects this removal. The removal would also delete the SOA record set, Azure Resource Manager blocks the call because the SOA is locked. No record sets are deleted.

The following PowerShell command creates a CanNotDelete lock against the SOA record of the given zone:

```
# Protect against zone delete with CanNotDelete lock on the record set

$lvl = "CanNotDelete"
$lnm = "<lock name>"
$rsc = "<zone name>/@"
$rty = "Microsoft.Network/DNSZones/SOA"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rsc -ResourceType $rty -ResourceGroupName
$rsg
```

Another option to prevent accidental zone deletion is by using a custom role. This role ensures the accounts used to manage your zones don't have zone delete permissions.

When you do need to delete a zone, you can enforce a two-step delete:

- First, grant zone delete permissions
- Second, grant permissions to delete the zone.

The custom role works for all zones accessed by those accounts. Accounts with zone delete permissions, such as the subscription owner, can still accidentally delete a zone.

It's possible to use both approaches - resource locks and custom roles - at the same time, as a defense-in-depth approach to DNS zone protection.

Next steps

- For more information about working with RBAC, see Get started with access management in the Azure portal.
- For more information about working with resource locks, see Lock resources with Azure Resource Manager.

Create DNS zones and record sets using the .NET SDK

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

You can automate operations to create, delete, or update DNS zones, record sets, and records by using the DNS SDK with the .NET DNS Management library. A full Visual Studio project is available here.

Create a service principal account

Typically, programmatic access to Azure resources is granted via a dedicated account rather than your own user credentials. These dedicated accounts are called 'service principal' accounts. To use the Azure DNS SDK sample project, you first need to create a service principal account and assign it the correct permissions.

1. Follow these instructions to create a service principal account (the Azure DNS SDK sample project assumes password-based authentication.)
2. Create a resource group (here's how).
3. Use Azure RBAC to grant the service principal account 'DNS Zone Contributor' permissions to the resource group (here's how.)
4. If using the Azure DNS SDK sample project, edit the 'program.cs' file as follows:
   - Insert the correct values for the `tenantId`, `clientId` (also known as account ID), `secret` (service principal account password) and `subscriptionId` as used in step 1.
   - Enter the resource group name chosen in step 2.
   - Enter a DNS zone name of your choice.

NuGet packages and namespace declarations

To use the Azure DNS .NET SDK, you need to install the **Azure DNS Management Library** NuGet package and other required Azure packages.

1. In **Visual Studio**, open a project or new project.
2. Go to **Tools > NuGet Package Manager > Manage NuGet Packages for Solution...**.
3. Click **Browse**, enable the **Include prerelease** checkbox, and type **Microsoft.Azure.Management.Dns** into the search box.
4. Select the package and click **Install** to add it to your Visual Studio project.
5. Repeat the process above to also install the following packages: **Microsoft.Rest.ClientRuntime.Azure.Authentication** and **Microsoft.Azure.Management.ResourceManager**.

Add namespace declarations

Add the following namespace declarations

```
using Microsoft.Rest.Azure.Authentication;
using Microsoft.Azure.Management.Dns;
using Microsoft.Azure.Management.Dns.Models;
```

Initialize the DNS management client

The `DnsManagementClient` contains the methods and properties necessary for managing DNS zones and record sets. The following code logs into the service principal account and creates a `DnsManagementClient` object.

```
// Build the service credentials and DNS management client
var serviceCreds = await ApplicationTokenProvider.LoginSilentAsync(tenantId, clientId, secret);
var dnsClient = new DnsManagementClient(serviceCreds);
dnsClient.SubscriptionId = subscriptionId;
```

Create or update a DNS zone

To create a DNS zone, first a "Zone" object is created to contain the DNS zone parameters. Because DNS zones are not linked to a specific region, the location is set to 'global'. In this example, an Azure Resource Manager 'tag' is also added to the zone.

To actually create or update the zone in Azure DNS, the zone object containing the zone parameters is passed to the `DnsManagementClient.Zones.CreateOrUpdateAsyc` method.

> **NOTE**
>
> DnsManagementClient supports three modes of operation: synchronous ('CreateOrUpdate'), asynchronous ('CreateOrUpdateAsync'), or asynchronous with access to the HTTP response ('CreateOrUpdateWithHttpMessagesAsync'). You can choose any of these modes, depending on your application needs.

Azure DNS supports optimistic concurrency, called Etags. In this example, specifying "*" for the 'If-None-Match' header tells Azure DNS to create a DNS zone if one does not already exist. The call fails if a zone with the given name already exists in the given resource group.

```
// Create zone parameters
var dnsZoneParams = new Zone("global"); // All DNS zones must have location = "global"

// Create an Azure Resource Manager 'tag'.  This is optional.  You can add multiple tags
dnsZoneParams.Tags = new Dictionary<string, string>();
dnsZoneParams.Tags.Add("dept", "finance");

// Create the actual zone.
// Note: Uses 'If-None-Match *' ETAG check, so will fail if the zone exists already.
// Note: For non-async usage, call dnsClient.Zones.CreateOrUpdate(resourceGroupName, zoneName, dnsZoneParams,
null, "*")
// Note: For getting the http response, call
dnsClient.Zones.CreateOrUpdateWithHttpMessagesAsync(resourceGroupName, zoneName, dnsZoneParams, null, "*")
var dnsZone = await dnsClient.Zones.CreateOrUpdateAsync(resourceGroupName, zoneName, dnsZoneParams, null,
"*");
```

Create DNS record sets and records

DNS records are managed as a record set. A record set is a set of records with the same name and record type within a zone. The record set name is relative to the zone name, not the fully qualified DNS name.

To create or update a record set, a "RecordSet" parameters object is created and passed to `DnsManagementClient.RecordSets.CreateOrUpdateAsync`. As with DNS zones, there are three modes of operation: synchronous ('CreateOrUpdate'), asynchronous ('CreateOrUpdateAsync'), or asynchronous with access to the HTTP response ('CreateOrUpdateWithHttpMessagesAsync').

As with DNS zones, operations on record sets include support for optimistic concurrency. In this example, since

neither 'If-Match' nor 'If-None-Match' are specified, the record set is always created. This call overwrites any existing record set with the same name and record type in this DNS zone.

```
// Create record set parameters
var recordSetParams = new RecordSet();
recordSetParams.TTL = 3600;

// Add records to the record set parameter object.  In this case, we'll add a record of type 'A'
recordSetParams.ARecords = new List<ARecord>();
recordSetParams.ARecords.Add(new ARecord("1.2.3.4"));

// Add metadata to the record set.  Similar to Azure Resource Manager tags, this is optional and you can add
multiple metadata name/value pairs
recordSetParams.Metadata = new Dictionary<string, string>();
recordSetParams.Metadata.Add("user", "Mary");

// Create the actual record set in Azure DNS
// Note: no ETAG checks specified, will overwrite existing record set if one exists
var recordSet = await dnsClient.RecordSets.CreateOrUpdateAsync(resourceGroupName, zoneName, recordSetName,
RecordType.A, recordSetParams);
```

Get zones and record sets

The `DnsManagementClient.Zones.Get` and `DnsManagementClient.RecordSets.Get` methods retrieve individual zones and record sets, respectively. RecordSets are identified by their type, name, and the zone and resource group they exist in. Zones are identified by their name and the resource group they exist in.

```
var recordSet = dnsClient.RecordSets.Get(resourceGroupName, zoneName, recordSetName, RecordType.A);
```

Update an existing record set

To update an existing DNS record set, first retrieve the record set, then update the record set contents, then submit the change. In this example, we specify the 'Etag' from the retrieved record set in the 'If-Match' parameter. The call fails if a concurrent operation has modified the record set in the meantime.

```
var recordSet = dnsClient.RecordSets.Get(resourceGroupName, zoneName, recordSetName, RecordType.A);

// Add a new record to the local object.  Note that records in a record set must be unique/distinct
recordSet.ARecords.Add(new ARecord("5.6.7.8"));

// Update the record set in Azure DNS
// Note: ETAG check specified, update will be rejected if the record set has changed in the meantime
recordSet = await dnsClient.RecordSets.CreateOrUpdateAsync(resourceGroupName, zoneName, recordSetName,
RecordType.A, recordSet, recordSet.Etag);
```

List zones and record sets

To list zones, use the *DnsManagementClient.Zones.List...* methods, which support listing either all zones in a given resource group or all zones in a given Azure subscription (across resource groups.) To list record sets, use *DnsManagementClient.RecordSets.List...* methods, which support either listing all record sets in a given zone or only those record sets of a specific type.

Note when listing zones and record sets that results may be paginated. The following example shows how to iterate through the pages of results. (An artificially small page size of '2' is used to force paging; in practice this parameter should be omitted and the default page size used.)

```
// Note: in this demo, we'll use a very small page size (2 record sets) to demonstrate paging
// In practice, to improve performance you would use a large page size or just use the system default
int recordSets = 0;
var page = await dnsClient.RecordSets.ListAllInResourceGroupAsync(resourceGroupName, zoneName, "2");
recordSets += page.Count();

while (page.NextPageLink != null)
{
    page = await dnsClient.RecordSets.ListAllInResourceGroupNextAsync(page.NextPageLink);
    recordSets += page.Count();
}
```

Next steps

Download the Azure DNS .NET SDK sample project, which includes further examples on how to use the Azure DNS .NET SDK, including examples for other DNS record types.

Use Azure DNS to provide custom domain settings for an Azure service

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

Azure DNS provides DNS for a custom domain for any of your Azure resources that support custom domains or that have a fully qualified domain name (FQDN). An example is you have an Azure web app and you want your users to access it by either using contoso.com, or www.contoso.com as an FQDN. This article walks you through configuring your Azure service with Azure DNS for using custom domains.

Prerequisites

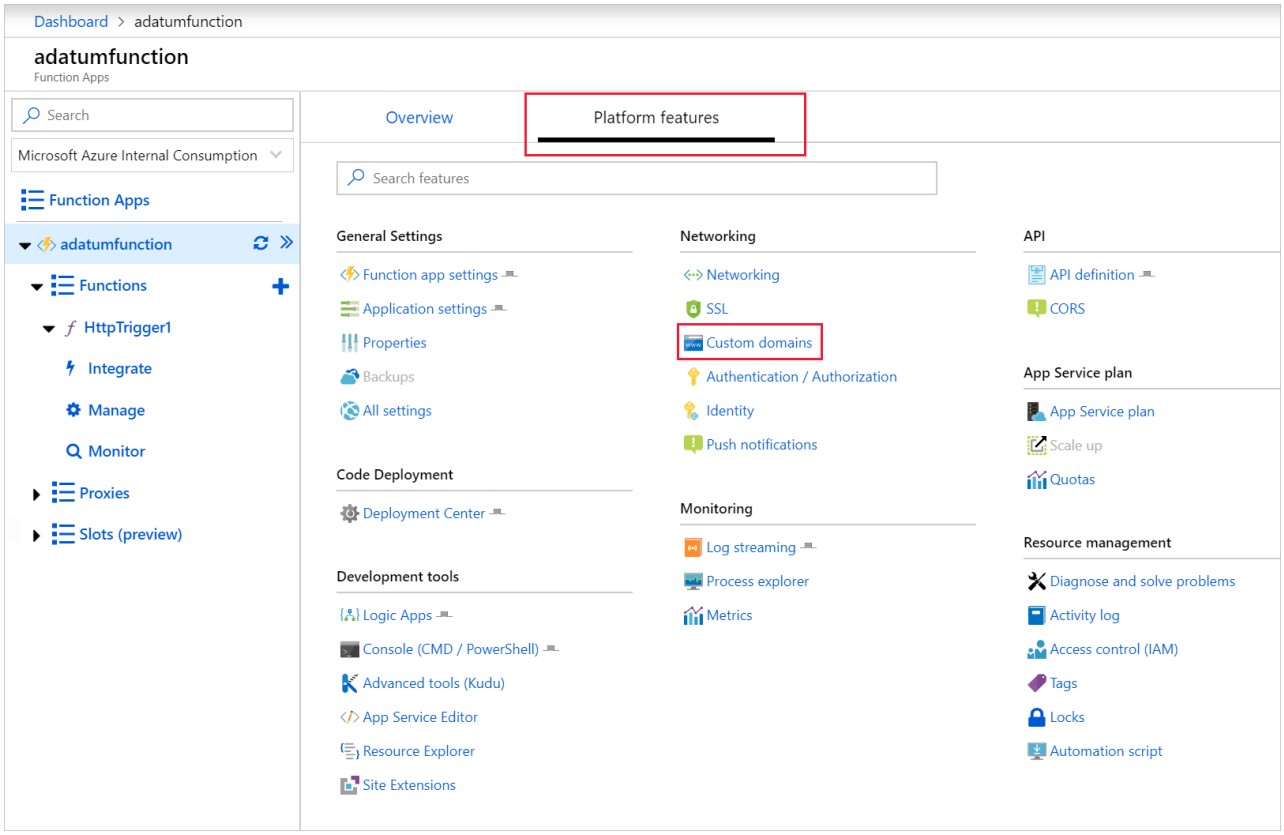
In order to use Azure DNS for your custom domain, you must first delegate your domain to Azure DNS. Visit Delegate a domain to Azure DNS for instructions on how to configure your name servers for delegation. Once your domain is delegated to your Azure DNS zone, you are able to configure the DNS records needed.

You can configure a vanity or custom domain for Azure Function Apps, Public IP addresses, App Service (Web Apps), Blob storage, and Azure CDN.

Azure Function App

To configure a custom domain for Azure function apps, a CNAME record is created as well as configuration on the function app itself.

Navigate to **Function App** and select your function app. Click **Platform features** and under **Networking** click **Custom domains**.



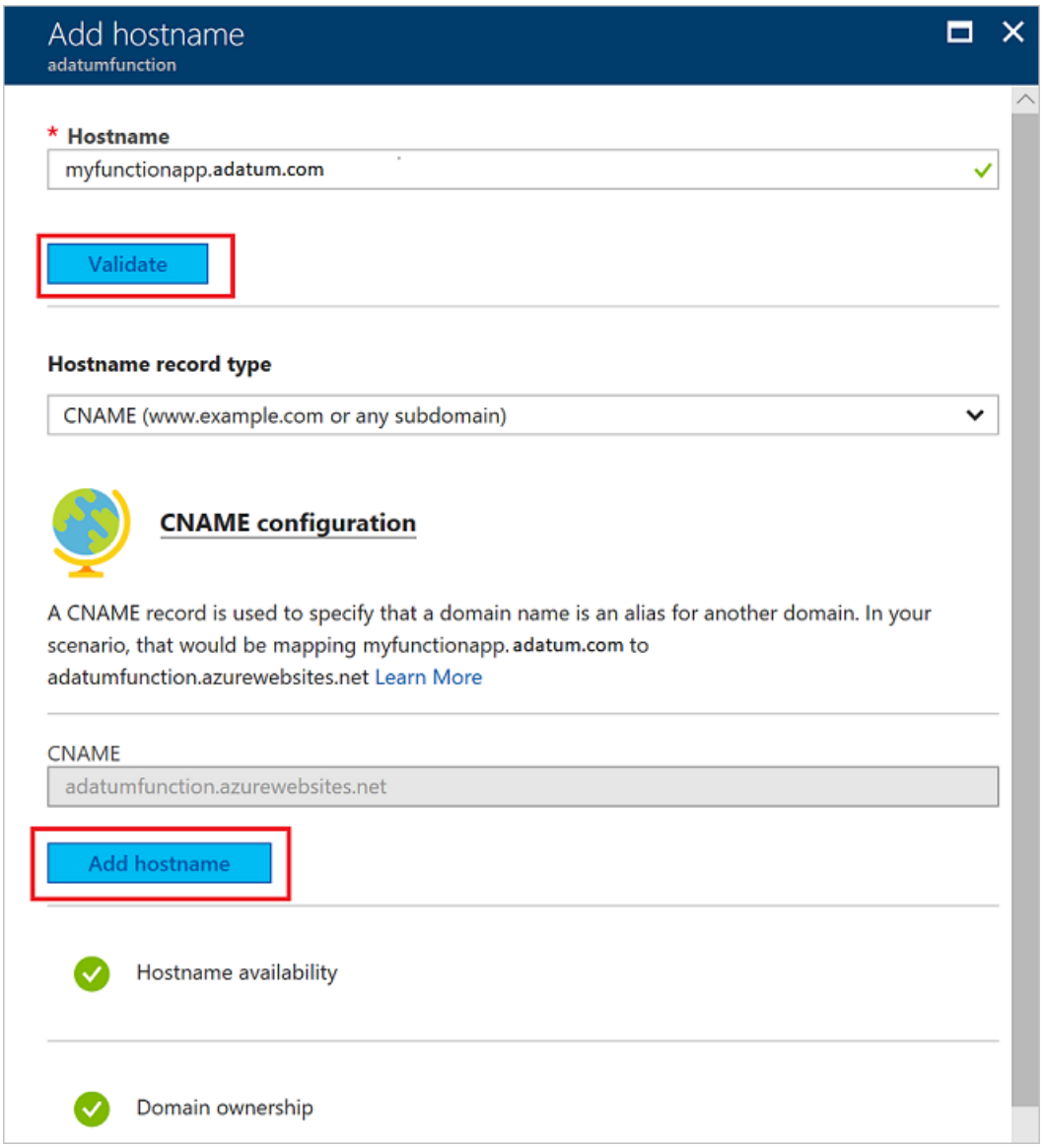
Note the current url on the **Custom domains** blade, this address is used as the alias for the DNS record created.

Navigate to your DNS Zone and click **+ Record set**. Fill out the following information on the **Add record set** blade and click **OK** to create it.

PROPERTY	VALUE	DESCRIPTION
Name	myfunctionapp	This value along with the domain name label is the FQDN for the custom domain name.
Type	CNAME	Use a CNAME record is using an alias.
TTL	1	1 is used for 1 hour
TTL unit	Hours	Hours are used as the time measurement
Alias	adatumfunction.azurewebsites.net	The DNS name you are creating the alias for, in this example it is the adatumfunction.azurewebsites.net DNS name provided by default to the function app.

Navigate back to your function app, click **Platform features**, and under **Networking** click **Custom domains**, then under **Custom Hostnames** click **+ Add hostname**.

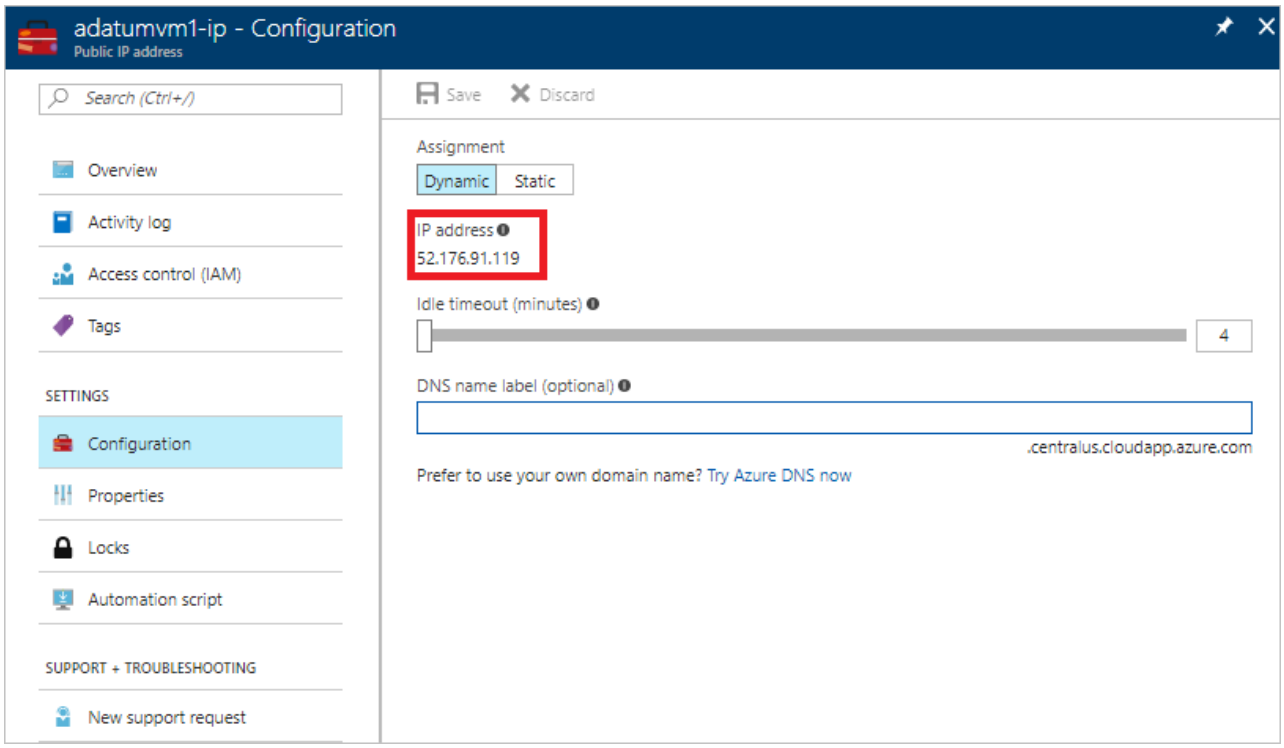
On the **Add hostname** blade, enter the CNAME record in the **hostname** text field and click **Validate**. If the record is found, the **Add hostname** button appears. Click **Add hostname** to add the alias.



Public IP address

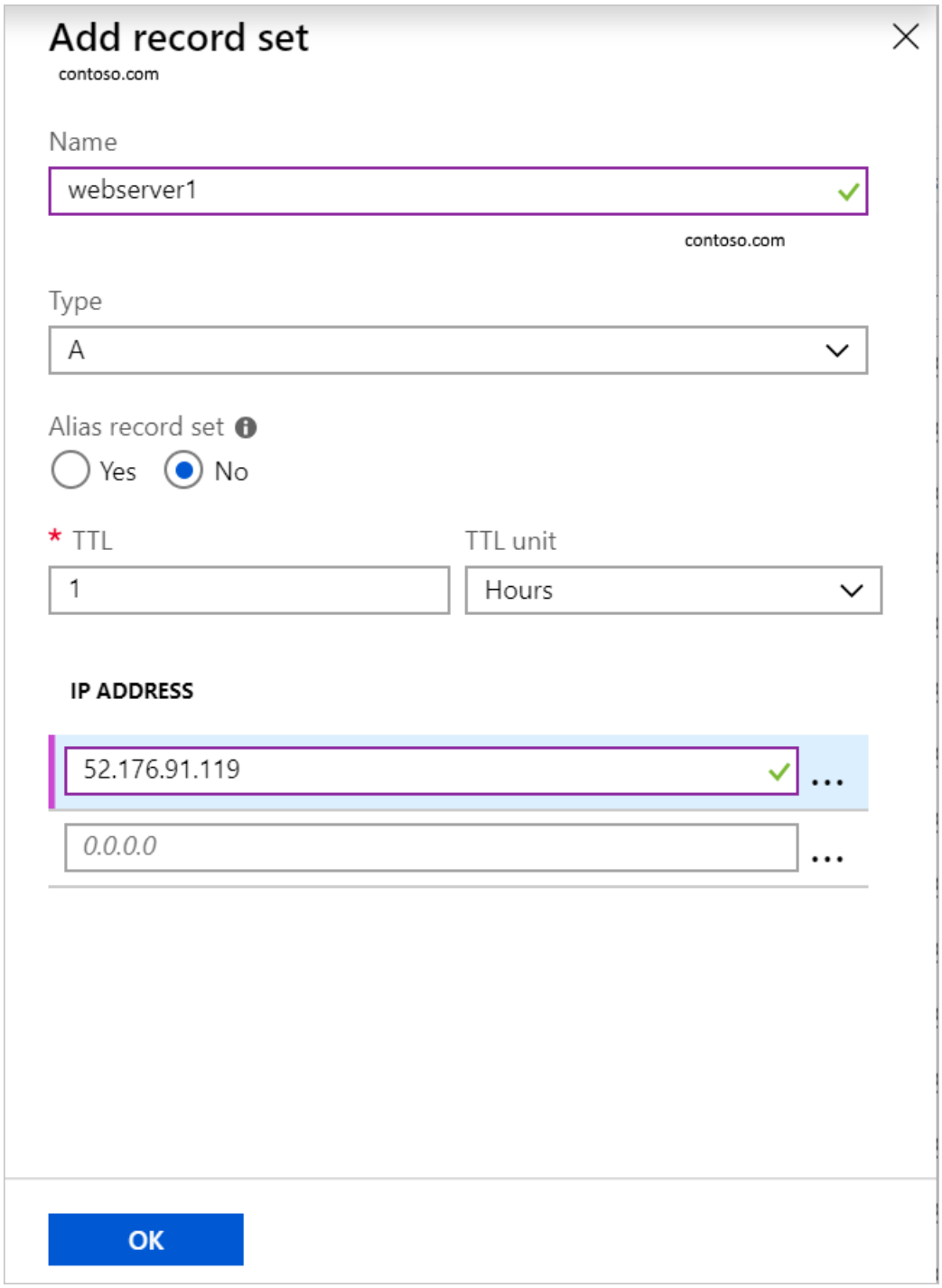
To configure a custom domain for services that use a public IP address resource such as Application Gateway, Load Balancer, Cloud Service, Resource Manager VMs, and, Classic VMs, an A record is used.

Navigate to **Networking** > **Public IP address**, select the Public IP resource and click **Configuration**. Notate the IP address shown.



Navigate to your DNS Zone and click **+ Record set**. Fill out the following information on the **Add record set** blade and click **OK** to create it.

PROPERTY	VALUE	DESCRIPTION
Name	mywebserver	This value along with the domain name label is the FQDN for the custom domain name.
Type	A	Use an A record as the resource is an IP address.
TTL	1	1 is used for 1 hour
TTL unit	Hours	Hours are used as the time measurement
IP Address	`<your ip address>`	The public IP address.



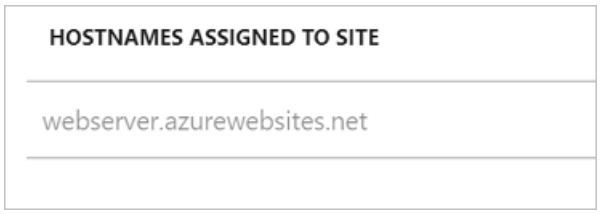
Once the A record is created, run `nslookup` to validate the record resolves.

App Service (Web Apps)

The following steps take you through configuring a custom domain for an app service web app.

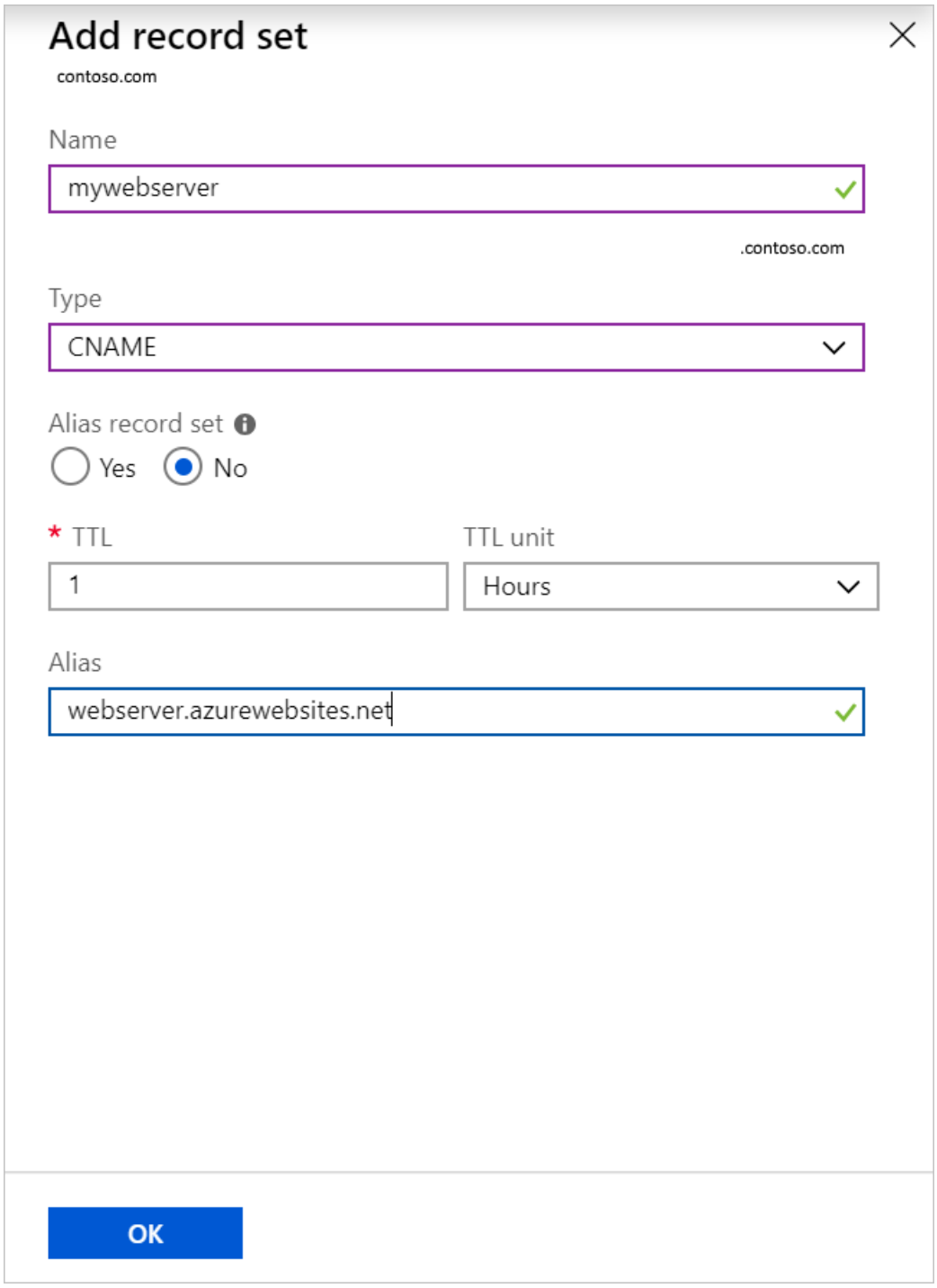
Navigate to **App Service** and select the resource you are configuring a custom domain name, and click **Custom domains**.

Note the current url on the **Custom domains** blade, this address is used as the alias for the DNS record created.

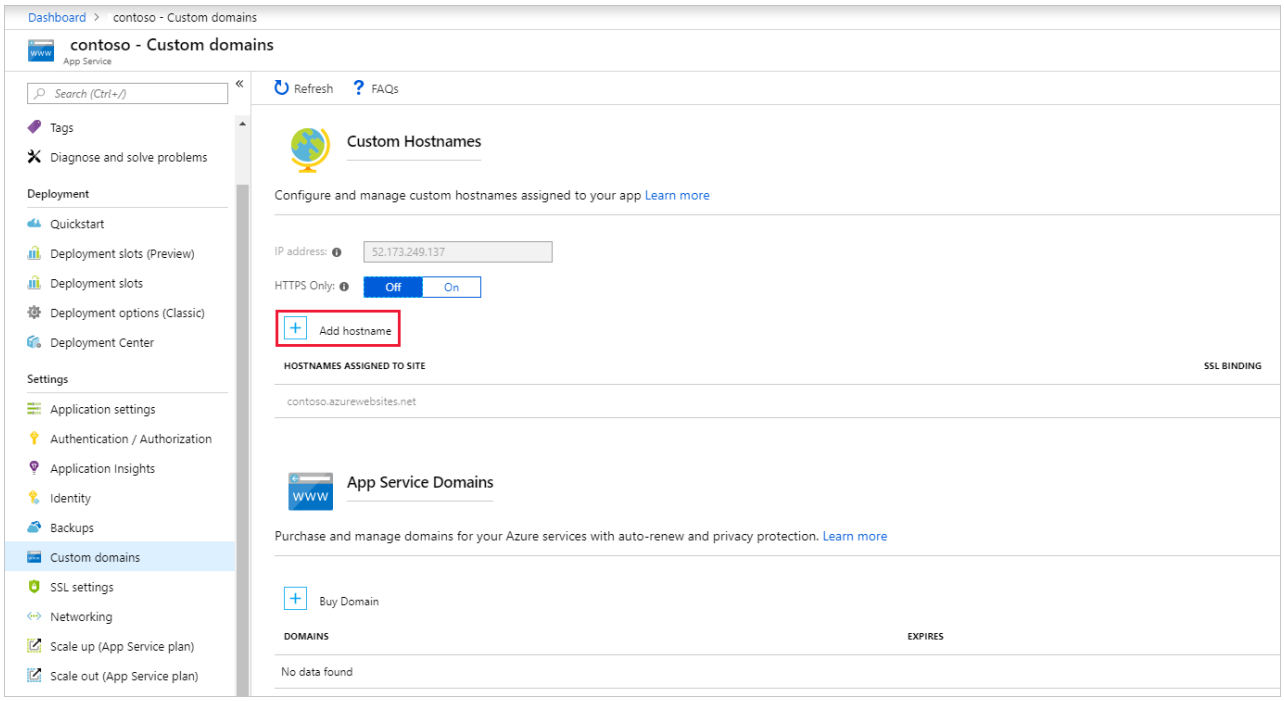


Navigate to your DNS Zone and click **+ Record set**. Fill out the following information on the **Add record set** blade and click **OK** to create it.

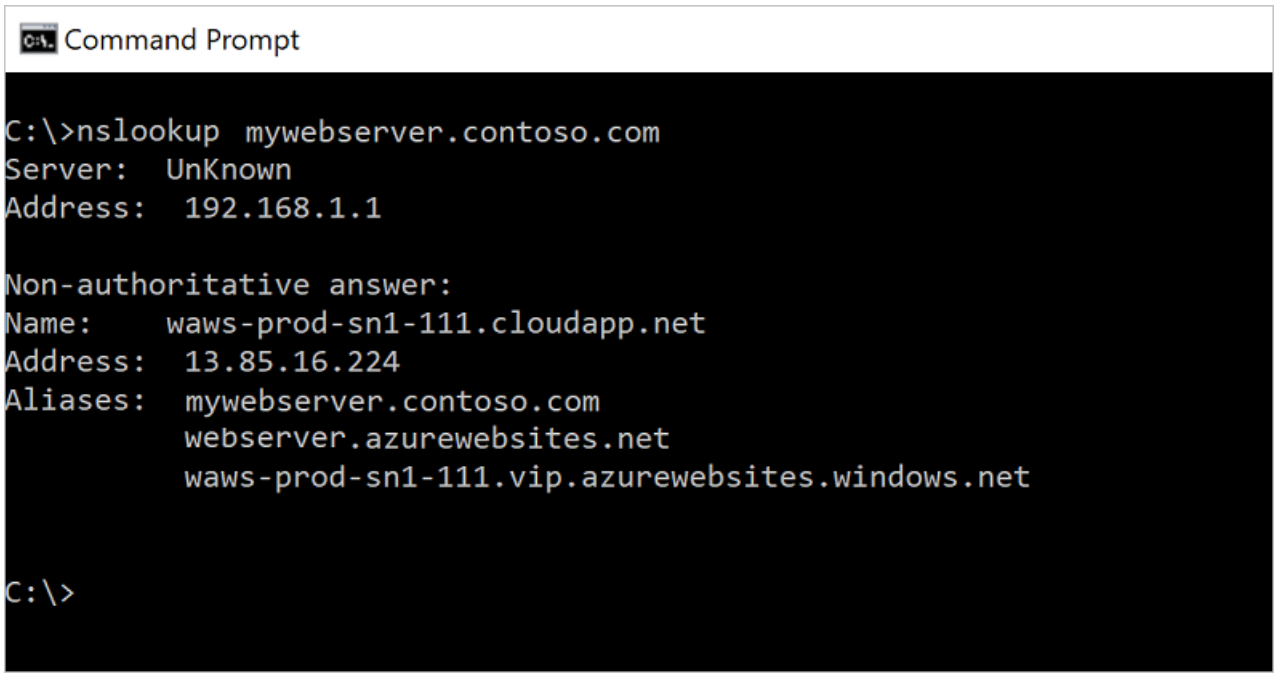
PROPERTY	VALUE	DESCRIPTION
Name	mywebserver	This value along with the domain name label is the FQDN for the custom domain name.
Type	CNAME	Use a CNAME record is using an alias. If the resource used an IP address, an A record would be used.
TTL	1	1 is used for 1 hour
TTL unit	Hours	Hours are used as the time measurement
Alias	webserver.azurewebsites.net	The DNS name you are creating the alias for, in this example it is the webserver.azurewebsites.net DNS name provided by default to the web app.



Navigate back to the app service that is configured for the custom domain name. Click **Custom domains**, then click **Hostnames**. To add the CNAME record you created, click **+ Add hostname**.



Once the process is complete, run **nslookup** to validate name resolution is working.



To learn more about mapping a custom domain to App Service, visit Map an existing custom DNS name to Azure Web Apps.

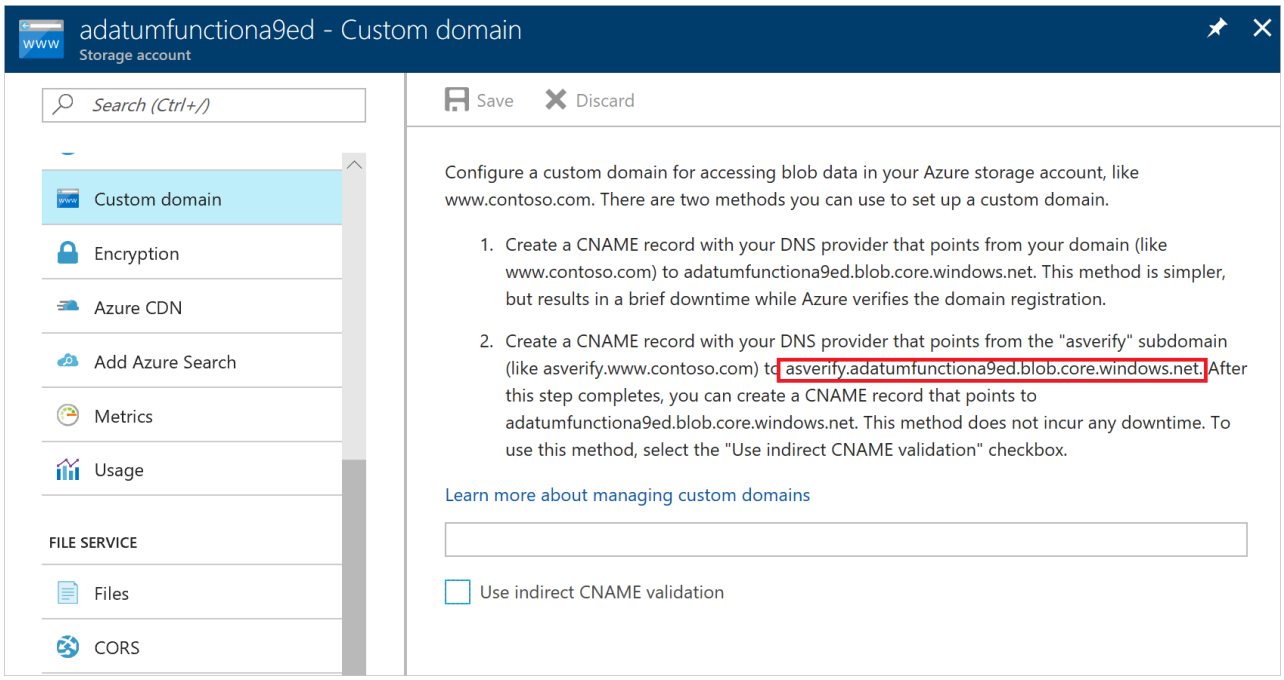
To learn how to migrate an active DNS name, see Migrate an active DNS name to Azure App Service.

If you need to purchase a custom domain, visit Buy a custom domain name for Azure Web Apps to learn more about App Service domains.

Blob storage

The following steps take you through configuring a CNAME record for a blob storage account using the asverify method. This method ensures there is no downtime.

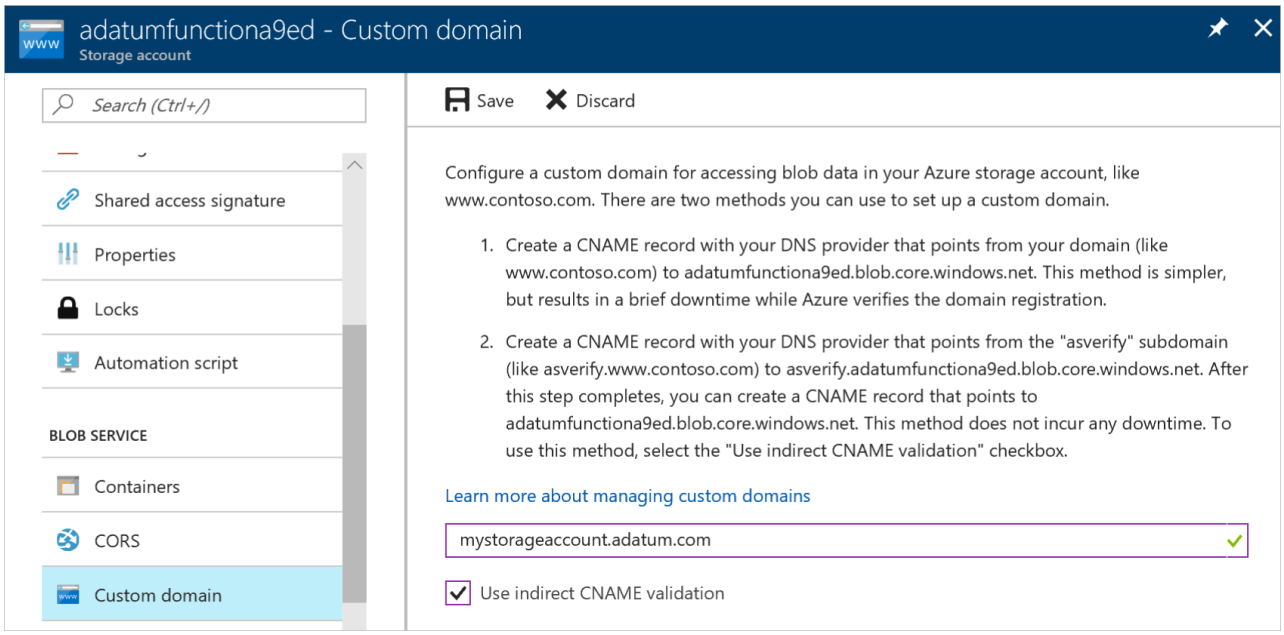
Navigate to **Storage** > **Storage Accounts**, select your storage account, and click **Custom domain**. Notate the FQDN under step 2, this value is used to create the first CNAME record



Navigate to your DNS Zone and click **+ Record set**. Fill out the following information on the **Add record set** blade and click **OK** to create it.

PROPERTY	VALUE	DESCRIPTION
Name	asverify.mystorageaccount	This value along with the domain name label is the FQDN for the custom domain name.
Type	CNAME	Use a CNAME record is using an alias.
TTL	1	1 is used for 1 hour
TTL unit	Hours	Hours are used as the time measurement
Alias	asverify.adatumfunctiona9ed.blob.core.windows.net	The DNS name you are creating the alias for, in this example it is the asverify.adatumfunctiona9ed.blob.core.windows.net DNS name provided by default to the storage account.

Navigate back to your storage account by clicking **Storage** > **Storage Accounts**, select your storage account and click **Custom domain**. Type in the alias you created without the asverify prefix in the text box, check **Use indirect CNAME validation, and click **Save**. Once this step is complete, return to your DNS zone and create a CNAME record without the asverify prefix. After that point, you are safe to delete the CNAME record with the cdnverify prefix.



Validate DNS resolution by running `nslookup`

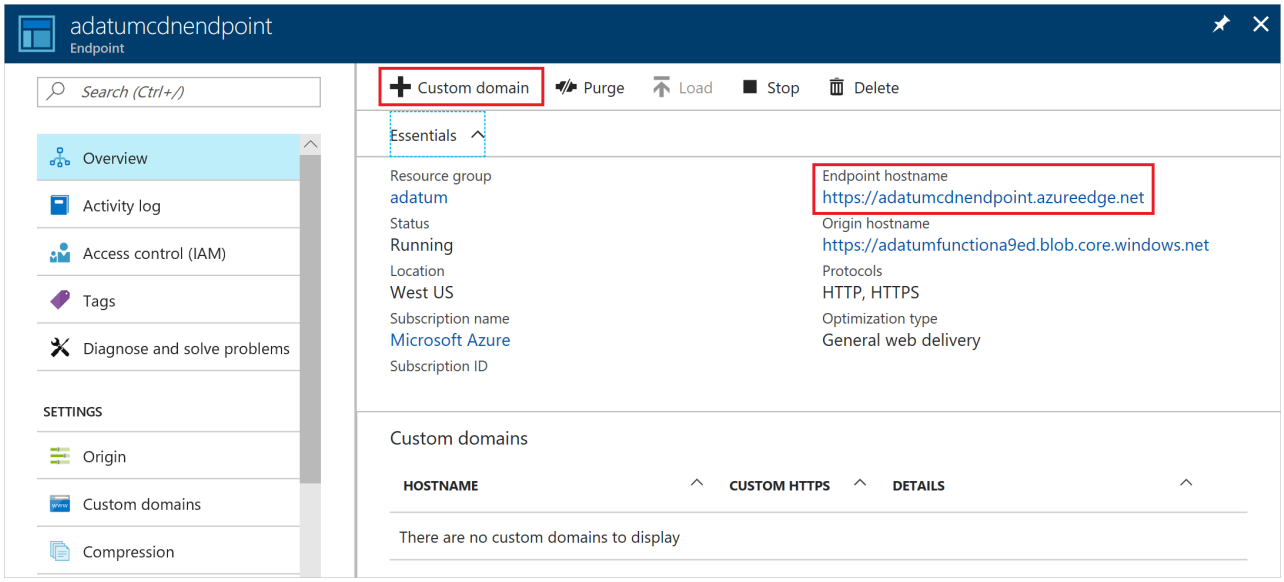
To learn more about mapping a custom domain to a blob storage endpoint visit Configure a custom domain name for your Blob storage endpoint

Azure CDN

The following steps take you through configuring a CNAME record for a CDN endpoint using the cdnverify method. This method ensures there is no downtime.

Navigate to **Networking** > **CDN Profiles**, select your CDN profile.

Select the endpoint you are working with and click **+ Custom domain**. Note the **Endpoint hostname** as this value is the record that the CNAME record points to.



Navigate to your DNS Zone and click **+ Record set**. Fill out the following information on the **Add record set** blade and click **OK** to create it.

PROPERTY	VALUE	DESCRIPTION

PROPERTY	VALUE	DESCRIPTION
Name	cdnverify.mycdnendpoint	This value along with the domain name label is the FQDN for the custom domain name.
Type	CNAME	Use a CNAME record is using an alias.
TTL	1	1 is used for 1 hour
TTL unit	Hours	Hours are used as the time measurement
Alias	cdnverify.adatumcdnendpoint.azureedge.net	The DNS name you are creating the alias for, in this example it is the cdnverify.adatumcdnendpoint.azureedge.net DNS name provided by default to the storage account.

Navigate back to your CDN endpoint by clicking **Networking** > **CDN Profiles**, and select your CDN profile. Click **+ Custom domain** and enter your CNAME record alias without the cdnverify prefix and click **Add**.

Once this step is complete, return to your DNS zone and create a CNAME record without the cdnverify prefix. After that point, you are safe to delete the CNAME record with the cdnverify prefix. For more information on CDN and how to configure a custom domain without the intermediate registration step visit Map Azure CDN content to a custom domain.

Next steps

Learn how to configure reverse DNS for services hosted in Azure.

How to protect private DNS zones and records

2/19/2020 • 9 minutes to read • Edit Online

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.

Private DNS zones and records are critical resources. Deleting a DNS zone or a single DNS record can result in a service outage. It's important that DNS zones and records are protected against unauthorized or accidental changes.

This article explains how Azure DNS enables you to protect your private DNS zones and records against such changes. We apply two powerful securities features provided by Azure Resource Manager: role-based access control and resource locks.

Role-based access control

Azure Role-Based Access Control (RBAC) enables fine-grained access management for Azure users, groups, and resources. With RBAC, you can grant the level of access that users need. For more information about how RBAC helps you manage access, see What is Role-Based Access Control.

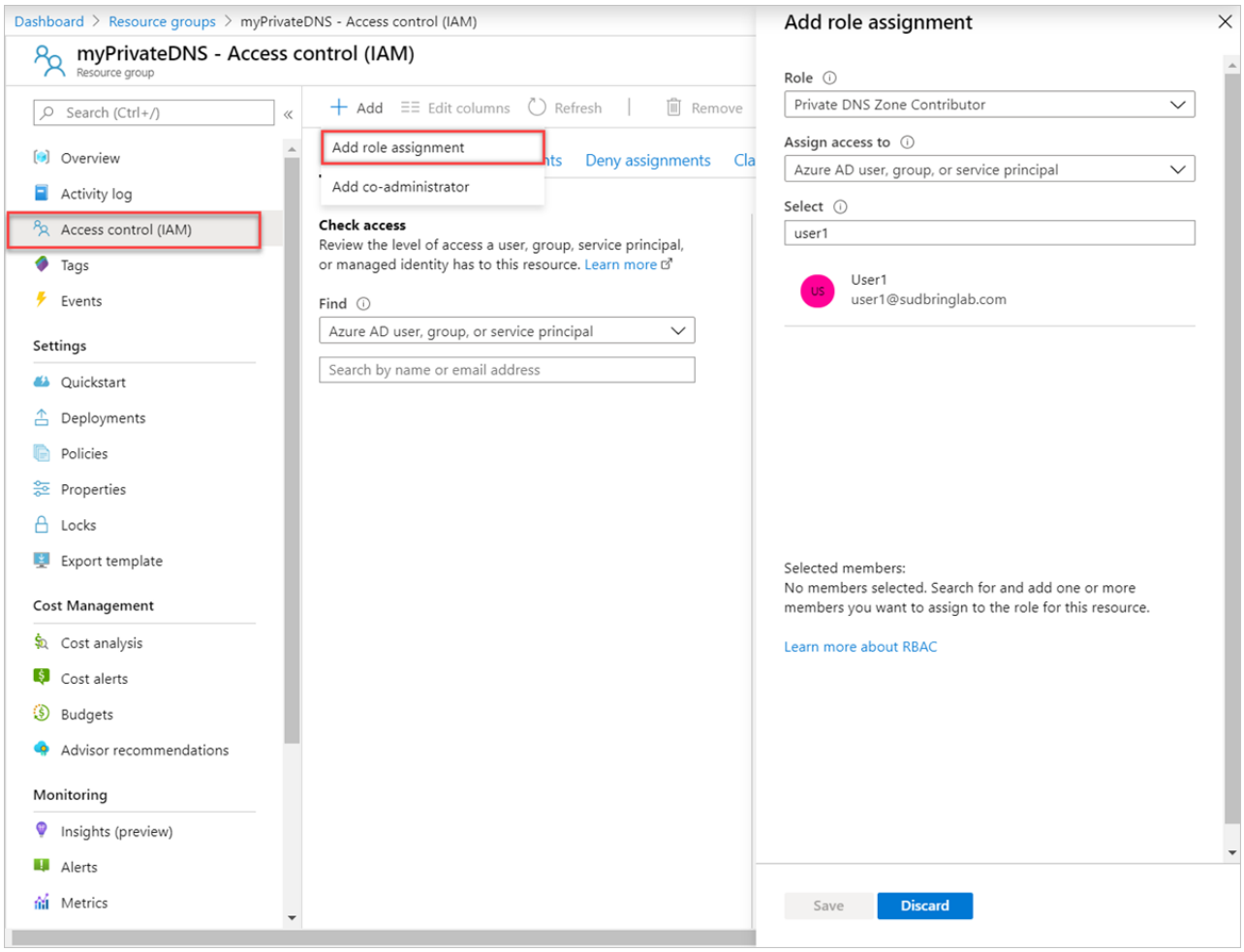
The Private DNS Zone Contributor role

The Private DNS Zone Contributor role is a built-in role for managing private DNS resources. This role applied to a user or group enables them to manage private DNS resources.

The resource group *myPrivateDNS* contains five zones for Contoso Corporation. Granting the DNS administrator Private DNS Zone Contributor permissions to that resource group, enables full control over those DNS zones. It avoids granting unnecessary permissions. The DNS administrator can't create or stop virtual machines.

The simplest way to assign RBAC permissions is via the Azure portal.

Open **Access control (IAM)** for the resource group, select **Add**, then select the **Private DNS Zone Contributor** role. Select the required users or groups to grant permissions.



Permissions can also be granted using Azure PowerShell:

```
# Grant 'Private DNS Zone Contributor' permissions to all zones in a resource group

$rsg = "<resource group name>"
$usr = "<user email address>"
$rol = "Private DNS Zone Contributor"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -ResourceGroupName $rsg
```

The equivalent command is also available via the Azure CLI:

```
# Grant 'Private DNS Zone Contributor' permissions to all zones in a resource group

az role assignment create \
--assignee "<user email address>" \
--role "Private DNS Zone Contributor" \
--resource-group "<resource group name>"
```

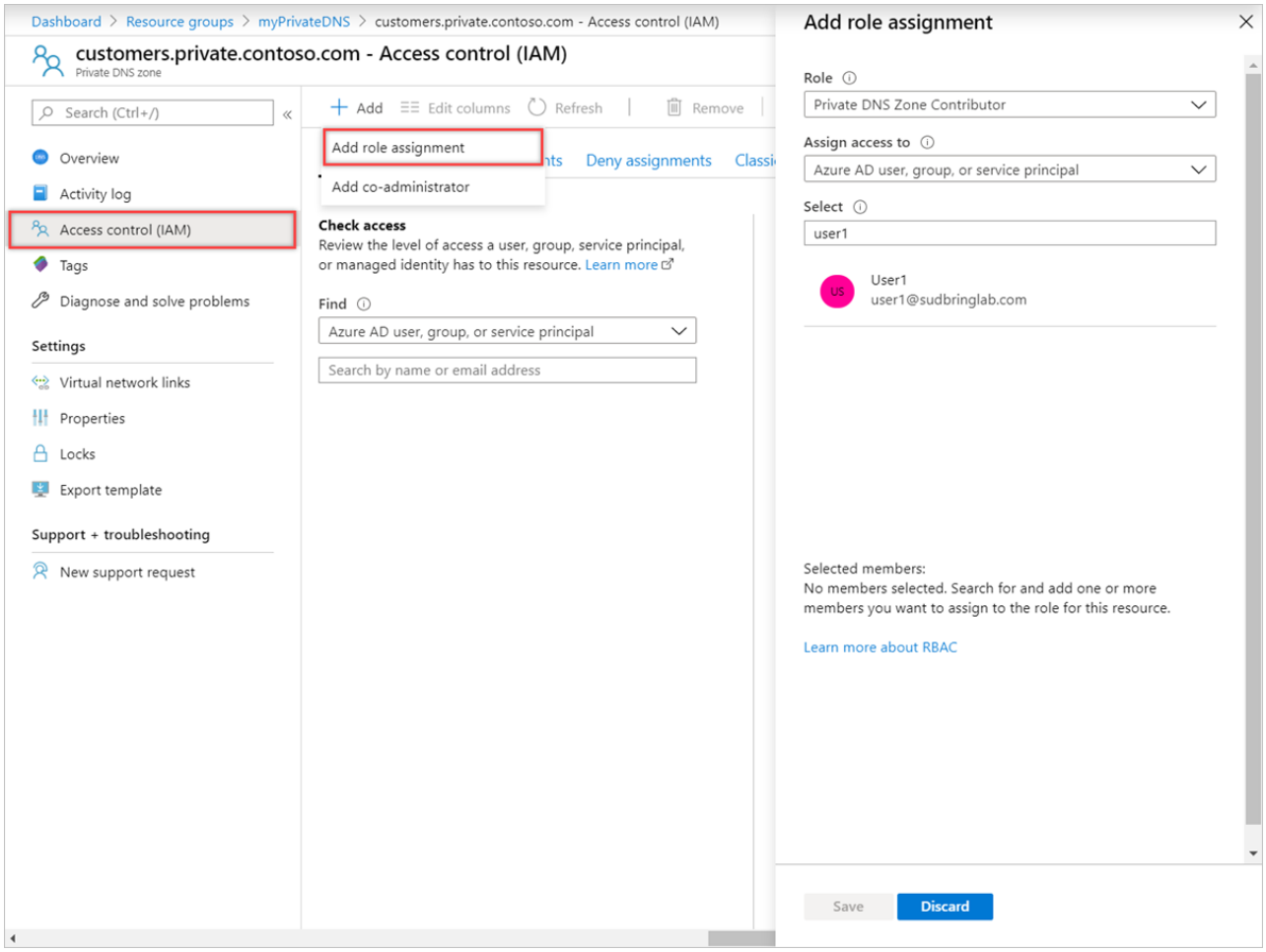
Private Zone level RBAC

Azure RBAC rules can be applied to a subscription, a resource group or to an individual resource. That resource can be an individual DNS zone, or an individual record set.

For example, the resource group *myPrivateDNS* contains the zone *private.contoso.com* and a subzone *customers.private.contoso.com*. CNAME records are created for each customer account. The administrator account used to manage CNAME records is assigned permissions to create records in the *customers.private.contoso.com* zone. The account can manage *customers.private.contoso.com* only.

Zone-level RBAC permissions can be granted via the Azure portal. Open **Access control (IAM)** for the zone,

select **Add**, then select the **Private DNS Zone Contributor** role. Select the required users or groups to grant permissions.



Permissions can also be granted using Azure PowerShell:

```
# Grant 'Private DNS Zone Contributor' permissions to a specific zone

$rsg = "<resource group name>"
$usr = "<user email address>"
$zon = "<zone name>"
$rol = "Private DNS Zone Contributor"
$rsc = "Microsoft.Network/privateDnsZones"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -ResourceGroupName $rsg -ResourceName $zon -
ResourceType $rsc
```

The equivalent command is also available via the Azure CLI:

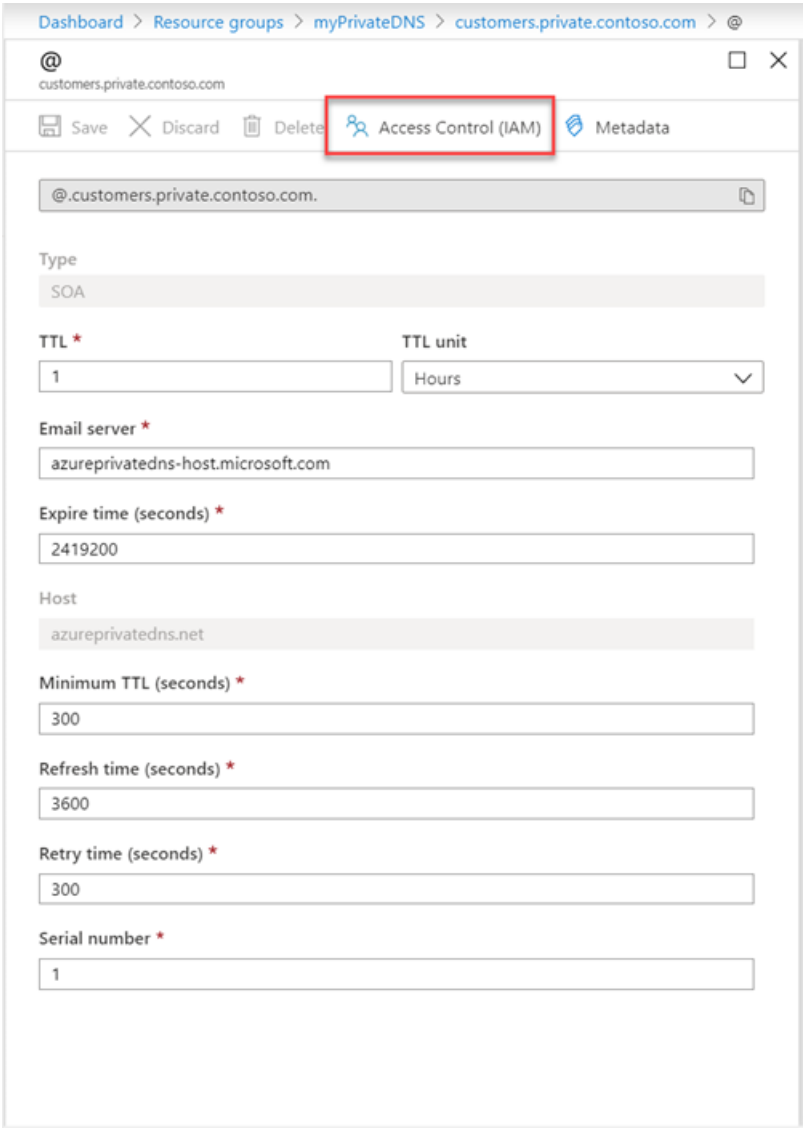
```
# Grant 'Private DNS Zone Contributor' permissions to a specific zone

az role assignment create \
--assignee <user email address> \
--role "Private DNS Zone Contributor" \
--scope "/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/privateDnsZones/<zone name>/"
```

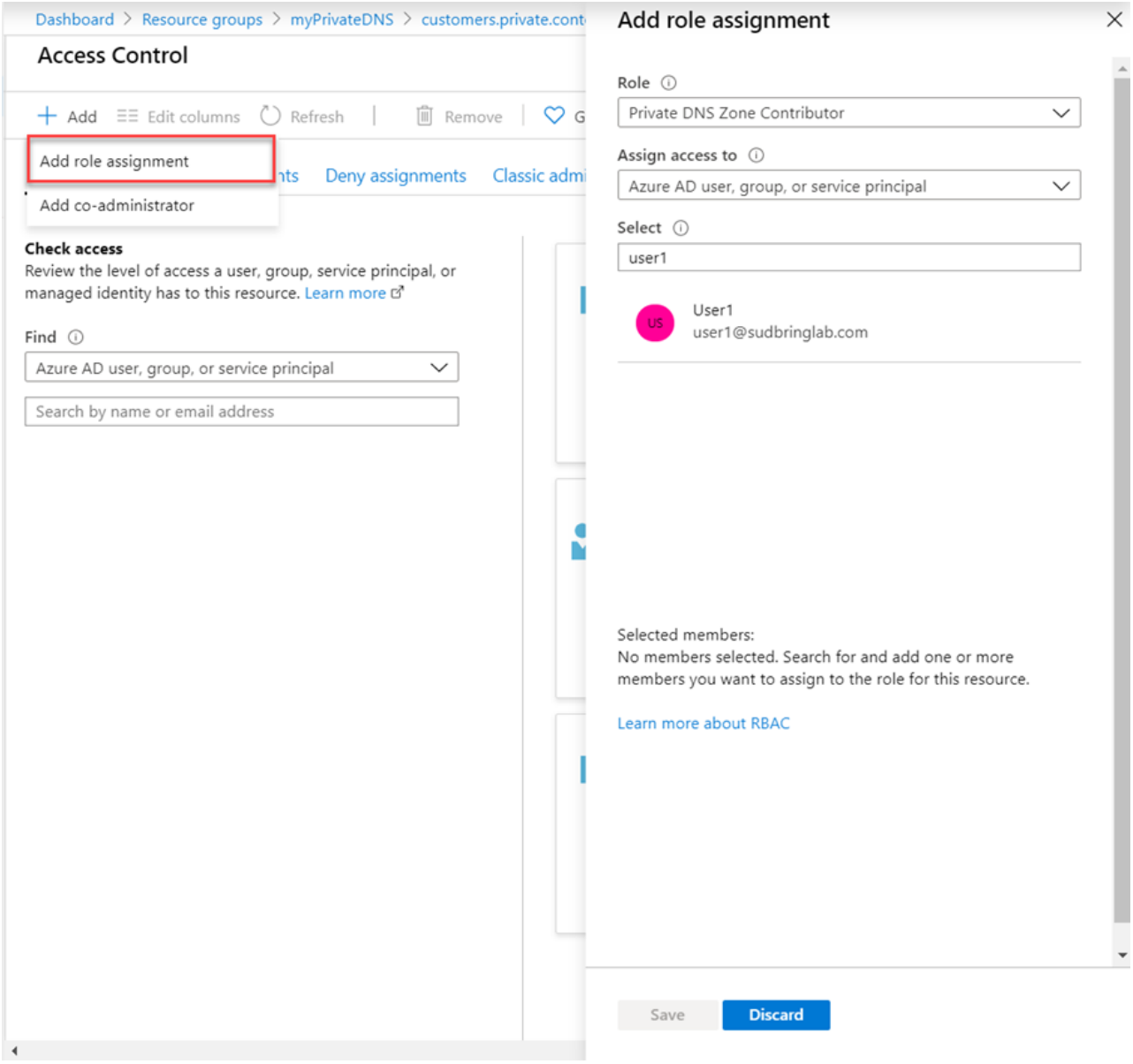
Record set level RBAC

Permissions are applied at the record set level. The user is granted control to entries they need and are unable to make any other changes.

Record-set level RBAC permissions can be configured via the Azure portal, using the **Access Control (IAM)**

button in the record set page:





Record-set level RBAC permissions can also be granted using Azure PowerShell:

```
# Grant permissions to a specific record set

$usr = "<user email address>"
$rol = "Private DNS Zone Contributor"
$sco =
"/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/privateDnsZones/<zone name>/<record type>/<record name>"

New-AzRoleAssignment -SignInName $usr -RoleDefinitionName $rol -Scope $sco
```

The equivalent command is also available via the Azure CLI:

```
# Grant permissions to a specific record set

az role assignment create \
--assignee "<user email address>" \
--role "Private DNS Zone Contributor" \
--scope "/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/privateDnsZones/<zone name>/<record type>/<record name>"
```

Custom roles

The built-in Private DNS Zone Contributor role enables full control over a DNS resource. It's possible to build your own custom Azure roles to provide finer-grained control.

The account that is used to manage CNAMEs is granted permission to manage CNAME records only. The account is unable to modify records of other types. The account is unable to do zone-level operations such as zone delete.

The following example shows a custom role definition for managing CNAME records only:

```
{
    "Name": "Private DNS CNAME Contributor",
    "Id": "",
    "IsCustom": true,
    "Description": "Can manage DNS CNAME records only.",
    "Actions": [
        "Microsoft.Network/privateDnsZones/CNAME/*",
        "Microsoft.Network/privateDNSZones/read",
        "Microsoft.Authorization/*/read",
        "Microsoft.Insights/alertRules/*",
        "Microsoft.ResourceHealth/availabilityStatuses/read",
        "Microsoft.Resources/deployments/*",
        "Microsoft.Resources/subscriptions/resourceGroups/read",
        "Microsoft.Support/*"
    ],
    "NotActions": [
    ],
    "AssignableScopes": [
        "/subscriptions/c276fc76-9cd4-44c9-99a7-4fd71546436e"
    ]
}
```

The Actions property defines the following DNS-specific permissions:

- `Microsoft.Network/privateDnsZones/CNAME/*` grants full control over CNAME records
- `Microsoft.Network/privateDNSZones/read` grants permission to read DNS private zones, but not to modify them, enabling you to see the zone in which the CNAME is being created.

> **NOTE**
> Using a custom RBAC role to prevent deleting record sets while still allowing them to be updated is not an effective control. It prevents record sets from being deleted, but it does not prevent them from being modified. Permitted modifications include adding and removing records from the record set, including removing all records to leave an empty record set. This has the same effect as deleting the record set from a DNS resolution viewpoint.

Custom role definitions can't currently be defined via the Azure portal. A custom role based on this role definition can be created using Azure PowerShell:

```
# Create new role definition based on input file

New-AzRoleDefinition -InputFile <file path>
```

It can also be created via the Azure CLI:

```
# Create new role definition based on input file

az role create -inputfile <file path>
```

The role can then be assigned in the same way as built-in roles, as described earlier in this article.

For more information on how to create, manage, and assign custom roles, see Custom Roles in Azure RBAC.

Resource locks

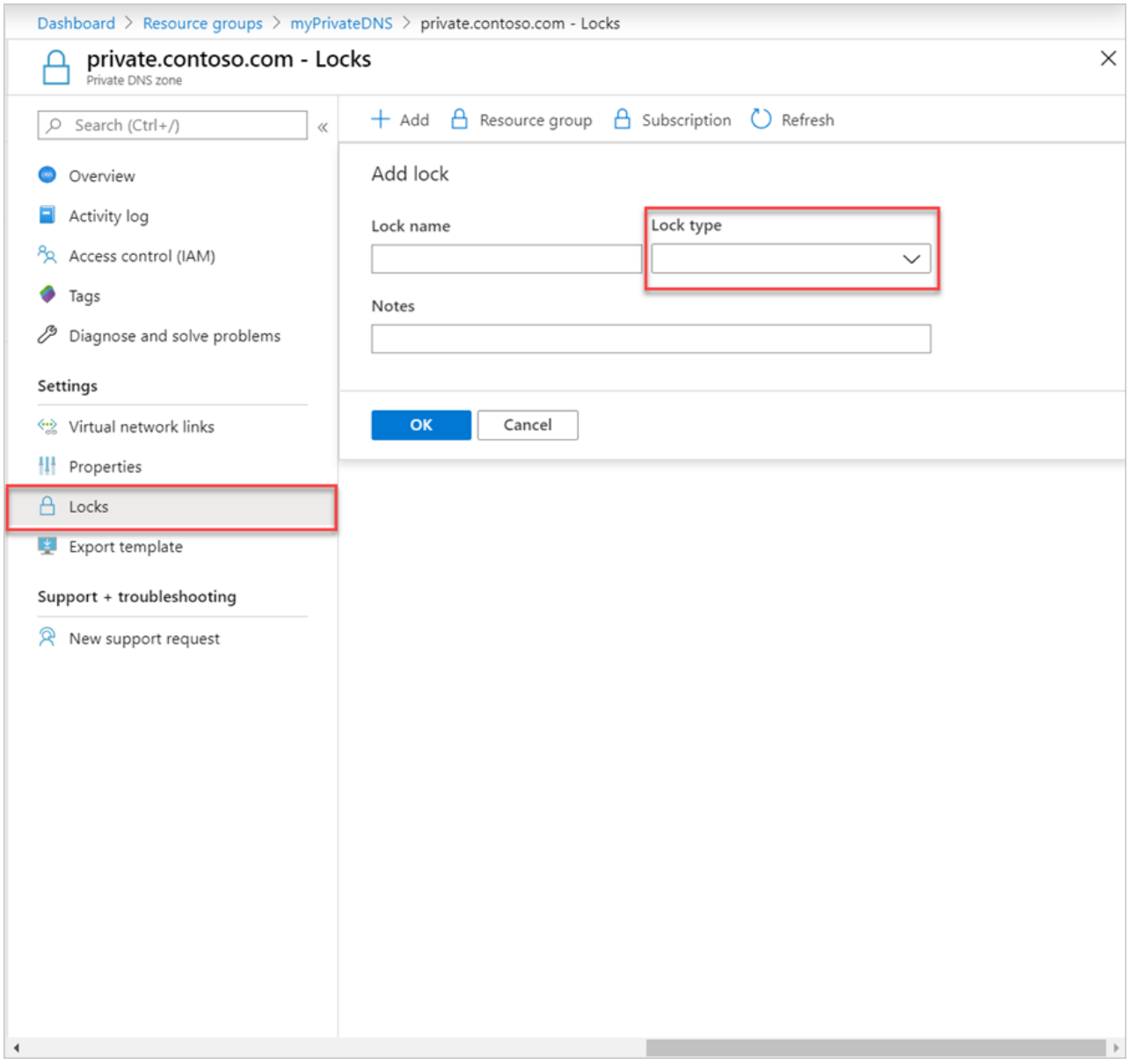
Azure Resource Manager supports another type of security control, the ability to lock resources. Resource locks are applied to the resource, and are effective across all users and roles. For more information, see Lock resources with Azure Resource Manager.

There are two types of resource lock: **CanNotDelete** and **ReadOnly**. These lock types can be applied either to a Private DNS zone, or to an individual record set. The following sections describe several common scenarios, and how to support them using resource locks.

Protecting against all changes

To prevent changes being made, apply a ReadOnly lock to the zone. This lock prevents new record sets from being created, and existing record sets from being modified or deleted.

Zone level resource locks can be created via the Azure portal. From the DNS zone page, select **Locks**, then select **+Add**:



Zone-level resource locks can also be created via Azure PowerShell:

```
# Lock a DNS zone

$lvl = "<lock level>"
$lnm = "<lock name>"
$rsc = "<zone name>"
$rty = "Microsoft.Network/privateDnsZones"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rsc -ResourceType $rty -ResourceGroupName $rsg
```

The equivalent command is also available via the Azure CLI:

```
# Lock a DNS zone

az lock create \
--lock-type "<lock level>" \
--name "<lock name>" \
--resource-name "<zone name>" \
--namespace "Microsoft.Network" \
--resource-type "privateDnsZones" \
--resource-group "<resource group name>"
```

Protecting individual records

To prevent an existing DNS record set against modification, apply a ReadOnly lock to the record set.

> **NOTE**
> Applying a CanNotDelete lock to a record set is not an effective control. It prevents the record set from being deleted, but it does not prevent it from being modified. Permitted modifications include adding and removing records from the record set, including removing all records to leave an empty record set. This has the same effect as deleting the record set from a DNS resolution viewpoint.

Record set level resource locks can currently only be configured using Azure PowerShell. They aren't supported in the Azure portal or Azure CLI.

Azure PowerShell

```
# Lock a DNS record set

$lvl = "<lock level>"
$lnm = "<lock name>"
$rnm = "<zone name>/<record set name>"
$rty = "Microsoft.Network/privateDnsZones"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rnm -ResourceType $rty -ResourceGroupName $rsg
```

Protecting against zone deletion

When a zone is deleted in Azure DNS, all record sets in the zone are deleted. This operation can't be undone. Accidentally deleting a critical zone has the potential to have a significant business impact. It's important to protect against accidental zone deletion.

Applying a CanNotDelete lock to a zone prevents the zone from being deleted. Locks are inherited by child resources. A lock prevents any record sets in the zone from being deleted. As described in the note above, it's ineffective since records can still be removed from the existing record sets.

As an alternative, apply a CanNotDelete lock to a record set in the zone, such as the SOA record set. The zone isn't

deleted without also deleting the record sets. This lock protects against zone deletion, while still allowing record sets within the zone to be modified freely. If an attempt is made to delete the zone, Azure Resource Manager detects this removal. The removal would also delete the SOA record set, Azure Resource Manager blocks the call because the SOA is locked. No record sets are deleted.

The following PowerShell command creates a CanNotDelete lock against the SOA record of the given zone:

```
# Protect against zone delete with CanNotDelete lock on the record set

$lvl = "CanNotDelete"
$lnm = "<lock name>"
$rnm = "<zone name>/@"
$rty = "Microsoft.Network/privateDnsZones/SOA"
$rsg = "<resource group name>"

New-AzResourceLock -LockLevel $lvl -LockName $lnm -ResourceName $rnm -ResourceType $rty -ResourceGroupName $rsg
```

Another option to prevent accidental zone deletion is by using a custom role. This role ensures the accounts used to manage your zones don't have zone delete permissions.

When you do need to delete a zone, you can enforce a two-step delete:

- First, grant zone delete permissions
- Second, grant permissions to delete the zone.

The custom role works for all zones accessed by those accounts. Accounts with zone delete permissions, such as the subscription owner, can still accidentally delete a zone.

It's possible to use both approaches - resource locks and custom roles - at the same time, as a defense-in-depth approach to DNS zone protection.

Next steps

- For more information about working with RBAC, see Get started with access management in the Azure portal.
- For more information about working with resource locks, see Lock resources with Azure Resource Manager.

Azure DNS troubleshooting guide

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

This article provides troubleshooting information for common Azure DNS questions.

If these steps don't resolve your issue, you can also search for or post your issue on our community support forum on MSDN. Or, you can open an Azure support request.

I can't create a DNS zone

To resolve common issues, try one or more of the following steps:

1. Review the Azure DNS audit logs to determine the failure reason.
2. Each DNS zone name must be unique within its resource group. That is, two DNS zones with the same name can't share a resource group. Try using a different zone name, or a different resource group.
3. You may see an error "You have reached or exceeded the maximum number of zones in subscription {subscription id}." Either use a different Azure subscription, delete some zones, or contact Azure Support to raise your subscription limit.
4. You may see an error "The zone '{zone name}' is not available." This error means that Azure DNS was unable to allocate name servers for this DNS zone. Try using a different zone name. Or, if you are the domain name owner you can contact Azure support to allocate name servers for you.

Recommended articles

- DNS zones and records
- Create a DNS zone

I can't create a DNS record

To resolve common issues, try one or more of the following steps:

1. Review the Azure DNS audit logs to determine the failure reason.
2. Does the record set exist already? Azure DNS manages records using record *sets*, which are the collection of records of the same name and the same type. If a record with the same name and type already exists, then to add another such record you should edit the existing record set.
3. Are you trying to create a record at the DNS zone apex (the 'root' of the zone)? If so, the DNS convention is to use the '@' character as the record name. Also note that the DNS standards don't permit CNAME records at the zone apex.
4. Do you have a CNAME conflict? The DNS standards don't allow a CNAME record with the same name as a record of any other type. If you have an existing CNAME, creating a record with the same name of a different type fails. Likewise, creating a CNAME fails if the name matches an existing record of a different type. Remove the conflict by removing the other record or choosing a different record name.
5. Have you reached the limit on the number of record sets permitted in a DNS zone? The current number of record sets and the maximum number of record sets are shown in the Azure portal, under the 'Properties' for the zone. If you've reached this limit, then either delete some record sets or contact Azure Support to raise your record set limit for this zone, then try again.

Recommended articles

- DNS zones and records
- Create a DNS zone

I can't resolve my DNS record

DNS name resolution is a multi-step process, which can fail for many reasons. The following steps help you investigate why DNS resolution is failing for a DNS record in a zone hosted in Azure DNS.

1. Confirm that the DNS records have been configured correctly in Azure DNS. Review the DNS records in the Azure portal, checking that the zone name, record name, and record type are correct.
2. Confirm that the DNS records resolve correctly on the Azure DNS name servers.
   - If you make DNS queries from your local PC, you may see cached results that don't reflect the current state of the name servers. Also, corporate networks often use DNS proxy servers, which prevent DNS queries from being directed to specific name servers. To avoid these problems, use a web-based name resolution service such as digwebinterface.
   - Be sure to specify the correct name servers for your DNS zone, as shown in the Azure portal.
   - Check that the DNS name is correct (you have to specify the fully qualified name, including the zone name) and the record type is correct
3. Confirm that the DNS domain name has been correctly delegated to the Azure DNS name servers. There are a many 3rd-party web sites that offer DNS delegation validation. This test is a *zone* delegation test, so you should only enter the DNS zone name and not the fully qualified record name.
4. Having completed the above, your DNS record should now resolve correctly. To verify, you can again use digwebinterface, this time using the default name server settings.

Recommended articles

- Delegate a domain to Azure DNS

How do I specify the 'service' and 'protocol' for an SRV record?

Azure DNS manages DNS records as record sets—the collection of records with the same name and the same type. For an SRV record set, the 'service' and 'protocol' need to be specified as part of the record set name. The other SRV parameters ('priority', 'weight', 'port' and 'target') are specified separately for each record in the record set.

Example SRV record names (service name 'sip', protocol 'tcp'):

- _sip._tcp (creates a record set at the zone apex)
- _sip._tcp.sipservice (creates a record set named 'sipservice')

Recommended articles

- DNS zones and records
- Create DNS record sets and records by using the Azure portal
- SRV record type (Wikipedia)

Next steps

- Learn about Azure DNS zones and records
- To start using Azure DNS, learn how to create a DNS zone and create DNS records.
- To migrate an existing DNS zone, learn how to import and export a DNS zone file.

Azure CLI examples for Azure DNS

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

The following table includes links to Azure CLI examples for Azure DNS.

Create a DNS zone and record	Creates a DNS zone and record for a domain name.

What is Azure Private Link?

2/28/2020 • 3 minutes to read • Edit Online

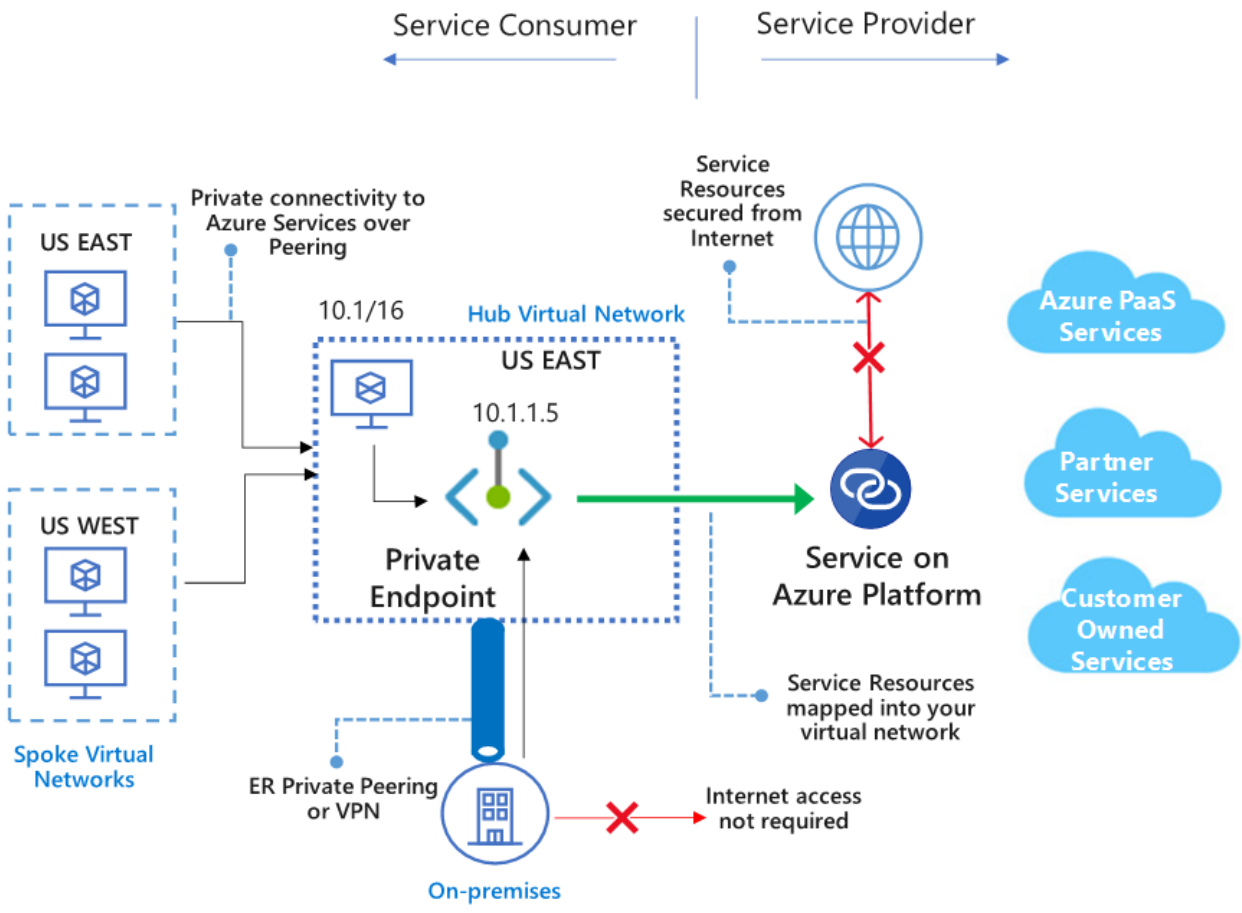
Azure Private Link enables you to access Azure PaaS Services such as:

- **Azure Storage**
- **Azure Cosmos DB**
- **Azure SQL Database**

Private Link enables access to hosted customer and partner services over a private endpoint in your virtual network.

Traffic between your virtual network and the service travels the Microsoft backbone network. Exposing your service to the public internet is no longer necessary. You can create your own private link service in your virtual network and deliver it to your customers. Setup and consumption using Azure Private Link is consistent across Azure PaaS, customer-owned, and shared partner services.

> **IMPORTANT**
>
> Azure Private Link is now generally available. Both Private Endpoint and Private Link service (service behind standard load balancer) are generally available. Different Azure PaaS will onboard to Azure Private Link at different schedules. Check availability section below for accurate status of Azure PaaS on Private Link. For known limitations, see Private Endpoint and Private Link Service.



Key benefits

Azure Private Link provides the following benefits:

- **Privately access services on the Azure platform**: Connect your virtual network to services in Azure without a public IP address at the source or destination. Service providers can render their services in their own virtual network and consumers can access those services in their local virtual network. The Private Link platform will handle the connectivity between the consumer and services over the Azure backbone network.
- **On-premises and peered networks**: Access services running in Azure from on-premises over ExpressRoute private peering, VPN tunnels, and peered virtual networks using private endpoints. There's no need to set up public peering or traverse the internet to reach the service. Private Link provides a secure way to migrate workloads to Azure.
- **Protection against data leakage**: A private endpoint is mapped to an instance of a PaaS resource instead of the entire service. Consumers can only connect to the specific resource. Access to any other resource in the service is blocked. This mechanism provides protection against data leakage risks.
- **Global reach**: Connect privately to services running in other regions. The consumer's virtual network could be in region A and it can connect to services behind Private Link in region B.
- **Extend to your own services**: Enable the same experience and functionality to render your service privately to consumers in Azure. By placing your service behind a standard Azure Load Balancer, you can enable it for Private Link. The consumer can then connect directly to your service using a private endpoint in their own virtual network. You can manage the connection requests using an approval call flow. Azure Private Link works for consumers and services belonging to different Azure Active Directory tenants.

Availability

The following table lists the Private Link services and the regions where they're available.

SCENARIO	SUPPORTED SERVICES	AVAILABLE REGIONS	STATUS
Private Link for customer-owned services	Private Link services behind standard Azure Load Balancer	All public regions	GA Learn more
Private Link for Azure PaaS services	Azure Storage	All public regions	Preview Learn more
	Azure Data Lake Storage Gen2	All public regions	Preview Learn more
	Azure SQL Database	All public regions	Preview Learn more
	Azure SQL Data Warehouse	All public regions	Preview Learn more
	Azure Cosmos DB	West Central US, WestUS, North Central US	Preview Learn more
	Azure Database for PostgreSQL - Single server	All public regions	Preview Learn more
	Azure Database for MySQL	All public regions	Preview Learn more

SCENARIO	SUPPORTED SERVICES	AVAILABLE REGIONS	STATUS
	Azure Database for MariaDB	All public regions	Preview Learn more
	Azure Key Vault	All public regions	Preview Learn more

For the most up-to-date notifications, check the Azure Virtual Network updates page.

Logging and monitoring

Azure Private Link has integration with Azure Monitor. This combination allows:

- Archival of logs to a storage account.
- Streaming of events to your Event Hub.
- Azure Monitor logging.

You can access the following information on Azure Monitor:

- **Private endpoint**:
  - Data processed by the Private Endpoint (IN/OUT)
- **Private Link service**:
  - Data processed by the Private Link service (IN/OUT)
  - NAT port availability

Pricing

For pricing details, see Azure Private Link pricing.

FAQs

For FAQs, see Azure Private Link FAQs.

Limits

For limits, see Azure Private Link limits.

Service Level Agreement

For SLA, see SLA for Azure Private Link.

Next steps

- Quickstart: Create a Private Endpoint using Azure portal
- Quickstart: Create a Private Link service by using the Azure portal

Migrating legacy Azure DNS private zones to new resource model

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

During public preview, private DNS zones were created using "dnszones" resource with "zoneType" property set to "Private". Such zones will not be supported after December 31, 2019 and must be migrated to GA resource model which makes use of "privateDnsZones" resource type instead of "dnszones". The migration process is simple, and we've provided a PowerShell script to automate this process. This guide provides step by step instruction to migrate your Azure DNS private zones to the new resource model.

To find out the dnszones resources that require migration; execute the below command in Azure CLI.

```
az account set --subscription <SubscriptionId>
az network dns zone list --query "[?zoneType=='Private']"
```

Prerequisites

Make sure you have installed latest version of Azure PowerShell. For more information on Azure PowerShell (Az) and how to install it visit https://docs.microsoft.com/powershell/azure/new-azureps-module-az

Make sure that you've Az.PrivateDns module for the Azure PowerShell installed. To install this module, open an elevated PowerShell window (Administrative mode) and enter following command

```
Install-Module -Name Az.PrivateDns
```

IMPORTANT

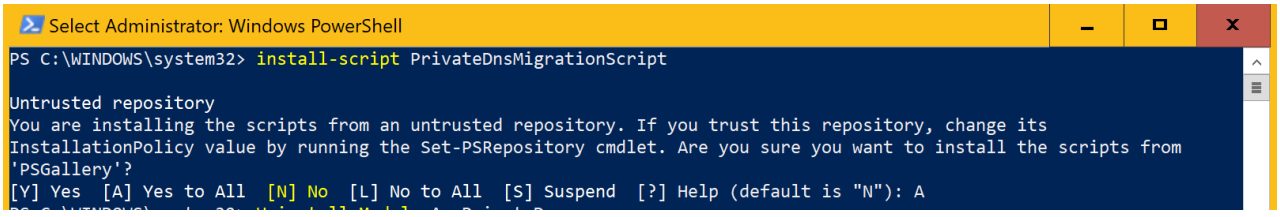
The migration process is fully automated and isn't expected to cause any downtime. However, if you're using Azure DNS private zones (preview) in a critical production environment you should execute the following migration process during a planned maintenance time window. Make sure that you don't modify the configuration or record-sets of a private DNS zones while you're running the migration script.

Installing the script

Open an elevated PowerShell window (Administrative mode) and run following command

```
install-script PrivateDnsMigrationScript
```

Enter "A" when prompted to install the script



You can also manually obtain the latest version of PowerShell script at

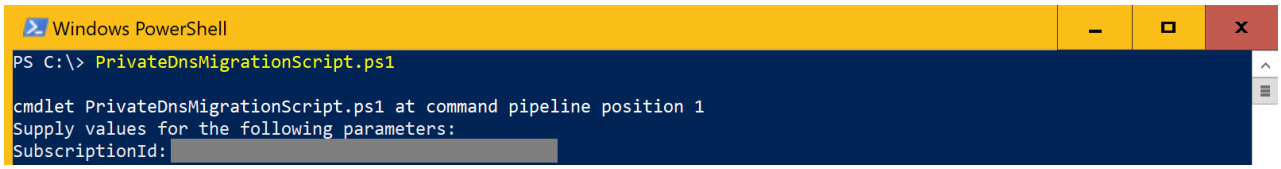
https://www.powershellgallery.com/packages/PrivateDnsMigrationScript

> **IMPORTANT**
> The migration script must not be run in Azure cloud shell and must be executed in a VM or local machine connected to internet.

Running the script

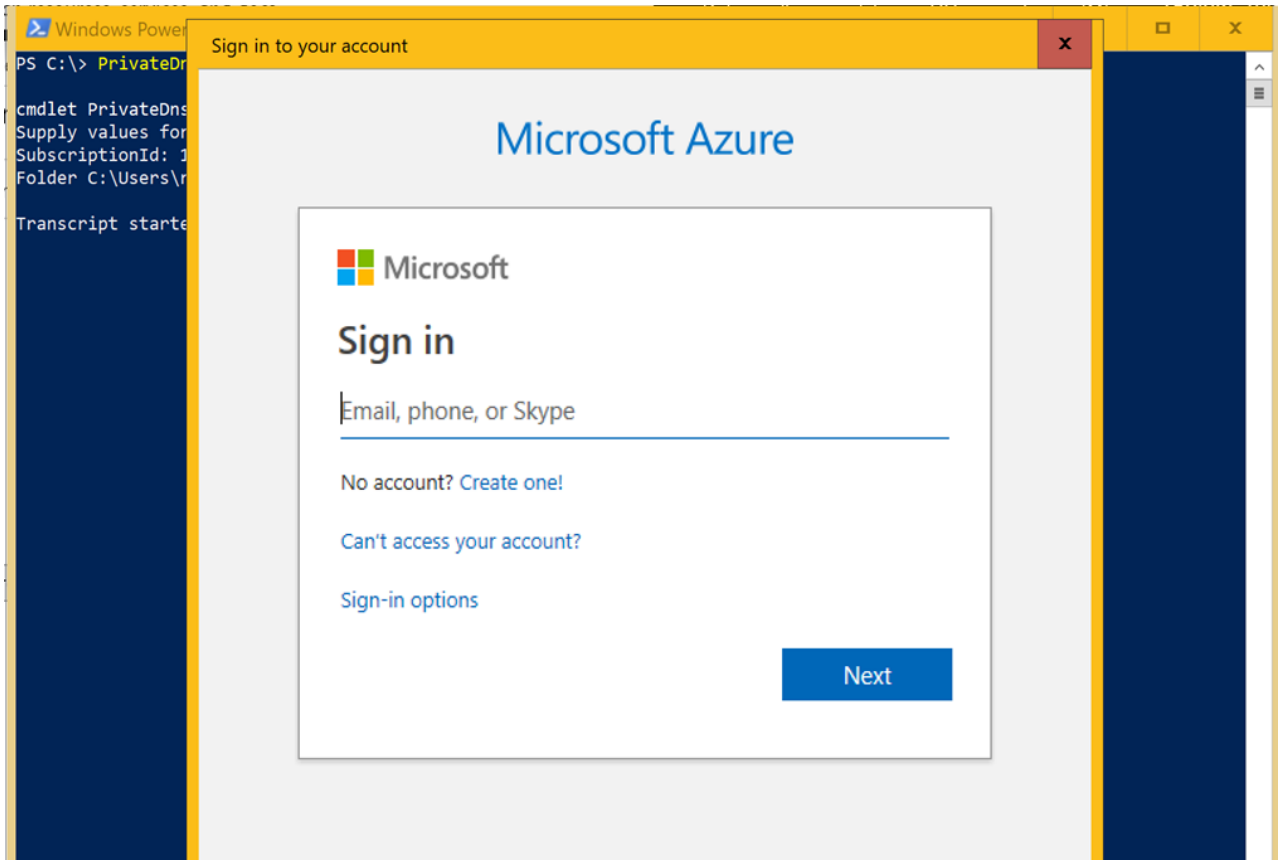
Execute following command to run the script

```
PrivateDnsMigrationScript.ps1
```



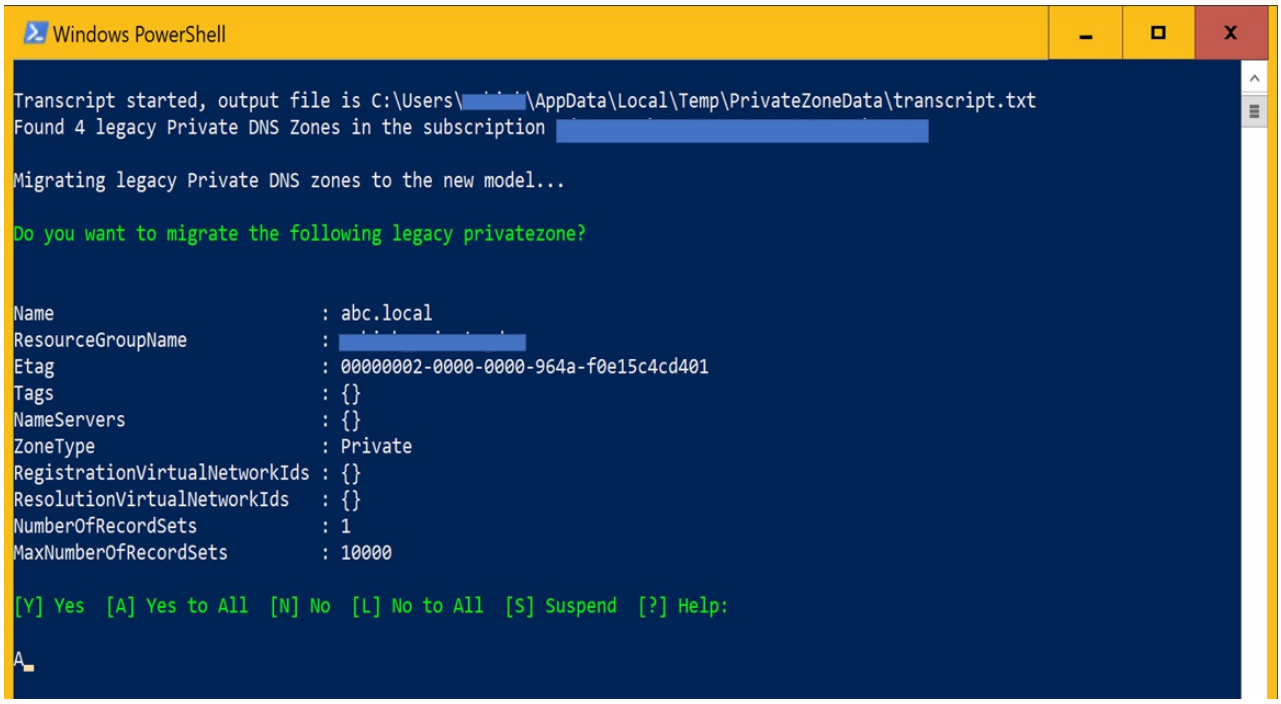
Enter the subscription ID and sign-in to Azure

You'll be prompted to enter subscription ID containing the private DNS zones that you intend to migrate. You'll be asked to sign-in to your Azure account. Complete the sign-in so that script can access the private DNS zone resources in the subscription.



Select the DNS zones you want to migrate

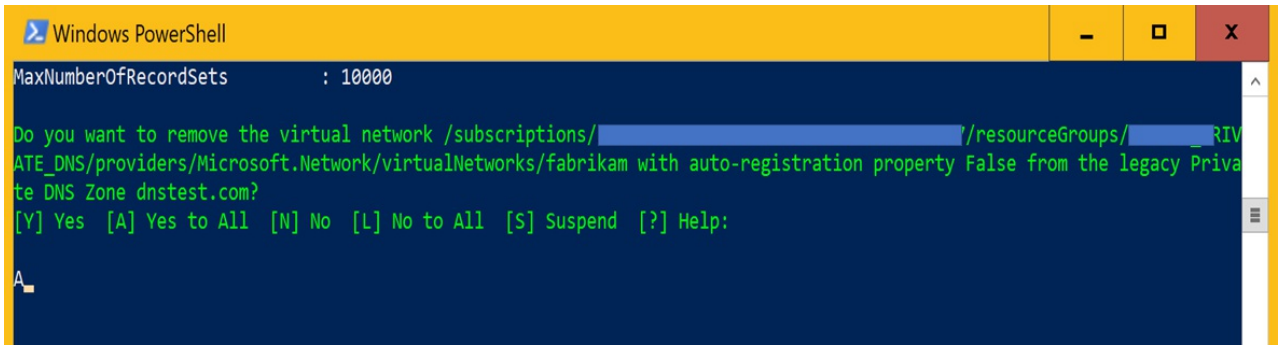
The script with get the list of all private DNS zones in the subscription and prompt you to confirm which ones you want to migrate. Enter "A" to migrate all private DNS zones. Once you execute this step, the script will create new private DNS zones using new resource model and copy the data into the new DSN zone. This step will not alter your existing private DNS zones in anyway.



Switching DNS resolution to the new DNS zones

Once the zones and records have been copied to the new resource model, the script will prompt you to switch the DNS resolution to new DNS zones. This step removes the association between legacy private DNS zones and your virtual networks. When the legacy zone is unlinked from the virtual networks, the new DNS zones created in above step would automatically take over the DNS resolution for those virtual networks.

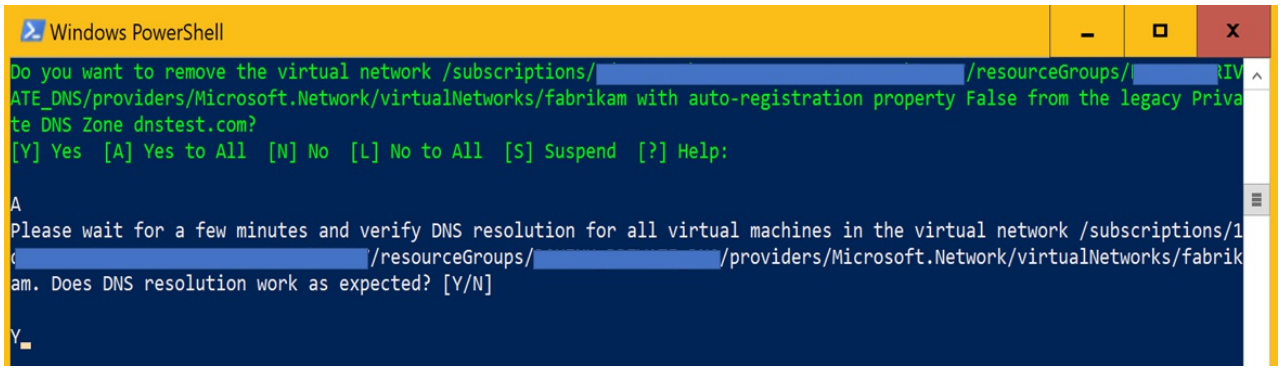
Select 'A' to switch the DNS resolution for all virtual networks.



Verify the DNS resolution

Before proceeding further, verify that DNS resolution on your DNS zones is working as expected. You can sign-in to your azure VMs and issue nslookup query against the migrated zones to verify that DNS resolution is working.

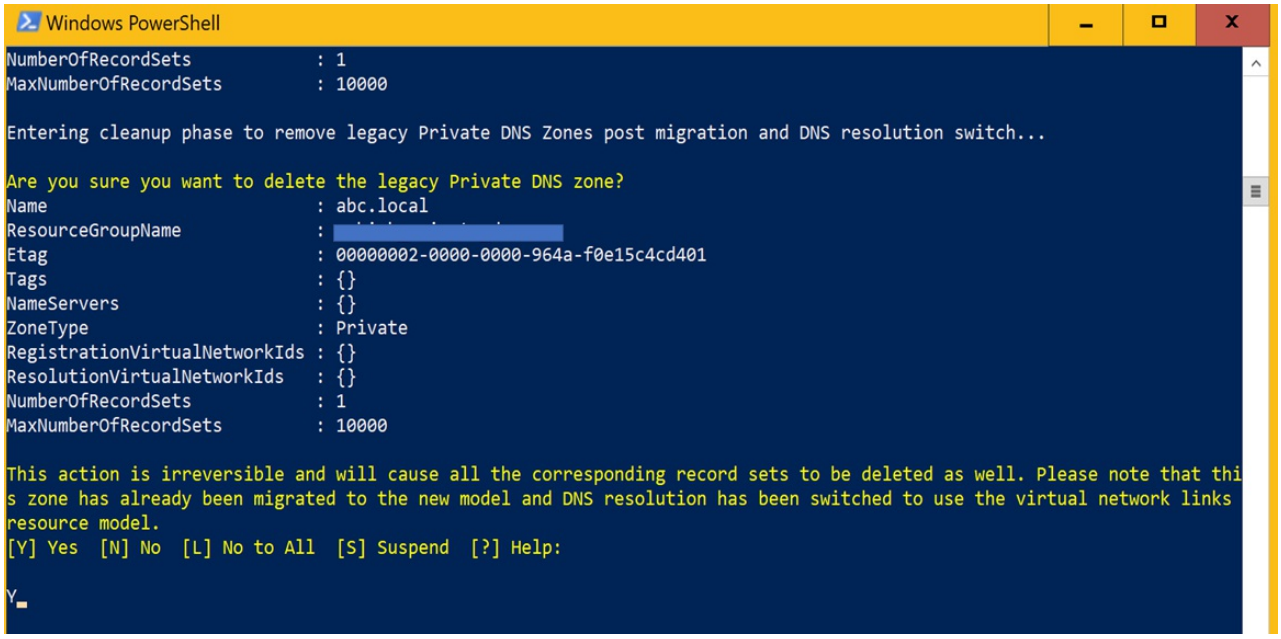
If you find that DNS queries aren't resolving, wait for a few minutes and retry the queries. If DNS queries are working as expected, enter 'Y' when script prompts you to remove the virtual network from the private DNS zone.



> **IMPORTANT**
>
> If because of any reason DNS resolution against the migrated zones isn't working as expected, enter 'N' in above step and script will switch the DNS resolution back to legacy zones. Create a support ticket and we can help you with migration of your DNS zones.

Cleanup

This step will delete the legacy DNS zones and should be executed only after you've verified that DNS resolution is working as expected. You'll be prompted to delete each private DNS zone. Enter 'Y' at every prompt after verifying that DNS resolution for that zones is working properly.



Update Your automation

If you're using automation including templates, PowerShell scripts or custom code developed using SDK, you must update your automation to use the new resource model for the private DNS zones. Below are the links to new private DNS CLI/PS/SDK documentation.

- Azure DNS private zones REST API
- Azure DNS private zones CLI
- Azure DNS private zones PowerShell
- Azure DNS private zones SDK

Need further help

Create a support ticket if you need further help with the migration process or because of any reason the above

listed steps don't work for you. Include the transcript file generated by the PowerShell script with your support ticket.

Next steps

- Learn how to create a private zone in Azure DNS using Azure PowerShell or Azure CLI.
- Read about some common private zone scenarios that can be realized with private zones in Azure DNS.
- For common questions and answers about private zones in Azure DNS, including specific behavior you can expect for certain kinds of operations, see Private DNS FAQ.
- Learn about DNS zones and records by visiting DNS zones and records overview.
- Learn about some of the other key networking capabilities of Azure.