Table of Contents

Azure DNS Overview

1/17/2017 • 1 min to read • Edit on GitHub

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

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

The Azure DNS service is based on Azure Resource Manager. As such, it benefits from Resource Manager features such as role-based access control, audit logs, and resource locking. Your domains and records can be managed via the Azure portal, Azure PowerShell cmdlets, and the cross-platform Azure CLI. Applications requiring automatic DNS management can integrate with the service via the REST API and SDKs.

Azure DNS does not currently support purchasing of domain names. If you want to purchase domains, 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.

Next steps

Create a DNS zone

DNS zones and records

1/17/2017 • 10 min to read • Edit on GitHub

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, 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 re-queried. 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'.

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

An NS record set is created automatically at the apex of each zone (name = '@'), and is deleted automatically when the zone is deleted (it cannot be deleted separately). You can modify the TTL of this record set, but you cannot modify the records, which are pre-configured to refer to the Azure DNS name servers assigned to the zone.

You can create and delete other NS records within the zone, other than at the zone apex. This allows you to configure child zones (see Delegating sub-domains in Azure DNS.)

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.

SPF records

Sender Policy Framework (SPF) records are used to specifying which email servers are permitted to send email on behalf of a given domain name. Correct configuration of SPF records is important to prevent recipients 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 permitted the use of the TXT record type to specify SPF records. This ambiguity led to confusion, which was resolved by RFC 7208. This states that SPF records should only be created using the TXT record type, and that the SPF record type is deprecated.

SPF records are supported by Azure DNS and should be created using the TXT record type. The obsolete SPF record type is not supported. When importing a DNS zone file, any SPF records using 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	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:

RESOURCE	DEFAULT LIMIT
Zones per subscription	100 [1]
Record sets per zone	5000 [1]
Records per record set	20

[1] Contact Azure Support in case you need to increase these limits.

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.

Create a DNS zone in the Azure portal

1/17/2017 • 3 min to read • Edit on GitHub

This article walks you through the steps to create a DNS zone using the Azure portal. You can also create a DNS zone using PowerShell or 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.

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).

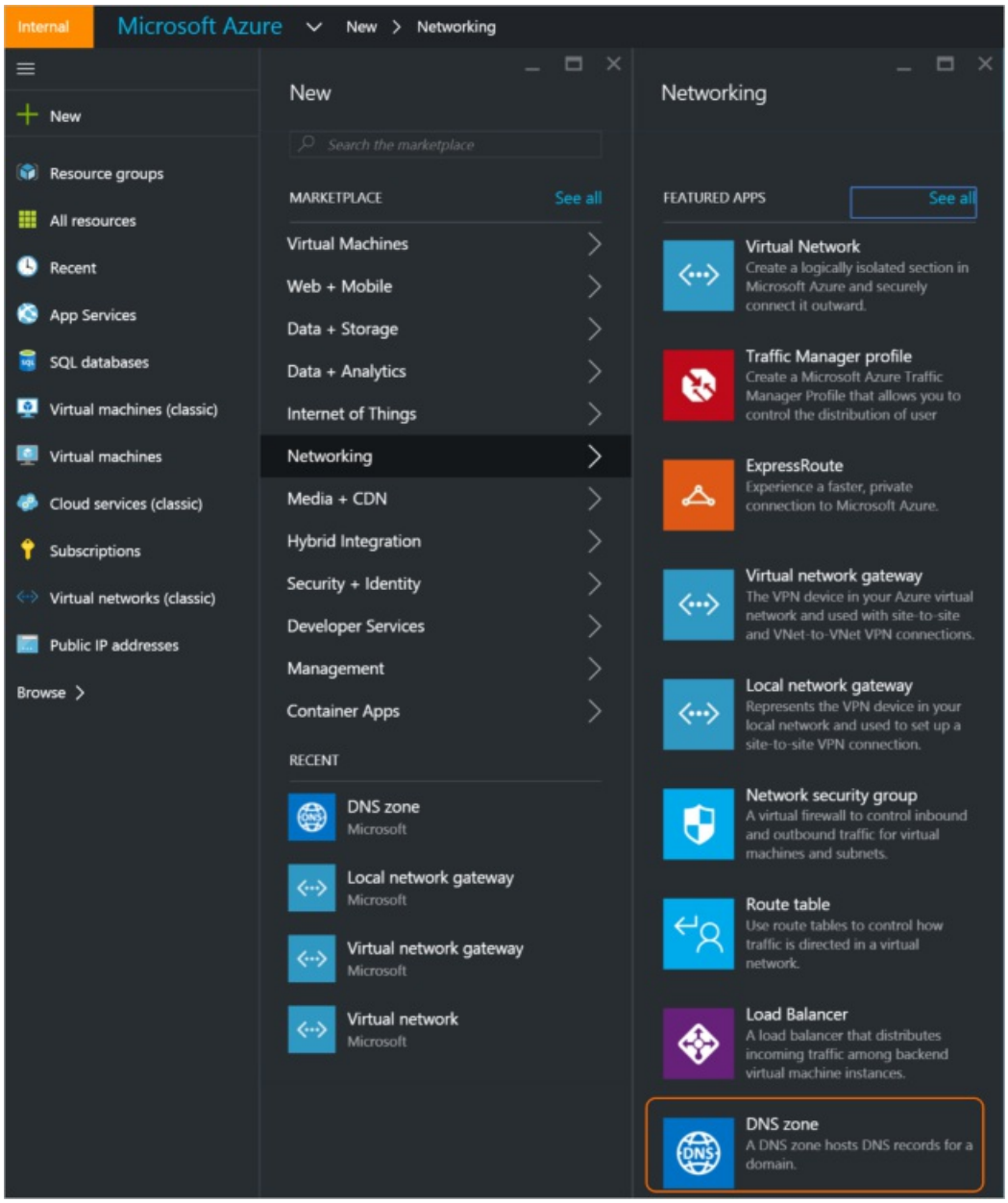
When creating a DNS zone in Azure DNS:

- 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.

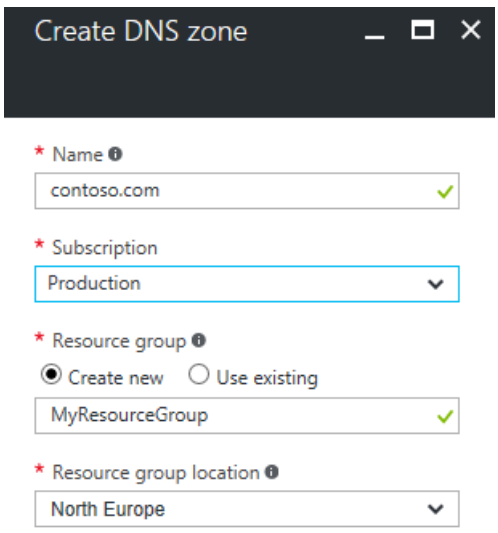
> **NOTE**
>
> You do not have to own a domain name to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to configure the Azure DNS name servers as the correct name servers for the domain name with the domain name registrar.
>
> For more information, see Delegate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

Create a DNS zone

1. Sign in to the Azure portal
2. On the Hub menu, click and click **New > Networking >** and then click **DNS zone** to open the Create DNS zone blade.

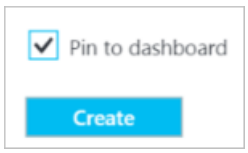


3. On the **Create DNS zone** blade, Name your DNS zone. For example, *contoso.com*.



4. Next, specify the resource group that you want to use. You can either create a new resource group, or select one that already exists. If you choose to create a new resource group, use the **Location** dropdown to specify the location of the resource group. Note that this setting 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.

5. You can leave the **Pin to dashboard** checkbox selected if you want to easily locate your new zone on your dashboard. Then click **Create**.



6. After you click Create, you'll see your new zone being configured on the dashboard.

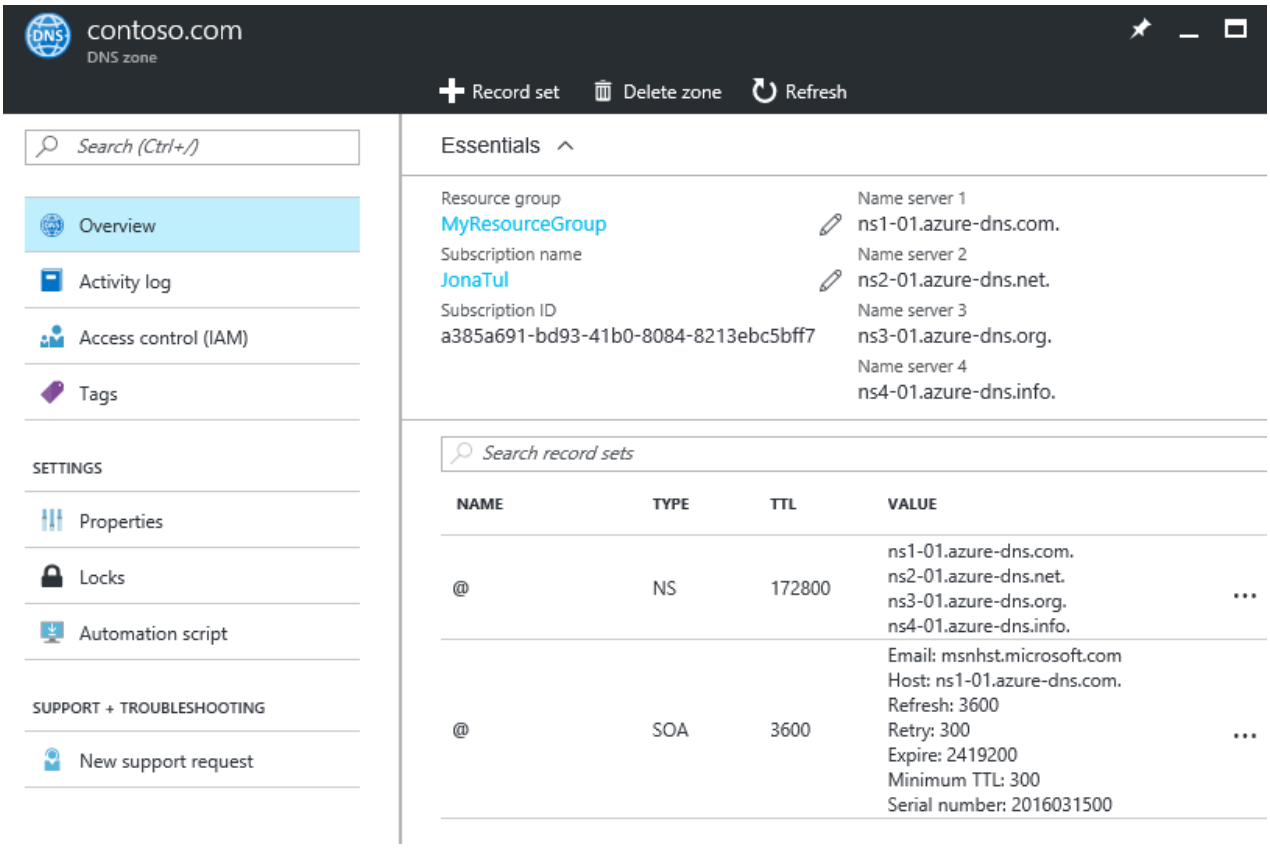
7. When your new zone has been created, the blade for your new zone will open on the dashboard.

View records

Creating a DNS zone also creates the following records:

- The "Start of Authority" (SOA) record. The SOA is present at the root of every DNS zone.
- The authoritative name server (NS) records. These show which name servers are hosting the zone. Azure DNS uses a pool of name servers, and so different name servers may be assigned to different zones in Azure DNS. See Delegate a domain to Azure DNS for more information.

In the lower part of the DNS Zone blade, you can see the record sets for the DNS zone.

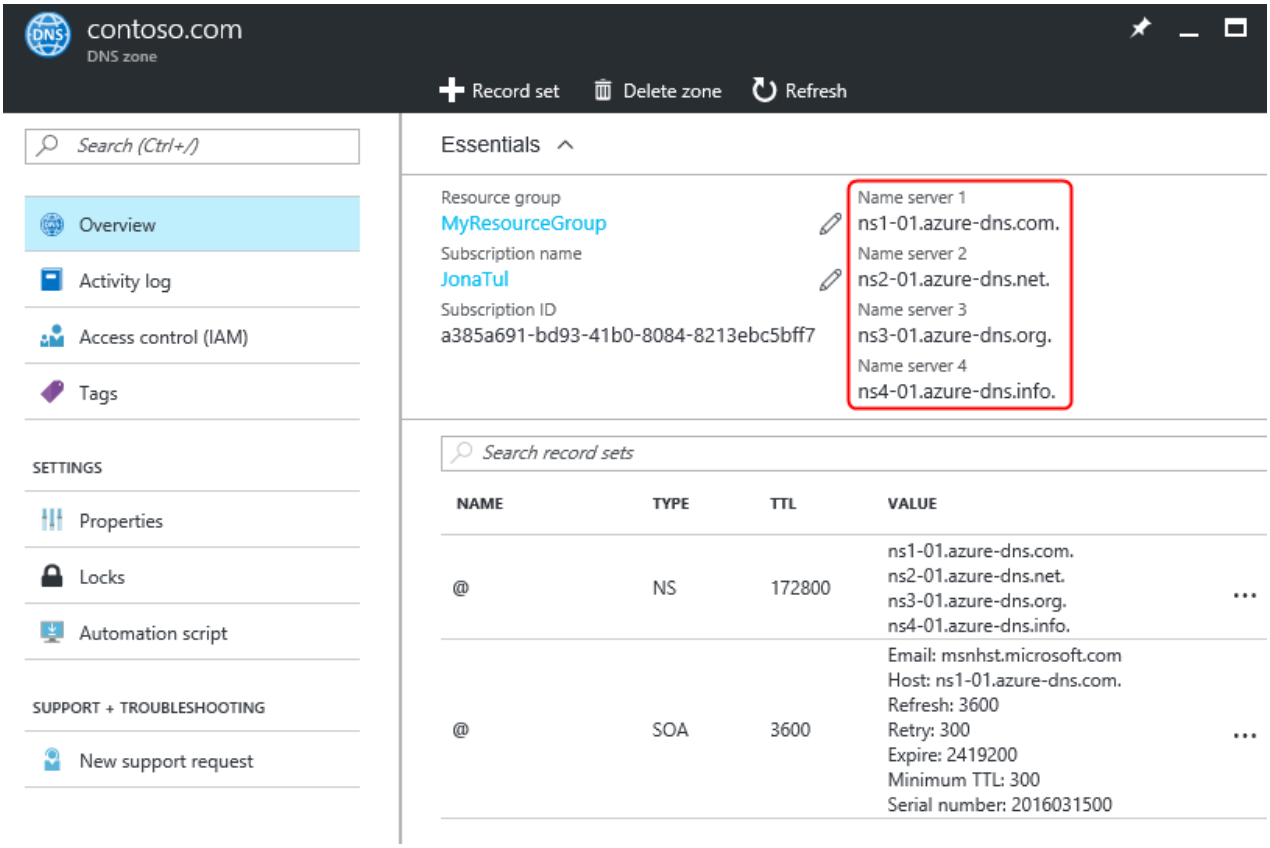


Test name servers

You can test your DNS zone is present on the Azure DNS name servers by using DNS tools such as nslookup, dig,

or the Resolve-DnsName PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. The name servers for your zone are given in the Azure portal:



Be sure the substitute the correct name server for your zone into the command below.

```
nslookup
> set type=SOA
> server ns1-01.azure-dns.com
> contoso.com

Server: ns1-01.azure-dns.com
Address:  208.76.47.1

contoso.com
        primary name server = ns1-01.azure-dns.com
        responsible mail addr = azuredns-hostmaster.microsoft.com
        serial  = 1
        refresh = 3600 (1 hour)
        retry   = 300 (5 mins)
        expire  = 2419200 (28 days)
        default TTL = 300 (5 mins)
```

Next steps

After creating a DNS zone, create record sets and records to create DNS records for your Internet domain.

Create a DNS zone using PowerShell

1/17/2017 • 4 min to read • Edit on GitHub

This article walks you through the steps to create a DNS zone using Azure PowerShell. You can also create a DNS zone using the cross-platform Azure CLI or 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.

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

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.

Sign in to your Azure account

Open your PowerShell console and connect to your account. For more information, see Using PowerShell with Resource Manager.

```
Login-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*. Use the example to create a DNS zone, substituting the values for your own.

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

Verify your DNS zone

View records

Creating a DNS zone also creates the following DNS records:

- The *Start of Authority* (SOA) record. This record is present at the root of every DNS zone.
- The authoritative name server (NS) records. These records show which name servers are hosting the zone. Azure DNS uses a pool of name servers, and so different name servers may be assigned to different zones in Azure DNS. For more information, see delegate a domain to Azure DNS.

To view these records, use `Get-AzureRmDnsRecordSet`:

```
Get-AzureRmDnsRecordSet -ZoneName contoso.com -ResourceGroupName MyAzureResourceGroup

Name              : @
ZoneName          : contoso.com
ResourceGroupName : MyAzureResourceGroup
Ttl               : 172800
Etag              : f573237b-088c-424a-b53c-08567d87d049
RecordType        : NS
Records           : {ns1-01.azure-dns.com., ns2-01.azure-dns.net., ns3-01.azure-dns.org., ns4-01.azure-
dns.info.}
Metadata          :

Name              : @
ZoneName          : contoso.com
ResourceGroupName : MyAzureResourceGroup
Ttl               : 3600
Etag              : bf88a27d-0eec-4847-ad42-f0c83b9a2c32
RecordType        : SOA
Records           : {[ns1-01.azure-dns.com.,azuredns-hostmaster.microsoft.com,3600,300,2419200,300]}
Metadata          :
```

NOTE

Record sets at the root (or *apex*) of a DNS Zone use **@** as the record set name.

Test name servers

You can test your DNS zone is present on the Azure DNS name servers by using DNS tools such as nslookup, dig, or the Resolve-DnsName PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. The name servers for your zone are given in the NS records, as listed by `Get-AzureRmDnsRecordSet` above. Be sure the substitute the correct values for your zone into the following example:

```
nslookup
> set type=SOA
> server ns1-01.azure-dns.com
> contoso.com

Server: ns1-01.azure-dns.com
Address:  40.90.4.1

contoso.com
        primary name server = ns1-01.azure-dns.com
        responsible mail addr = azuredns-hostmaster.microsoft.com
        serial  = 1
        refresh = 3600 (1 hour)
        retry   = 300 (5 mins)
        expire  = 2419200 (28 days)
        default TTL = 300 (5 mins)
```

Next steps

After creating a DNS zone, create record sets and records to create DNS records for your Internet domain.

Create an Azure DNS zone using CLI

1/17/2017 • 5 min to read • Edit on GitHub

This article walks you through the steps to create a DNS zone using the cross-platform Azure CLI, which is available for Windows, Mac and Linux. You can also create a DNS zone using Azure PowerShell or 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.

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

```
azure login
```

Switch CLI mode

Azure DNS uses Azure Resource Manager. Make sure you switch CLI mode to use Azure Resource Manager commands.

```
azure config mode arm
```

Select the subscription

Check the subscriptions for the account.

```
azure account list
```

Choose which of your Azure subscriptions to use.

```
azure account set "subscription name"
```

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.

```
azure group create -n myresourcegroup --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.

```
azure provider register --namespace Microsoft.Network
```

Create a DNS zone

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

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

```
azure network dns zone create MyResourceGroup contoso.com
```

Verify your DNS zone

View records

Creating a DNS zone also creates the following DNS records:

- The *Start of Authority* (SOA) record. This record is present at the root of every DNS zone.
- The authoritative name server (NS) records. These records show which name servers are hosting the zone. Azure DNS uses a pool of name servers, and so different name servers may be assigned to different zones in Azure DNS. For more information, see delegate a domain to Azure DNS.

To view these records, use `azure network dns-record-set list`:

```
azure network dns record-set list MyResourceGroup contoso.com

info:    Executing command network dns record-set list
+ Looking up the DNS Record Sets
data:    Name                          : @
data:    Type                          : NS
data:    TTL                           : 172800
data:    Records:
data:      ns1-01.azure-dns.com.
data:      ns2-01.azure-dns.net.
data:      ns3-01.azure-dns.org.
data:      ns4-01.azure-dns.info.
data:
data:    Name                          : @
data:    Type                          : SOA
data:    TTL                           : 3600
data:    Email                         : azuredns-hostmaster.microsoft.com
data:    Host                          : ns1-01.azure-dns.com.
data:    Serial Number                 : 2
data:    Refresh Time                  : 3600
data:    Retry Time                    : 300
data:    Expire Time                   : 2419200
data:    Minimum TTL                   : 300
data:
info:    network dns record-set list command OK
```

> **NOTE**
>
> Record sets at the root (or *apex*) of a DNS Zone use **@** as the record set name.

Test name servers

You can test your DNS zone is present on the Azure DNS name servers by using DNS tools such as nslookup, dig, or the `Resolve-DnsName` PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. The name servers for your zone are given in the NS records, as given by `azure network dns record-set list`.

The following example uses 'dig' to query the domain contoso.com using the name servers assigned to the DNS zone. Be sure the substitute the correct values for your zone.

```
 > dig @ns1-01.azure-dns.com contoso.com

 <<>> DiG 9.10.2-P2 <<>> @ns1-01.azure-dns.com contoso.com
(1 server found)
global options: +cmd
 Got answer:
->>HEADER<<- opcode: QUERY, status: NOERROR, id: 60963
 flags: qr aa rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1
 WARNING: recursion requested but not available

 OPT PSEUDOSECTION:
 EDNS: version: 0, flags:; udp: 4000
  QUESTION SECTION:
contoso.com.                        IN      A

 AUTHORITY SECTION:
contoso.com.         3600     IN      SOA     ns1-01.azure-dns.com. azuredns-hostmaster.microsoft.com. 1 3600
300 2419200 300

Query time: 93 msec
SERVER: 208.76.47.5#53(208.76.47.5)
WHEN: Tue Jul 21 16:04:51 Pacific Daylight Time 2015
MSG SIZE  rcvd: 120
```

Next steps

After creating a DNS zone, create DNS record sets and records in your zone.

Create DNS record sets and records by using the Azure portal

1/17/2017 • 4 min to read • Edit on GitHub

This article walks you through the process of creating records and records sets by using the Azure portal. To do this, you first need to understand DNS records and 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, 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.

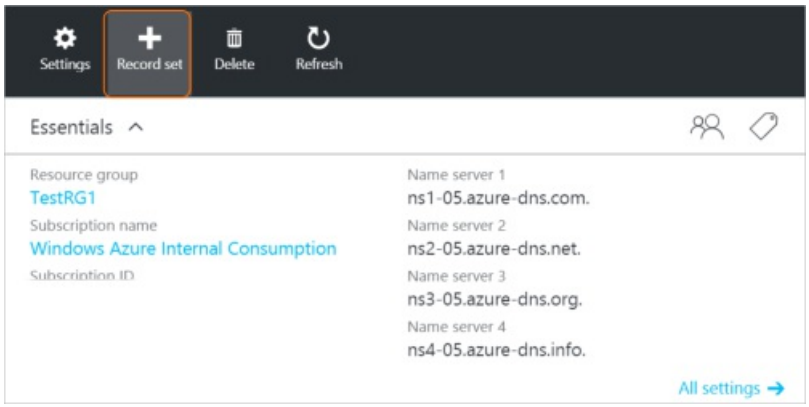
The examples on this page all use the 'A' DNS record type. The process for other record types is similar.

If your new record has the same name and type as an existing record, you need to add it to the existing record set—see Manage DNS records and record sets by using the Azure portal. If your new record has a different name and type to all existing records, you need to create a new record set, as explained below.

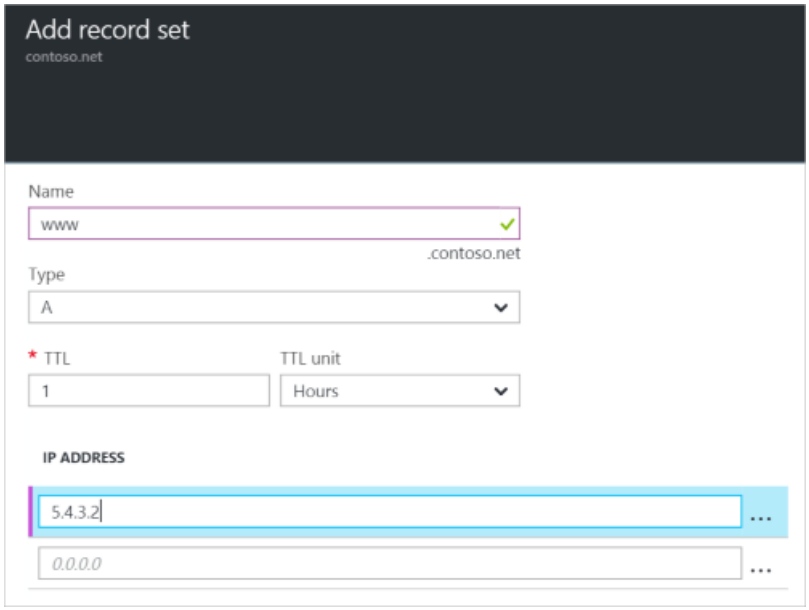
Create records in a new record set

The following example walks you through the process of creating a record set and record by using the Azure portal.

1. Sign in to the portal.
2. Go to the **DNS zone** blade in which you want to create a record set.
3. At the top of the **DNS zone** blade, select **+ Record set** to open the **Add record set** blade.



4. On the **Add record set** blade, name your record set. For example, you could name your record set "**www**".



5. Select the type of record you want to create. For example, select **A**.
6. Set the **TTL**. The default time to live in the portal is one hour.
7. Add the details of each record in the record set. In this case, since the record type is 'A', you need to add the A record IP addresses, one IP address per line.
8. After you finish adding IP addresses, select **OK** at the bottom of the blade. The DNS record set will be created.

Verify name resolution

You can test your DNS records are present on the Azure DNS name servers by using DNS tools such as nslookup, dig, or the Resolve-DnsName PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. Be sure the substitute the correct values for your records zone into the command below.

```
nslookup
> set type=A
> server ns1-01.azure-dns.com
> www.contoso.com

Server:  ns1-01.azure-dns.com
Address:  40.90.4.1

Name:    www.contoso.com
Address:  1.2.3.4
```

Next steps

Learn how to delegate your domain name to the Azure DNS name servers

To manage your record set and records, see Manage DNS records and record sets by using the Azure portal.

Create DNS record sets and records by using PowerShell

1/17/2017 • 4 min to read • Edit on GitHub

This article walks you through the process of creating records and records sets by using 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, 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 record set and record

This section describes how to create DNS records in Azure DNS. The examples assume you have already installed Azure PowerShell, signed in, and created a DNS zone.

The examples on this page all use the 'A' DNS record type. For other record types and further details on how to manage DNS records and record sets, see Manage DNS records and record sets by using PowerShell.

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 records in a new record set

You create record sets by using the `New-AzureRmDnsRecordSet` 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.

To create a record set in the apex of the zone (in this case, "contoso.com"), use the record name "@", including the quotation marks. This is a common DNS convention.

The following example creates a new 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-AzureRmDnsRecordSet -Name "www" -RecordType "A" -ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup"
-Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -IPv4Address "1.2.3.4")
```

If you need to create a new record set containing more than one record, you first need to create a local array containing the records to be added. This is passed to `New-AzureRmDnsRecordSet` as follows:

```
$aRecords = @()
$aRecords += New-AzureRmDnsRecordConfig -IPv4Address "1.2.3.4"
$aRecords += New-AzureRmDnsRecordConfig -IPv4Address "2.3.4.5"
New-AzureRmDnsRecordSet -Name "www" –ZoneName "contoso.com" -ResourceGroupName "MyResourceGroup" -Ttl 3600 -
RecordType A -DnsRecords $aRecords
```

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-AzureRmDnsRecordSet -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-AzureRmDnsRecordConfig -RecordSet $rs -Ipv4Address "5.6.7.8"
   ```

3. Commit the change back to the Azure DNS service

   ```
   Set-AzureRmDnsRecordSet -RecordSet $rs
   ```

Verify name resolution

You can test your DNS records are present on the Azure DNS name servers by using DNS tools such as nslookup, dig, or the Resolve-DnsName PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. Be sure the substitute the correct values for your records zone

into the following example:

```
nslookup
> set type=A
> server ns1-01.azure-dns.com
> www.contoso.com

Server:  ns1-01.azure-dns.com
Address:  40.90.4.1

Name:    www.contoso.com
Address:  1.2.3.4
```

Next steps

Learn how to delegate your domain name to the Azure DNS name servers

Learn how to manage DNS zones by using PowerShell.

Learn how to manage DNS records and record sets by using PowerShell.

Create DNS records using the Azure CLI

1/17/2017 • 4 min to read • Edit on GitHub

This article walks you through the process of creating records and records sets by using the Azure CLI.

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, 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 record set and record

This section describes how to create DNS records in Azure DNS. The examples assume you have already installed

the Azure CLI, signed in, and created a DNS zone.

The examples on this page all use the 'A' DNS record type. For other record types and further details on how to manage DNS records and record sets, see Manage DNS records and record sets by using the Azure CLI.

Create a DNS record

To create a DNS record, use the `azure network dns record-set add-record` command. For help, see `azure network dns record-set add-record -h`.

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.

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 Manage DNS records in Azure DNS using the Azure CLI.

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*.

```
azure network dns record-set add-record MyResourceGroup contoso.com www A -a 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:

```
azure network dns record-set add-record MyResourceGroup contoso.com "@" A -a 1.2.3.4
```

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 `-a <IPv4 address>`. See `azure network dns record-set add-record -h` to list the parameters for other record types. For examples for each record type, see Manage DNS records and record sets by using the Azure CLI.

Verify name resolution

You can test your DNS records are present on the Azure DNS name servers by using DNS tools such as nslookup, dig, or the Resolve-DnsName PowerShell cmdlet.

If you haven't yet delegated your domain to use the new zone in Azure DNS, you need to direct the DNS query directly to one of the name servers for your zone. Be sure the substitute the correct values for your records zone into the command below.

```
nslookup
> set type=A
> server ns1-01.azure-dns.com
> www.contoso.com

Server:  ns1-01.azure-dns.com
Address:  40.90.4.1

Name:    www.contoso.com
Address:  1.2.3.4
```

Next steps

Learn how to delegate your domain name to the Azure DNS name servers

Learn how to manage DNS zones by using the Azure CLI.

Learn how to manage DNS records and record sets by using the Azure CLI.

Delegate a domain to Azure DNS

1/17/2017 • 8 min to read • Edit on GitHub

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 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 a number of 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 will verify if the Internet domain you want to use is available and allow you to purchase it. Once the domain name is registered, you will be 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.

> **NOTE**
> To find out more information on who owns a given domain name, or for information on how to buy a domain, see Internet domain management in Azure AD.

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.

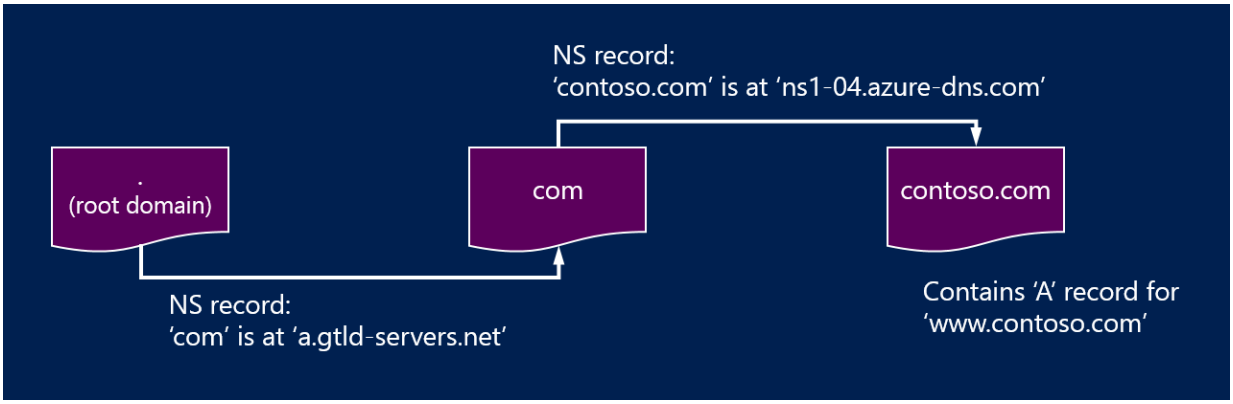
> **NOTE**
> Azure DNS provides an authoritative DNS service. It does not provide a recursive DNS service.
>
> Cloud Services and VMs in Azure are automatically configured to use a recursive DNS services that is provided separately as part of Azure's infrastructure. For information on how to change these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



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.com' zone contains the NS records for 'contoso.com' (in addition to the NS records in 'com'). These are called authoritative NS records and they sit at the apex of the child zone.

Delegating a domain to Azure DNS

Once you create your DNS zone in Azure DNS, you need to set up NS records in the parent zone to make Azure DNS the authoritative source for name resolution for your zone. For domains purchased from a registrar, your registrar will offer the option to set up these NS records.

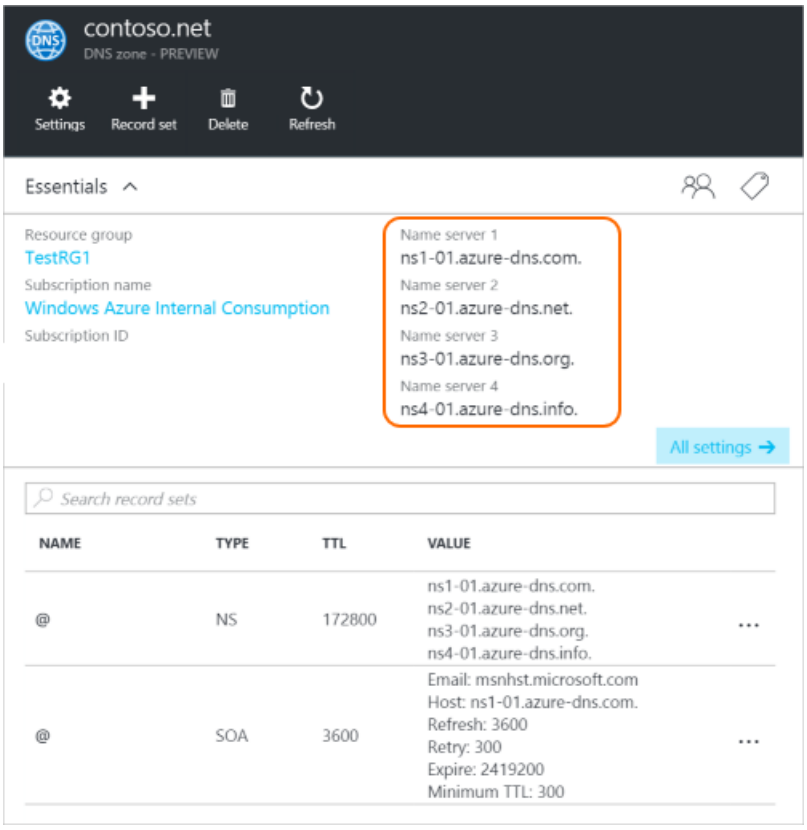
> **NOTE**
> You do not have to own a domain in order to create a DNS zone with that domain name in Azure DNS. However, you do need to own the domain to set up the delegation to Azure DNS with the registrar.

For example, suppose you purchase the domain 'contoso.com' and create a zone with the name 'contoso.com' in Azure DNS. As the owner of the domain, your registrar will offer you the option to configure the name server addresses (that is, the NS records) for your domain. The registrar will store these NS records in the parent domain, in this case '.com'. Clients around the world will then be directed to your domain in Azure DNS zone when trying to resolve DNS records in 'contoso.com'.

Finding the name server names

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

The easiest way to see the name servers assigned to your zone is via the Azure portal. 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 containing the assigned name servers. To see the name server names via Azure PowerShell or Azure CLI, you simply need to retrieve these records.

Using Azure PowerShell, the authoritative NS records can be retrieved as follows. Note that the record name "@" is used to refer to records at the apex of the zone.

```
PS> $zone = Get-AzureRmDnsZone -Name contoso.net -ResourceGroupName MyResourceGroup
PS> Get-AzureRmDnsRecordSet -Name "@" -RecordType NS -Zone $zone

Name              : @
ZoneName          : contoso.net
ResourceGroupName : MyResourceGroup
Ttl               : 3600
Etag              : 5fe92e48-cc76-4912-a78c-7652d362ca18
RecordType        : NS
Records           : {ns1-01.azure-dns.com, ns2-01.azure-dns.net, ns3-01.azure-dns.org,
                    ns4-01.azure-dns.info}
Tags              : {}
```

You can also use the cross-platform Azure CLI to retrieve the authoritative NS records and hence discover the name servers assigned to your zone:

```
C:\> azure network dns record-set show MyResourceGroup contoso.net @ NS
info:    Executing command network dns record-set show
    + Looking up the DNS Record Set "@" of type "NS"
data:    Id                              :
/subscriptions/.../resourceGroups/MyResourceGroup/providers/Microsoft.Network/dnszones/contoso.net/NS/@
data:    Name                            : @
data:    Type                            : Microsoft.Network/dnszones/NS
data:    Location                        : global
data:    TTL                             : 172800
data:    NS records
data:        Name server domain name     : ns1-01.azure-dns.com.
data:        Name server domain name     : ns2-01.azure-dns.net.
data:        Name server domain name     : ns3-01.azure-dns.org.
data:        Name server domain name     : ns4-01.azure-dns.info.
data:
info:    network dns record-set show command OK
```

To set up delegation

Each registrar has their own DNS management tools to change the name server records for a domain. In the registrar's DNS management page, edit the NS records and replace the NS records with the ones Azure DNS created.

When delegating a domain to Azure DNS, you must use the name server names provided by Azure DNS. You should always use all 4 name server names, regardless of the name of your domain. Domain delegation does not require the name server name to use the same top-level domain as your domain.

You should not use 'glue records' to point to the Azure DNS name server IP addresses, since these IP addresses may change in future. Delegations using name server names in your own zone, sometimes called 'vanity name servers', are not currently supported in Azure DNS.

To verify name resolution is working

After completing the delegation, you can verify that name resolution is working by using a tool such as 'nslookup' to query the SOA record for your zone (which is also automatically created when the zone is created).

Note that you do not have to specify the Azure DNS name servers, since the normal DNS resolution process will find the name servers automatically if the delegation has been set up correctly.

```
nslookup -type=SOA contoso.com

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

contoso.com
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)
```

Delegating sub-domains in Azure DNS

If you want to set up a separate child zone, you can delegate a sub-domain in Azure DNS. For example, having set up and delegated 'contoso.com' in Azure DNS, suppose you would like to set up a separate child zone, 'partners.contoso.com'.

Setting up a sub-domain follows a similar process as a normal delegation. The only difference is that in step 3

the NS records must be created in the parent zone 'contoso.com' in Azure DNS, rather than being set up via a domain registrar.

1. Create the child zone 'partners.contoso.com' in Azure DNS.
2. Look up the authoritative NS records in the child zone to obtain the name servers hosting the child zone in Azure DNS.
3. Delegate the child zone by configuring NS records in the parent zone pointing to the child zone.

To delegate a sub-domain

The following PowerShell example demonstrates how this works. The same steps can be executed via the Azure Portal, or via the cross-platform Azure CLI.

Step 1. Create the parent and child zones

First, we create the parent and child zones. These can be in same resource group or different resource groups.

```
$parent = New-AzureRmDnsZone -Name contoso.com -ResourceGroupName RG1
$child = New-AzureRmDnsZone -Name partners.contoso.com -ResourceGroupName RG1
```

Step 2. Retrieve NS records

Next, we retrieve the authoritative NS records from child zone as shown in the next example. This contains the name servers assigned to the child zone.

```
$child_ns_recordset = Get-AzureRmDnsRecordSet -Zone $child -Name "@" -RecordType NS
```

Step 3. Delegate the child zone

Create corresponding NS record set in the parent zone to complete the delegation. Note that the record set name in the parent zone matches the child zone name, in this case "partners".

```
$parent_ns_recordset = New-AzureRmDnsRecordSet -Zone $parent -Name "partners" -RecordType NS -Ttl 3600
$parent_ns_recordset.Records = $child_ns_recordset.Records
Set-AzureRmDnsRecordSet -RecordSet $parent_ns_recordset
```

To verify name resolution is working

You can verify that everything is set up correctly by looking up the SOA record of the child zone.

```
nslookup -type=SOA partners.contoso.com

Server: ns1-08.azure-dns.com
Address: 208.76.47.8

partners.contoso.com
    primary name server = ns1-08.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)
```

Next steps

Manage DNS zones

Manage DNS records

Create DNS records for a web app in a custom domain

1/17/2017 • 3 min to read • Edit on GitHub

You can use Azure DNS to host a custom domain for your web apps. For example, you are creating an Azure web app and you want your users to access it by either using contoso.com, or www.contoso.com as an FQDN.

To do this, you have to create two records:

- A root "A" record pointing to contoso.com
- 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.

Before you begin

Before you begin, you must first 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 DNS 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.

1. Create an A record for your custom domain

An A record is used to map a name to its IP address. In the following example we will assign @ as an A record to an IPv4 address:

Step 1

Create an A record and assign to a variable $rs

```
$rs= New-AzureRMDnsRecordSet -Name "@" -RecordType "A" -ZoneName "contoso.com" -ResourceGroupName
"MyAzureResourceGroup" -Ttl 600
```

Step 2

Add the IPv4 value to the previously created record set "@" using the $rs variable assigned. The IPv4 value assigned will be the IP address for your web app.

To find the IP address for a web app, follow the steps in Configure a custom domain name in Azure App Service.

```
Add-AzureRMDnsRecordConfig -RecordSet $rs -Ipv4Address <your web app IP address>
```

Step 3

Commit the changes to the record set. Use `Set-AzureRMDnsRecordSet` to upload the changes to the record set to Azure DNS:

```
Set-AzureRMDnsRecordSet -RecordSet $rs
```

2. Create a CNAME record for your custom domain

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

Step 1

Open PowerShell and create a new CNAME record set and assign to a variable $rs. This example will create a record set type CNAME with a "time to live" of 600 seconds in DNS zone named "contoso.com".

```
$rs = New-AzureRMDnsRecordSet -ZoneName contoso.com -ResourceGroupName myresourcegroup -Name "www" -RecordType
"CNAME" -Ttl 600

Name              : www
ZoneName          : contoso.com
ResourceGroupName : myresourcegroup
Ttl               : 600
Etag              : 8baceeb9-4c2c-4608-a22c-229923ee1856
RecordType        : CNAME
Records           : {}
Tags              : {}
```

Step 2

Once the CNAME record set is created, you need to create an alias value which will point to the web app.

Using the previously assigned variable "$rs" you can use the PowerShell command below to create the alias for the web app contoso.azurewebsites.net.

```
Add-AzureRMDnsRecordConfig -RecordSet $rs -Cname "contoso.azurewebsites.net"

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

Step 3

Commit the changes using the `Set-AzureRMDnsRecordSet` cmdlet:

```
Set-AzureRMDnsRecordSet -RecordSet $rs
```

You can validate the record was created correctly by querying the "www.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
```

Create an "awverify" record for web apps

If you decide to use an A record for your web app, you must go through a verification process to ensure you own the custom domain. This verification step is done by creating a special CNAME record named "awverify". This section applies to A records only.

Step 1

Create the "awverify" record. In the example below, we will create the "aweverify" record for contoso.com to verify ownership for the custom domain.

```
$rs = New-AzureRMDnsRecordSet -ZoneName contoso.com -ResourceGroupName myresourcegroup -Name "awverify" -
RecordType "CNAME" -Ttl 600

Name              : awverify
ZoneName          : contoso.com
ResourceGroupName : myresourcegroup
Ttl               : 600
Etag              : 8baceeb9-4c2c-4608-a22c-229923ee1856
RecordType        : CNAME
Records           : {}
Tags              : {}
```

Step 2

Once the record set "awverify" is created, assign the CNAME record set alias. In the example below, we will assign the CNAMe record set alias to awverify.contoso.azurewebsites.net.

```
Add-AzureRMDnsRecordConfig -RecordSet $rs -Cname "awverify.contoso.azurewebsites.net"

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

Step 3

Commit the changes using the `Set-AzureRMDnsRecordSet cmdlet`, as shown in the command below.

```
Set-AzureRMDnsRecordSet -RecordSet $rs
```

Next steps

Follow the steps in Configuring a custom domain name for App Service to configure your web app to use a custom domain.

How to manage DNS Zones using PowerShell

1/17/2017 • 6 min to read • Edit on GitHub

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.

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

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.

Sign in to your Azure account

Open your PowerShell console and connect to your account. For more information, see Using PowerShell with Resource Manager.

```
Login-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"
}
```

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

1/17/2017 • 5 min to read • Edit on GitHub

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.

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

```
azure login
```

Switch CLI mode

Azure DNS uses Azure Resource Manager. Make sure you switch CLI mode to use Azure Resource Manager commands.

```
azure config mode arm
```

Select the subscription

Check the subscriptions for the account.

```
azure account list
```

Choose which of your Azure subscriptions to use.

```
azure account set "subscription name"
```

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.

```
azure group create -n myresourcegroup --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.

```
azure provider register --namespace Microsoft.Network
```

Getting help

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

```
azure network dns -h
azure network dns zone -h
azure network dns zone create -h
```

Create a DNS zone

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

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

```
azure network dns zone create MyResourceGroup 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`):

```
azure network dns zone create MyResourceGroup contoso.com -t "project=demo";"env=test"
```

Get a DNS zone

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

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

```
azure network dns zone show MyResourceGroup contoso.com

info:    Executing command network dns zone show
+ Looking up the dns zone "contoso.com"
data:    Id                              : /subscriptions/.../contoso.com
data:    Name                            : contoso.com
data:    Type                            : Microsoft.Network/dnszones
data:    Location                        : global
data:    Number of record sets           : 2
data:    Max number of record sets       : 5000
data:    Name servers:
data:        ns1-01.azure-dns.com.
data:        ns2-01.azure-dns.net.
data:        ns3-01.azure-dns.org.
data:        ns4-01.azure-dns.info.
data:    Tags                            : project=demo;env=test
info:    network dns zone show command OK
```

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

List DNS zones

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

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

```
azure network dns zone list MyResourceGroup
```

Omitting the resource group lists all zones in the subscription:

```
azure network dns zone list
```

Update a DNS zone

Changes to a DNS zone resource can be made using `azure network dns zone set`. For help, see `azure network dns zone set -h`.

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.

```
azure network dns zone set MyResourceGroup contoso.com -t "team=support"
```

Delete a DNS Zone

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

> **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 `--quiet` switch (short form `-q`) suppresses this prompt.

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

```
azure network dns zone delete MyResourceGroup 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

1/17/2017 • 3 min to read • Edit on GitHub

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

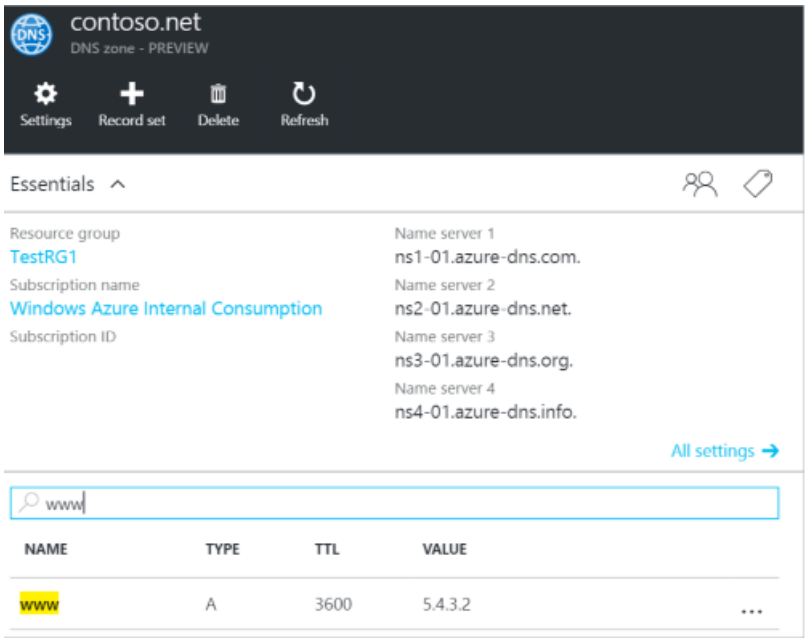
It's important to understand the difference between DNS record sets and individual DNS records. A record set is a collection of records in a zone that have the same name and are the same type. For more information, see Create DNS record sets and records by using the Azure portal.

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

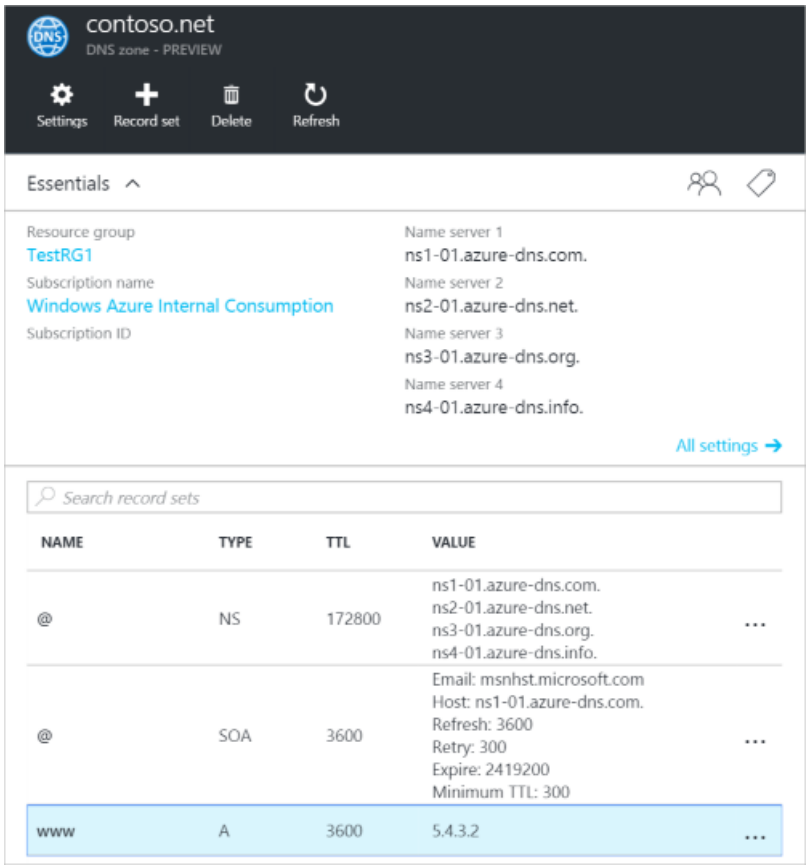
1. In the Azure portal, go to the **DNS zone** blade.
2. Search for the record set and select it. This opens the record set properties.



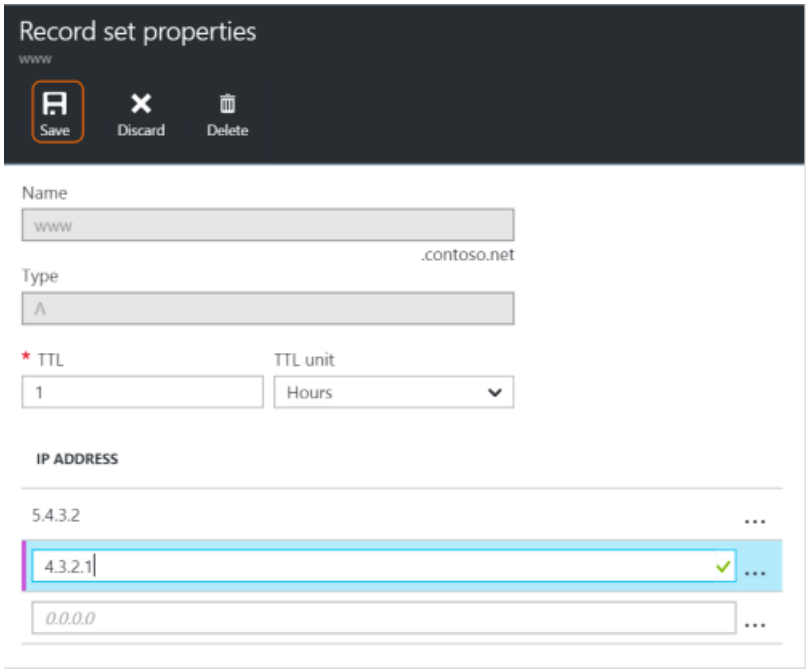
Add a new record to a record set

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.

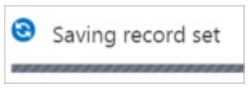


2. Specify the record set properties by filling in the fields.



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.



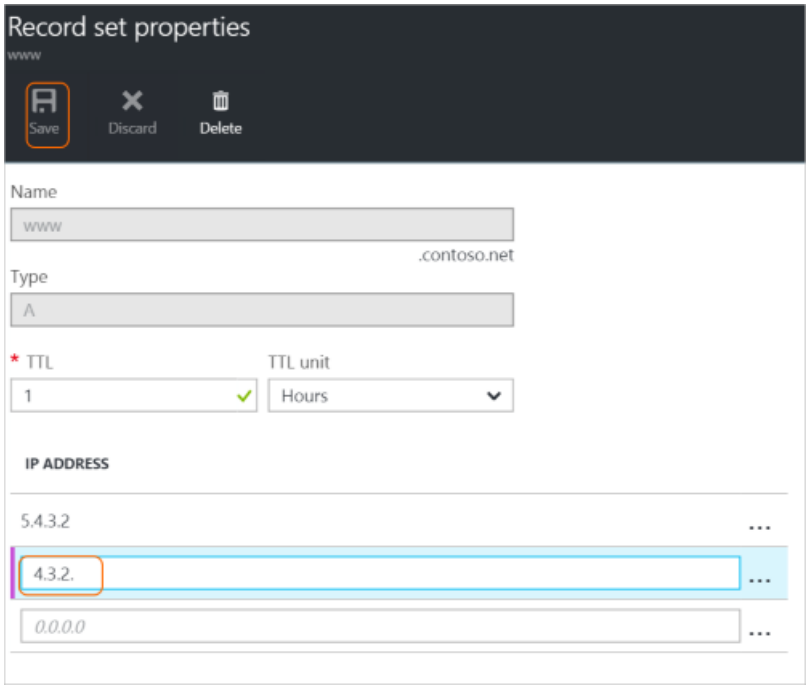
After the record has been saved, the values on the **DNS zone** blade will reflect the new record.

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.



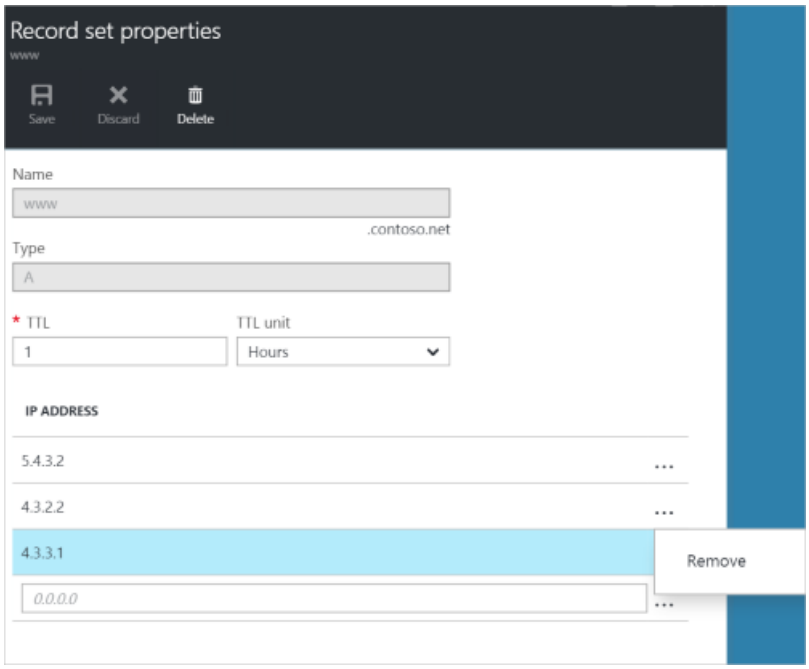
3. Click **Save** at the top of the blade to save your settings. In the upper right corner, you'll see the notification that the record has been saved.

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.

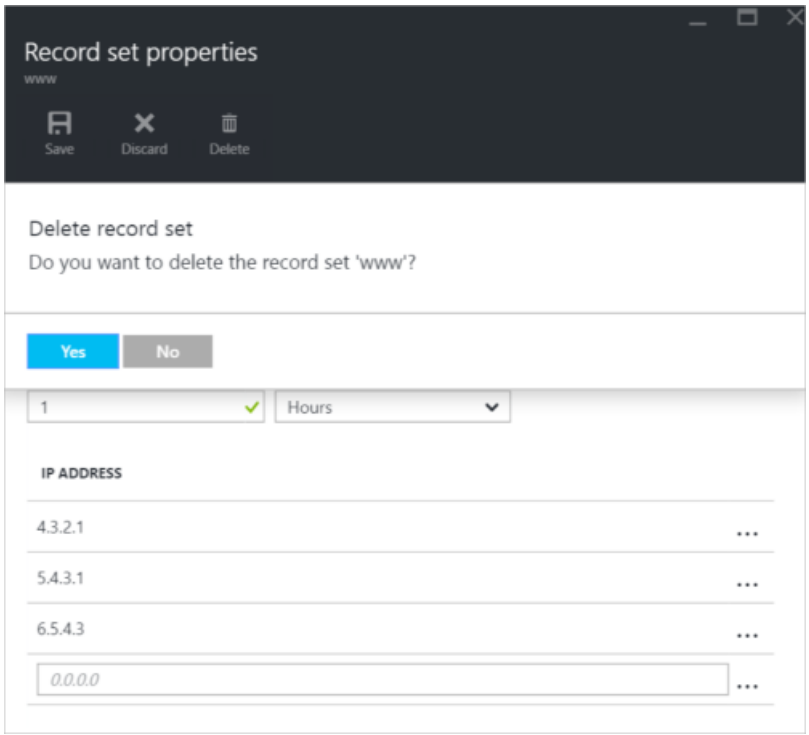
1. On the **Record set properties** blade for your record set, search for the record.
2. Click the record that you want to remove. Then select **Remove**.



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

You cannot add to, remove, or modify the records in the automatically created NS record set at the zone apex (name = "@"). The only change that's permitted is to modify the record set TTL.

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 How to manage reverse DNS records for your services using PowerShell.

Manage DNS records in Azure DNS using Azure PowerShell

1/17/2017 • 14 min to read • Edit on GitHub

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.

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, 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-AzureRmDnsRecordSet` 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-AzureRmDnsRecordSet -Name www -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup -Ttl
3600 -DnsRecords (New-AzureRmDnsRecordConfig -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 "@" (including quotation marks):

```
New-AzureRmDnsRecordSet -Name "@" -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup -Ttl
3600 -DnsRecords (New-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet` as follows:

```
$aRecords = @()
$aRecords += New-AzureRmDnsRecordConfig -IPv4Address 1.2.3.4
$aRecords += New-AzureRmDnsRecordConfig -IPv4Address 2.3.4.5
New-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -Name www -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup -Ttl
3600 -DnsRecords (New-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -Name "test-aaaa" -RecordType AAAA -ZoneName contoso.com -ResourceGroupName
MyResourceGroup -Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -Ipv6Address 2607:f8b0:4009:1803::1005)
```

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-AzureRmDnsRecordSet -Name test-cname -RecordType CNAME -ZoneName contoso.com -ResourceGroupName
MyResourceGroup -Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet -Name "@" -RecordType MX -ZoneName contoso.com -ResourceGroupName MyResourceGroup -Ttl
3600 -DnsRecords (New-AzureRmDnsRecordConfig -Exchange mail.contoso.com -Preference 5)
```

Create an NS record set with a single record

```
New-AzureRmDnsRecordSet -Name test-ns -RecordType NS -ZoneName contoso.com -ResourceGroupName MyResourceGroup -
Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -Nsdname ns1.contoso.com)
```

Create a PTR record set with a single 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.

```
New-AzureRmDnsRecordSet -Name 10 -RecordType PTR -ZoneName my-arpa-zone.com -ResourceGroupName MyResourceGroup
-Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet -Name _sip._tls -RecordType SRV -ZoneName contoso.com -ResourceGroupName
MyResourceGroup -Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet -Name test-txt -RecordType TXT -ZoneName contoso.com -ResourceGroupName MyResourceGroup
-Ttl 3600 -DnsRecords (New-AzureRmDnsRecordConfig -Value "This is a TXT record")
```

Get a record set

To retrieve an existing record set, use `Get-AzureRmDnsRecordSet`. This cmdlet returns a local object that represents the record set in Azure DNS.

As with `New-AzureRmDnsRecordSet`, 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-AzureRmDnsRecordSet -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-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyResourceGroup
$rs = Get-AzureRmDnsRecordSet -Name www -RecordType A -Zone $zone
```

List record sets

You can also use `Get-AzureRmDnsZone` 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-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -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:

```
$recordsets = Get-AzureRmDnsRecordSet -ZoneName contoso.com -ResourceGroupName MyResourceGroup | where
{$_.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-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyResourceGroup
$recordsets = Get-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -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-AzureRmDnsRecordConfig -RecordSet $rs -Ipv4Address"5.6.7.8
```

3. Commit the change back to the Azure DNS service.

```
Set-AzureRmDnsRecordSet -RecordSet $rs
```

Using `Set-AzureRmDnsRecordSet` *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-AzureRmDnsRecordSet -Name www –ZoneName contoso.com -ResourceGroupName MyResourceGroup -RecordType A | Add-
AzureRmDnsRecordConfig -Ipv4Address 5.6.7.8 | Set-AzureRmDnsRecordSet
```

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-AzureRmDnsRecordConfig` with other parameters specific to each record type. The parameters for each record type are the same as for the `New-AzureRmDnsRecordConfig` 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-AzureRmDnsRecordSet -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-AzureRmDnsRecordConfig -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-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -Name www –ZoneName contoso.com -ResourceGroupName MyResourceGroup -RecordType A |
Remove-AzureRmDnsRecordConfig -Ipv4Address 5.6.7.8 | Set-AzureRmDnsRecordSet
```

Different record types are supported by passing the appropriate type-specific parameters to `Remove-AzureRmDnsRecordSet`. The parameters for each record type are the same as for the `New-AzureRmDnsRecordConfig` 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-AzureRmDnsRecordSet`.
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-AzureRmDnsRecordSet` cmdlet. This *replaces* the existing record set in Azure DNS with the record set specified.

When using `Set-AzureRmDnsRecordSet`, 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-AzureRmDnsRecordSet -name www -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup
$rs.Records[0].Ipv4Address = 9.8.7.6
Set-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -Name "@" -RecordType SOA -ZoneName contoso.com -ResourceGroupName
MyResourceGroup
$rs.Records[0].Email = "admin.contoso.com"
Set-AzureRmDnsRecordSet -RecordSet $rs
```

To modify NS records at the zone apex

You cannot add to, remove, or modify the records in the automatically-created NS record set at the zone apex (`-Name "@"`, including quote marks). The only changes permitted are to modify the record set TTL and metadata.

The following example shows how to change the TTL property of the NS record set:

```
$rs = Get-AzureRmDnsRecordSet -Name "@" -RecordType NS -ZoneName contoso.com -ResourceGroupName MyResourceGroup
$rs.Ttl = 300
Set-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -RecordSet $rs
```

Delete a record set

Record sets can be deleted by using the `Remove-AzureRmDnsRecordSet` 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 "@"`). These are created automatically when the zone was created, and are deleted 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-AzureRmDnsRecordSet -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-AzureRmDnsZone -Name contoso.com -ResourceGroupName MyResourceGroup
Remove-AzureRmDnsRecordSet -Name www -RecordType A -Zone $zone
```

As a third option, the record set itself can be specified using a record set object:

```
$rs = Get-AzureRmDnsRecordSet -Name www -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup
Remove-AzureRmDnsRecordSet -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-AzureRmDnsRecordSet -Name www -RecordType A -ZoneName contoso.com -ResourceGroupName MyResourceGroup |
Remove-AzureRmDnsRecordSet
```

Confirmation prompts

The `New-AzureRmDnsRecordSet`, `Set-AzureRmDnsRecordSet`, and `Remove-AzureRmDnsRecordSet` 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 in Azure DNS using the Azure CLI

1/17/2017 • 12 min to read • Edit on GitHub

This article shows you how to manage DNS records for your DNS zone by using the cross-platform Azure command-line interface (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, 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 `azure network dns record-set add-record` command. For help, see `azure network dns record-set add-record -h`.

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 adda 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*.

```
azure network dns record-set add-record MyResourceGroup contoso.com www A -a 1.2.3.4
```

To create a record in the apex of the zone (in this case, "contoso.com"), use the record name "@", including the quotation marks:

```
azure network dns record-set add-record MyResourceGroup contoso.com "@" A -a 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 `azure network dns record-set create` command. For help, see `azure network dns record-set create -h`.

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 with a 60-second TTL, by using the `--ttl` parameter (short form `-l`):

```
azure network dns record-set create MyResourceGroup contoso.com www A --ttl 60
```

The following example creates a record set with two metadata entries, "dept=finance" and "environment=production", by using the `--metadata` parameter (short form `-m`):

```
azure network dns record-set create MyResourceGroup contoso.com www A --metadata
"dept=finance;environment=production"
```

Having created an empty record set, records can be added using `azure network dns record-set 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 `-a <IPv4 address>`. The parameters for each record type can be listed using `azure network dns record-set add-record -h`.

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

```
azure network dns record-set add-record MyResourceGroup contoso.com test-aaaa AAAA --ipv6-address
2607:f8b0:4009:1803::1005
```

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.

```
azure network dns record-set add-record  MyResourceGroup contoso.com  test-cname 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").

```
azure network dns record-set add-record MyResourceGroup contoso.com  "@" MX --exchange mail.contoso.com --
preference 5
```

Create an NS record

```
azure network dns record-set add-record MyResourceGroup  contoso.com  test-ns 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.

```
azure network dns record-set add-record MyResourceGroup my-arpa-zone.com "10" PTR --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.

```
azure network dns record-set add-record MyResourceGroup contoso.com  "_sip._tls" SRV --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.

```
azure network dns record-set add-record MyResourceGroup contoso.com test-txt TXT --text "This is a TXT record"
```

Get a record set

To retrieve an existing record set, use `azure network dns record-set show`. For help, see `azure network dns record-set show -h`.

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*:

```
azure network dns record-set show MyResourceGroup contoso.com www A
```

List record sets

You can list all records in a DNS zone by using the `azure network dns record-set list` command. For help, see `azure network dns record-set list -h`.

This example returns all record sets in the zone *contoso.com*, in resource group *MyResourceGroup*, regardless of name or record type:

```
azure network dns record-set list MyResourceGroup contoso.com
```

This example returns all record sets that match the given record type (in this case, 'A' records):

```
azure network dns record-set list MyResourceGroup contoso.com --type A
```

Add a record to an existing record set

You can use `azure network dns record-set 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 `azure network dns record-set delete-record`. For help, see `azure network dns record-set delete-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 **not** deleted. Instead, an empty record set is left. To delete the record set instead, see Delete a record set.

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 `azure network dns record-set add-record`. These parameters are described in Create a DNS record and Create records of other types above.

This command prompts for confirmation. This prompt can be suppressed using the `--quiet` switch (short form `-q`).

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*. The confirmation prompt is suppressed.

```
azure network dns record-set delete-record MyResourceGroup contoso.com www A -a 1.2.3.4 --quiet
```

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, MX, NS, PTR, SRV, or TXT record

To modify an existing record of type A, AAAA, 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:

```
azure network dns record-set add-record MyResourceGroup contoso.com www A -a 5.6.7.8
azure network dns record-set delete-record MyResourceGroup contoso.com www A -a 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

To modify a CNAME record, use `azure network dns record-set add-record` to add the new record value. Unlike other record types, a CNAME record set can only contain a single record. Therefore, the existing record is *replaced* when the new record is added, and does not need to be deleted separately. You will be prompted to accept this replacement.

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:

```
azure network dns record-set add-record MyResourceGroup contoso.com www CNAME --cname www.fabrikam.net
```

To modify an SOA record

Use `azure network dns record-set set-soa-record` to modify the SOA for a given DNS zone. For help, see `azure network dns record-set set-soa-record -h`.

The following example shows how to set the 'email' property of the SOA record for the zone *contoso.com* in the resource group *MyResourceGroup*:

```
azure network dns record-set set-soa-record rg1 contoso.com --email admin.contoso.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 set`. For help, see `azure network dns record-set set -h`.

The following example shows how to modify a record set TTL, in this case to 60 seconds:

```
azure network dns record-set set MyResourceGroup contoso.com www A --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 `azure network dns record-set set`. For help, see `azure network dns record-set set -h`.

The following example shows how to modify a record set with two metadata entries, "dept=finance" and "environment=production", by using the `--metadata` parameter (short form `-m`). Note that any existing metadata is *replaced* by the values given.

```
azure network dns record-set set MyResourceGroup contoso.com www A --metadata
"dept=finance;environment=production"
```

Delete a record set

Record sets can be deleted by using the `azure network dns record-set delete` command. For help, see `azure network dns record-set delete -h`. 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*:

```
azure network dns record-set delete MyResourceGroup contoso.com www A
```

You are prompted to confirm the delete operation. To suppress this prompt, use the `--quiet` switch (short form `-q`).

Next steps

Learn more about zones and records in Azure DNS.
Learn how to protect your zones and records when using Azure DNS.

How to manage reverse DNS records for your Azure services using Azure PowerShell

1/17/2017 • 7 min to read • Edit on GitHub

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 will retrieve the reverse DNS record of the sending server's IP address, and verify if that host is authorized to send e-mail from the originating domain. (Please note however that Azure Compute services do not support sending emails to external domains.)

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 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.

For more information on reverse DNS, please see Reverse DNS Lookup.

Azure support for reverse DNS

Azure supports two separate scenarios relating to reverse DNS:

1. Hosting the ARPA zone corresponding to your IP address block.
2. Allowing you to configure the reverse DNS record for the IP address assigned to your Azure service.

To support the former, Azure DNS can be used to host your ARPA zones and manage the PTR records for each reverse DNS lookup. The process of creating the 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.

To support the latter, Azure enables you to configure the reverse lookup for the IP addresses allocated to your 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. **The remainder of this article describes this scenario in detail.**

> **NOTE**
> Azure has two different deployment models for creating and working with resources: Resource Manager and classic. This article covers using the Resource Manager deployment model, which Microsoft recommends for most new deployments instead of the classic deployment model.

For more information about the classic deployment model, see How to manage reverse DNS records for your Azure services (classic) using Azure PowerShell.

Validation of reverse DNS records

To ensure a third party can't create reverse DNS records mapping to your DNS domains, Azure only allows the creation of a reverse DNS record where one of the following is true:

- The "ReverseFqdn" is the same as the "Fqdn" for the Public IP Address resource for which it has been specified, or the "Fqdn" for any Public IP Address within the same subscription e.g., "ReverseFqdn" is "contosoapp1.northus.cloudapp.azure.com.".
- The "ReverseFqdn" forward resolves to the name or IP of the Public IP Address for which it has been specified, or to any Public IP Address "Fqdn" or IP within the same subscription e.g., "ReverseFqdn" is "app1.contoso.com." which is a CName alias for "contosoapp1.northus.cloudapp.azure.com."

Validation checks are only performed when the reverse DNS property for a Public IP Address is set or modified. Periodic re-validation is not performed.

Add reverse DNS to existing Public IP addresses

You can add reverse DNS to an existing Public IP Address using the "Set-AzureRmPublicIpAddress" cmdlet:

```
PS C:\> $pip = Get-AzureRmPublicIpAddress -Name PublicIP -ResourceGroupName NRP-DemoRG-PS
PS C:\> $pip.DnsSettings.ReverseFqdn = "contosoapp1.westus.cloudapp.azure.com."
PS C:\> Set-AzureRmPublicIpAddress -PublicIpAddress $pip
```

If you wish to add reverse DNS to an existing Public IP Address that doesn't already have a DNS name, you must also specify a DNS name. You can add achieve this using the "Set-AzureRmPublicIpAddress" cmdlet:

```
PS C:\> $pip = Get-AzureRmPublicIpAddress -Name PublicIP -ResourceGroupName NRP-DemoRG-PS
PS C:\> $pip.DnsSettings = New-Object -TypeName
Microsoft.Azure.Commands.Network.Models.PSPublicIpAddressDnsSettings
PS C:\> $pip.DnsSettings.DomainNameLabel = "contosoapp1"
PS C:\> $pip.DnsSettings.ReverseFqdn = "contosoapp1.westus.cloudapp.azure.com."
PS C:\> Set-AzureRmPublicIpAddress -PublicIpAddress $pip
```

Create a Public IP Address with reverse DNS

You can add a new Public IP Address with the reverse DNS property specified using the "New-AzureRmPublicIpAddress" cmdlet:

```
PS C:\> New-AzureRmPublicIpAddress -Name PublicIP2 -ResourceGroupName NRP-DemoRG-PS -Location WestUS -
AllocationMethod Dynamic -DomainNameLabel "contosoapp2" -ReverseFqdn "contosoapp2.westus.cloudapp.azure.com."
```

View reverse DNS for existing Public IP Addresses

You can view the configured value for an existing Public IP Address using the "Get-AzureRmPublicIpAddress" cmdlet:

```
PS C:\> Get-AzureRmPublicIpAddress -Name PublicIP2 -ResourceGroupName NRP-DemoRG-PS
```

Remove reverse DNS from existing Public IP Addresses

You can remove a reverse DNS property from an existing Public IP Address using the "Set-AzureRmPublicIpAddress" cmdlet. This is done by setting the ReverseFqdn property value to blank:

```
PS C:\> $pip = Get-AzureRmPublicIpAddress -Name PublicIP -ResourceGroupName NRP-DemoRG-PS
PS C:\> $pip.DnsSettings.ReverseFqdn = ""
PS C:\> Set-AzureRmPublicIpAddress -PublicIpAddress $pip
```

FAQ - Hosting your ARPA zone in Azure DNS

Can I host ARPA zones for my ISP-assigned IP blocks on Azure DNS?

Yes. Hosting the ARPA zones for your own IP ranges in Azure DNS is fully supported.

Simply create the zone in Azure DNS, 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.

You can also import an existing reverse lookup zone using the Azure CLI.

How much does hosting my ARPA zone cost?

Hosting the ARPA zone for your ISP-assigned IP block in Azure DNS is charged at standard Azure DNS rates.

Can I host ARPA zones for both IPv4 and IPv6 addresses in Azure DNS?

Yes.

FAQ - Reverse DNS for your Azure-assigned IP address

How much do reverse DNS records cost?

They're free! There is no additional cost for reverse DNS records or queries.

Will the reverse DNS records for my Azure-assigned Public IP Address resolve from the internet?

Yes. Once you set the reverse DNS property for your Public IP Address, Azure manages all the DNS delegations and DNS zones required to ensure that reverse DNS record resolves for all internet users.

Will a default reverse DNS record be created for my Public IP Addresses?

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 (e.g., "app1.contoso.com.").

What happens if the validation checks for the reverse DNS I've specified fail?

Where the validation for reverse DNS checks fail, the service management operation will fail. Please correct the

reverse DNS value as required, and retry.

Can I manage reverse DNS for my Azure Website?

Reverse DNS is not supported for Azure Websites. Reverse DNS is supported for Azure Virtual Machines.

Can I configure multiple reverse DNS records for my Public IP Address?

No. Azure supports a single reverse DNS record for each Public IP Address. Each Public IP Address however can have their own reverse DNS record.

Can I configure reverse DNS records for an IPv6 Public IP Address?

No. At this time, reverse DNS records are supported for IPv4 Public IP Addresses only.

Can I configure a reverse DNS record for my Public IP Address without having a DomainNameLabel specified?

No. To leverage reverse DNS records for your Public IP Addresses, you must specify the DomainNameLabel property.

Can I send emails to external domains from my Azure Compute services?

No. Azure Compute services do not support sending emails to external domains.

How to manage reverse DNS records for your Azure services using the Azure CLI

1/17/2017 • 7 min to read • Edit on GitHub

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 will retrieve the reverse DNS record of the sending server's IP address, and verify if that host is authorized to send e-mail from the originating domain. (Please note however that Azure Compute services do not support sending emails to external domains.)

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 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.

For more information on reverse DNS, please see Reverse DNS Lookup.

Azure support for reverse DNS

Azure supports two separate scenarios relating to reverse DNS:

1. Hosting the ARPA zone corresponding to your IP address block.
2. Allowing you to configure the reverse DNS record for the IP address assigned to your Azure service.

To support the former, Azure DNS can be used to host your ARPA zones and manage the PTR records for each reverse DNS lookup. The process of creating the 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.

To support the latter, Azure enables you to configure the reverse lookup for the IP addresses allocated to your 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. **The remainder of this article describes this scenario in detail.**

> **NOTE**
> Azure has two different deployment models for creating and working with resources: Resource Manager and classic. This article covers using the Resource Manager deployment model, which Microsoft recommends for most new deployments instead of the classic deployment model.

For more information about the classic deployment model, see How to manage reverse DNS records for your Azure services (classic) using Azure PowerShell.

Validation of reverse DNS records

To ensure a third party can't create reverse DNS records mapping to your DNS domains, Azure only allows the creation of a reverse DNS record where one of the following is true:

- The "ReverseFqdn" is the same as the "Fqdn" for the Public IP Address resource for which it has been specified, or the "Fqdn" for any Public IP Address within the same subscription e.g., "ReverseFqdn" is "contosoapp1.northus.cloudapp.azure.com.".
- The "ReverseFqdn" forward resolves to the name or IP of the Public IP Address for which it has been specified, or to any Public IP Address "Fqdn" or IP within the same subscription e.g., "ReverseFqdn" is "app1.contoso.com." which is a CName alias for "contosoapp1.northus.cloudapp.azure.com."

Validation checks are only performed when the reverse DNS property for a Public IP Address is set or modified. Periodic re-validation is not performed.

Add reverse DNS to existing Public IP addresses

You can add reverse DNS to an existing Public IP address using the azure network public-ip set:

```
azure network public-ip set -n PublicIp -g NRP-DemoRG-PS -f contosoapp1.westus.cloudapp.azure.com.
```

If you wish to add reverse DNS to an existing Public IP Address that doesn't already have a DNS name, you must also specify a DNS name. You can add achieve this using the azure network public-ip set:

```
azure network public-ip set -n PublicIp -g NRP-DemoRG-PS -d contosoapp1 -f
contosoapp1.westus.cloudapp.azure.com.
```

Create a Public IP Address with reverse DNS

You can add a new Public IP Address with the reverse DNS property specified using the azure network public-ip create:

```
azure network public-ip create -n PublicIp3 -g NRP-DemoRG-PS -l westus -d contosoapp3 -f
contosoapp3.westus.cloudapp.azure.com.
```

View reverse DNS for existing Public IP Addresses

You can view the configured value for an existing Public IP Address using the azure network public-ip show:

```
azure network public-ip show -n PublicIp3 -g NRP-DemoRG-PS
```

Remove reverse DNS from existing Public IP Addresses

You can remove a reverse DNS property from an existing Public IP Address using azure network public-ip set. This is done by setting the ReverseFqdn property value to blank:

```
azure network public-ip set -n PublicIp3 -g NRP-DemoRG-PS –f “”
```

FAQ - Hosting your ARPA zone in Azure DNS

Can I host ARPA zones for my ISP-assigned IP blocks on Azure DNS?

Yes. Hosting the ARPA zones for your own IP ranges in Azure DNS is fully supported.

Simply create the zone in Azure DNS, 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.

You can also import an existing reverse lookup zone using the Azure CLI.

How much does hosting my ARPA zone cost?

Hosting the ARPA zone for your ISP-assigned IP block in Azure DNS is charged at standard Azure DNS rates.

Can I host ARPA zones for both IPv4 and IPv6 addresses in Azure DNS?

Yes.

FAQ - Reverse DNS for your Azure-assigned IP address

How much do reverse DNS records cost?

They're free! There is no additional cost for reverse DNS records or queries.

Will the reverse DNS records for my Azure-assigned Public IP Address resolve from the internet?

Yes. Once you set the reverse DNS property for your Public IP Address, Azure manages all the DNS delegations and DNS zones required to ensure that reverse DNS record resolves for all internet users.

Will a default reverse DNS record be created for my Public IP Addresses?

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 (e.g., "app1.contoso.com.").

What happens if the validation checks for the reverse DNS I've specified fail?

Where the validation for reverse DNS checks fail, the service management operation will fail. Please correct the reverse DNS value as required, and retry.

Can I manage reverse DNS for my Azure Website?

Reverse DNS is not supported for Azure Websites. Reverse DNS is supported for Azure Virtual Machines.

Can I configure multiple reverse DNS records for my Public IP Address?

No. Azure supports a single reverse DNS record for each Public IP Address. Each Public IP Address however can have their own reverse DNS record.

Can I configure reverse DNS records for an IPv6 Public IP Address?

No. At this time, reverse DNS records are supported for IPv4 Public IP Addresses only.

Can I configure a reverse DNS record for my Public IP Address without having a DomainNameLabel specified?

No. To leverage reverse DNS records for your Public IP Addresses, you must specify the DomainNameLabel property.

Can I send emails to external domains from my Azure Compute services?

No. Azure Compute services do not support sending emails to external domains.

How to manage reverse DNS records for your Azure services (classic) using Azure PowerShell

1/17/2017 • 6 min to read • Edit on GitHub

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 will retrieve the reverse DNS record of the sending server's IP address, and verify if that host is authorized to send e-mail from the originating domain. (Please note however that Azure Compute services do not support sending emails to external domains.)

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 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.

For more information on reverse DNS, please see Reverse DNS Lookup.

Azure support for reverse DNS

Azure supports two separate scenarios relating to reverse DNS:

1. Hosting the ARPA zone corresponding to your IP address block.
2. Allowing you to configure the reverse DNS record for the IP address assigned to your Azure service.

To support the former, Azure DNS can be used to host your ARPA zones and manage the PTR records for each reverse DNS lookup. The process of creating the 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.

To support the latter, Azure enables you to configure the reverse lookup for the IP addresses allocated to your 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. **The remainder of this article describes this scenario in detail.**

> **IMPORTANT**
> Azure has two different deployment models for creating and working with resources: Resource Manager and Classic. This article covers using the Classic deployment model. Microsoft recommends that most new deployments use the Resource Manager model. Learn how to perform these steps using the Resource Manager model.

Validation of reverse DNS records

To ensure a third party can't create reverse DNS records mapping to your DNS domains, Azure only allows the creation of a reverse DNS record where one of the following is true:

- The reverse DNS FQDN is the name of the Cloud Service for which it has been specified, or any Cloud Service name within the same subscription e.g., reverse DNS is "contosoapp1.cloudapp.net.".
- The reverse DNS FQDN forward resolves to the name or IP of the Cloud Service for which it has been specified, or to any Cloud Service name or IP within the same subscription e.g., reverse DNS is "app1.contoso.com." which is a CName alias for contosoapp1.cloudapp.net.

Validation checks are only performed when the reverse DNS property for a Cloud Service is set or modified. Periodic re-validation is not performed.

Add reverse DNS to existing Cloud Services

You can add a reverse DNS record to an existing Cloud Service using the "Set-AzureService" cmdlet:

```
PS C:\> Set-AzureService –ServiceName “contosoapp1” –Description “App1 with Reverse DNS” –ReverseDnsFqdn
“contosoapp1.cloudapp.net.”
```

Create a Cloud Service with reverse DNS

You can add a new Cloud Service with the reverse DNS property specified using the "Set-AzureService" cmdlet:

```
PS C:\> New-AzureService –ServiceName “contosoapp1” –Location “West US” –Description “App1 with Reverse DNS” –
ReverseDnsFqdn “contosoapp1.cloudapp.net.”
```

View reverse DNS for existing Cloud Services

You can view the configured value for an existing Cloud Service using the "Get-AzureService" cmdlet:

```
PS C:\> Get-AzureService "contosoapp1"
```

Remove reverse DNS from existing Cloud Services

You can remove a reverse DNS property from an existing Cloud Service using the "Set-AzureService" cmdlet. This is done by setting the reverse DNS property value to blank:

```
PS C:\> Set-AzureService –ServiceName “contosoapp1” –Description “App1 with Reverse DNS” –ReverseDnsFqdn “”
```

FAQ - Hosting your ARPA zone in Azure DNS

Can I host ARPA zones for my ISP-assigned IP blocks on Azure DNS?

Yes. Hosting the ARPA zones for your own IP ranges in Azure DNS is fully supported.

Simply create the zone in Azure DNS, 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.

You can also import an existing reverse lookup zone using the Azure CLI.

How much does hosting my ARPA zone cost?

Hosting the ARPA zone for your ISP-assigned IP block in Azure DNS is charged at standard Azure DNS rates.

Can I host ARPA zones for both IPv4 and IPv6 addresses in Azure DNS?

Yes.

FAQ - Reverse DNS for your Azure-assigned IP address

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 Cloud Service, Azure manages all the DNS delegations and DNS zones required to ensure that reverse DNS record resolves for all internet users.

Will a default reverse DNS record be created for my Cloud 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 (e.g., "app1.contoso.com.").

What happens if the validation checks for the reverse DNS I've specified fail?

Where the validation for reverse DNS checks fail, the service management operation will fail. Please correct the reverse DNS value as required, and retry.

Can I manage reverse DNS for my Azure Website?

Reverse DNS is not supported for Azure Websites. Reverse DNS is supported for Azure PaaS roles and IaaS virtual machines.

Can I configure multiple reverse DNS records for my Cloud Service?

No. Azure supports a single reverse DNS record for each Azure Cloud Service. Each Azure Cloud Service however can have their own reverse DNS record.

Can I send emails to external domains from my Azure Compute services?

No. Azure Compute services do not support sending emails to external domains.

Import and export a DNS zone file using the Azure CLI

1/17/2017 • 8 min to read • Edit on GitHub

This article will walk 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 particularly 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 will need to obtain a copy of the zone file. The source of this file will depend 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 service management 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**.

Working with zone files from GoDaddy

Zone files downloaded from GoDaddy have a slightly nonstandard format. You need to correct this before you import these zone files into Azure DNS. DNS names in the RData of each DNS record are specified as fully qualified names, but they don't have a terminating "." This means they are interpreted by other DNS systems as relative names. You need to edit the zone file to append the terminating "." to their names before you import them into Azure DNS.

Import a DNS zone file into Azure DNS

Importing a zone file will create 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.
- Alternatively, by specifying the `--force` option, the import process will replace existing record sets with new

record sets. Existing record sets that do not have a corresponding record set in the imported zone file will not be removed.

- When record sets are merged, the time to live (TTL) of preexisting record sets is used. When `--force` is used, the TTL of the new record set is used.
- Start of Authority (SOA) parameters (except `host`) are always taken from the imported zone file, regardless of whether `--force` is used. 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 will not replace an existing CNAME record with the same name unless the `--force` parameter is specified.
- 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. This is independent of the use of `--force`.

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 will be 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, 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 will be concatenated and truncated to 255 characters.

CLI format and values

The format of the Azure CLI command to import a DNS zone is:

```
azure network dns zone import [options] <resource group> <zone name> <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 will be created for you. If the zone already exists, the imported record sets will be merged with existing record sets. To overwrite the existing record sets, use the `--force` option.

To verify the format of a zone file without actually importing it, use the `--parse-only` option.

Step 1. Import a zone file

To import a zone file for the zone **contoso.com**.

1. Sign in to your Azure subscription by using the Azure CLI.

   ```
   azure login
   ```

2. Select the subscription where you want to create your new DNS zone.

   ```
   azure account set <subscription name>
   ```

3. Azure DNS is an Azure Resource Manager-only service, so the Azure CLI must be switched to Resource Manager mode.

   ```
   azure config mode arm
   ```

4. Before you use the Azure DNS service, you must register your subscription to use the Microsoft.Network resource provider. (This is a one-time operation for each subscription.)

   ```
   azure provider register Microsoft.Network
   ```

5. If you don't have one already, you also need to create a Resource Manager resource group.

   ```
   azure group create myresourcegroup westeurope
   ```

6. To import the zone **contoso.com** from the file **contoso.com.txt** into a new DNS zone in the resource group **myresourcegroup**, run the command `azure network dns zone import`.
   This command will load the zone file and parse it. The command will execute a series of commands on the Azure DNS service to create the zone and all of the record sets in the zone. The command will also report 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.

   ```
   azure network dns zone import myresourcegroup contoso.com 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.

  ```
   azure network dns record-set list myresourcegroup contoso.com
  ```

- You can list the records by using the PowerShell cmdlet `Get-AzureRmDnsRecordSet`.
- You can use `nslookup` to verify name resolution for the records. Because the zone isn't delegated yet, you will need to specify the correct Azure DNS name servers explicitly. The sample below shows how to retrieve the name server names assigned to the zone. IT also shows how to query the "www" record by using `nslookup`.

```
C:\>azure network dns record-set show myresourcegroup contoso.com @ NS
info:Executing command network dns record-set show
+ Looking up the DNS Record Set "@" of type "NS"
data:Id:
/subscriptions/.../resourceGroups/myresourcegroup/providers/Microsoft.Network/dnszones/contoso.com/NS/@
data:Name: @
data:Type: Microsoft.Network/dnszones/NS
data:Location: global
data:TTL : 3600
data:NS records
data:Name server domain name : ns1-01.azure-dns.com
data:Name server domain name : ns2-01.azure-dns.net
data:Name server domain name : ns3-01.azure-dns.org
data:Name server domain name : ns4-01.azure-dns.info
data:
info:network dns record-set show command OK

C:\> nslookup www.contoso.com ns1-01.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 will 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 import a DNS zone is:

```azurecli
azure network dns zone export [options] <resource group> <zone name> <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

1. Sign in to your Azure subscription by using the Azure CLI.

```
azure login
```

2. Select the subscription where you want to create your new DNS zone.

```
azure account set <subscription name>
```

3. Azure DNS is an Azure Resource Manager-only service. The Azure CLI must be switched to Resource Manager mode.

```
azure config mode arm
```

4. 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 will call the Azure DNS service to enumerate record sets in the zone and export the results to a BIND-compatible zone file.

```
azure network dns zone export myresourcegroup contoso.com contoso.com.txt
```

Using Azure DNS with other Azure services

1/17/2017 • 2 min to read • Edit on GitHub

Azure DNS is a hosted DNS management and name resolution service. This allows you to create public DNS names for the other applications and services you have deployed in Azure. Creating a name for an Azure service in your custom domain is as simple as adding a record of the correct type for your service.

- For dynamically allocated IP addresses, you must 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.
- For statically allocated IP addresses, you can create a DNS A record using any name, including a *naked domain* name at the zone apex.

The following table outlines the supported record types that can be used for various Azure services. As you can see from this table, Azure DNS only supports DNS records for Internet-facing network resources. Azure DNS cannot be used for name resolution of internal, private addresses.

AZURE SERVICE	NETWORK INTERFACE	DESCRIPTION
Application Gateway	Front-end Public IP	You can create a DNS A or CNAME record.
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 is dynamically assigned. Therefore, you must create a CNAME record for an IPv6 address.
Traffic Manager	Public name	You can only create a CNAME that maps to the trafficmanager.net name assigned to your Traffic Manager profile. For more information, see How Traffic Manager works.
Cloud Service	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. This rule applies to VMs created in the classic portal because they are deployed as a cloud service. For more information, see Configure a custom domain name in Cloud Services.
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
Resource Manager VMs	Public IP	Resource Manager VMs can have Public IP addresses. A VM with a Public IP address may also be behind a load balancer. You can create a DNS A or CNAME record for the Public address. This custom name can be used to bypass the VIP on the load balancer.
Classic VMs	Public IP	Classic VMs created using PowerShell or CLI can be configured with a dynamic or static (reserved) virtual address. You can create a DNS CNAME or A record, respectively.

How to protect DNS zones and records

1/17/2017 • 9 min to read • Edit on GitHub

DNS zones and records are critical resources. Deleting a DNS zone or even just a single DNS record can result in a total service outage. It is therefore important that critical DNS zones and records are protected against unauthorized or accidental changes.

This article explains how Azure DNS enables you to protect your DNS zones and records against such changes. We apply two powerful security features provided by Azure Resource Manager: role-based access control and resource locks.

Role-based access control

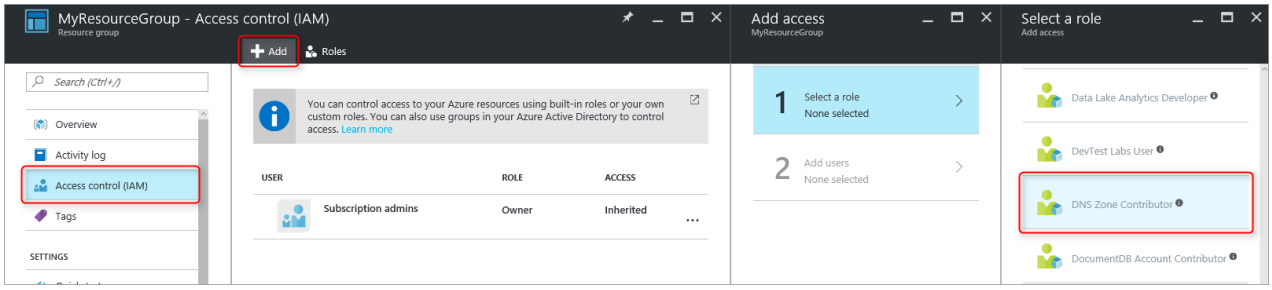
Azure Role-Based Access Control (RBAC) enables fine-grained access management for Azure users, groups, and resources. Using RBAC, you can grant precisely the amount of access that users need to perform their jobs. For more information about how RBAC helps you manage access, see What is Role-Based Access Control.

The 'DNS Zone Contributor' role

The 'DNS Zone Contributor' role is a built-in role provided by Azure for managing DNS resources. Assigning DNS Zone Contributor permissions to a user or group enables that group to manage DNS resources, but not resources of any other type.

For example, suppose the resource group 'myzones' 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 also avoids granting unnecessary permissions, for example the DNS administrator cannot create or stop Virtual Machines.

The simplest way to assign RBAC permissions is via the Azure portal. Open the 'Access control (IAM)' blade for the resource group, then click '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 all zones in a resource group
New-AzureRmRoleAssignment -SignInName <user email address> -RoleDefinitionName "DNS Zone Contributor" -
ResourceGroupName <resource group name>
```

The equivalent command is also available via the Azure CLI:

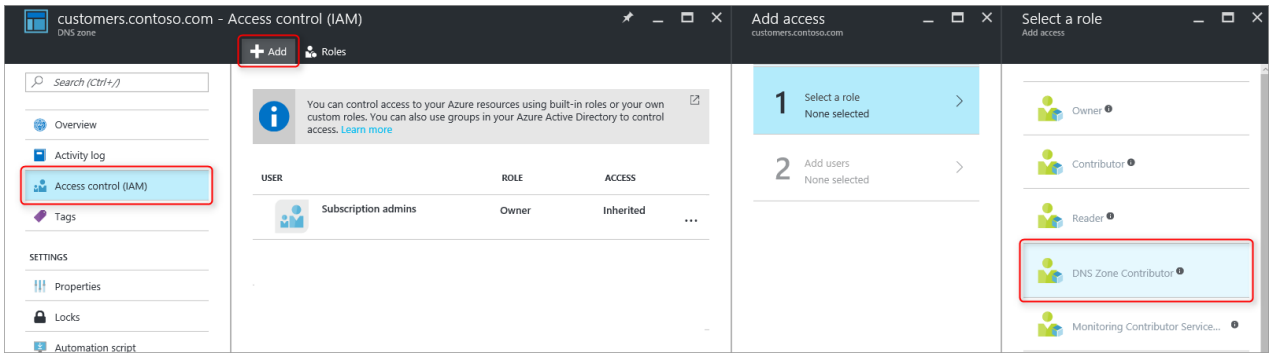
```
# Grant 'DNS Zone Contributor' permissions to all zones in a resource group
azure role assignment create --signInName  <user email address> --roleName "DNS Zone Contributor" --
resourceGroup <resource group name>
```

Zone level RBAC

Azure RBAC rules can be applied to a subscription, a resource group or to an individual resource. In the case of Azure DNS, that resource can be an individual DNS zone, or even an individual record set.

For example, suppose the resource group 'myzones' contains the zone 'contoso.com' and a subzone 'customers.contoso.com' in which CNAME records are created for each customer account. The account used to manage these CNAME records should be assigned permissions to create records in the 'customers.contoso.com' zone only, it should not have access to the other zones.

Zone-level RBAC permissions can be granted via the Azure portal. Open the 'Access control (IAM)' blade for the zone, then click '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
New-AzureRmRoleAssignment -SignInName <user email address> -RoleDefinitionName "DNS Zone Contributor" -
ResourceGroupName <resource group name> -ResourceName <zone name> -ResourceType Microsoft.Network/DNSZones
```

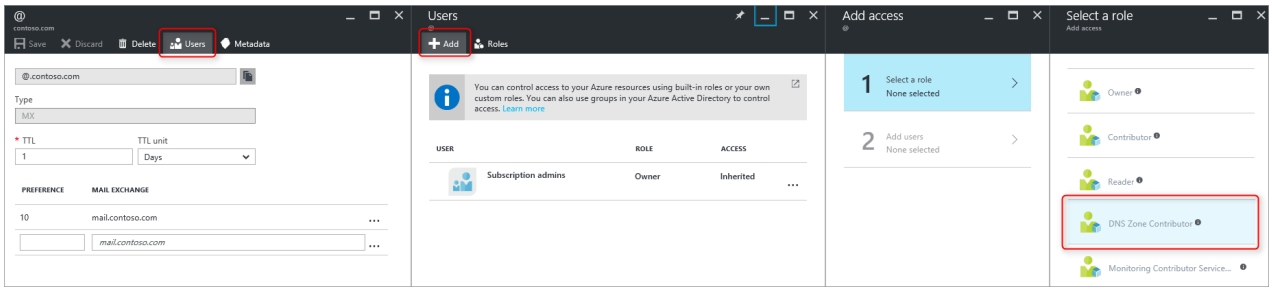
The equivalent command is also available via the Azure CLI:

```
# Grant 'DNS Zone Contributor' permissions to a specific zone
azure role assignment create --signInName <user email address> --roleName "DNS Zone Contributor" --resource-name
<zone name> --resource-type Microsoft.Network/DNSZones --resource-group <resource group name>
```

Record set level RBAC

We can go one step further. Consider the mail administrator for Contoso Corporation, who needs access to the MX and TXT records at the apex of the 'contoso.com' zone. She doesn't need access to any other MX or TXT records, or to any records of any other type. Azure DNS allows you to assign permissions at the record set level, to precisely the records that the mail administrator needs access to. The mail administrator is granted precisely the control she needs, and is unable to make any other changes.

Record-set level RBAC permissions can be configured via the Azure portal, using the 'Users' button in the record set blade:



Record-set level RBAC permissions can also be granted using Azure PowerShell:

```
# Grant permissions to a specific record set
New-AzureRmRoleAssignment -SignInName <user email address> -RoleDefinitionName "DNS Zone Contributor" -Scope
"/subscriptions/<subscription id>/resourceGroups/<resource group
name>/providers/Microsoft.Network/dnszones/<zone name>/<record type>/<record name>"
```

The equivalent command is also available via the Azure CLI:

```
# Grant permissions to a specific record set
azure role assignment create --signInName <user email address> --roleName "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 is also possible to build your own customer Azure roles, to provide even finer-grained control.

Consider again the example in which a CNAME record in the zone 'customers.contoso.com' is created for each Contoso Corporation customer account. The account used to manage these CNAMEs should be granted permission to manage CNAME records only. It is then unable to modify records of other types (such as changing MX records) or perform 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 cannot 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-AzureRmRoleDefinition -InputFile <file path>
```

It can also be created via the Azure CLI:

```
# Create new role definition based on input file
azure 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

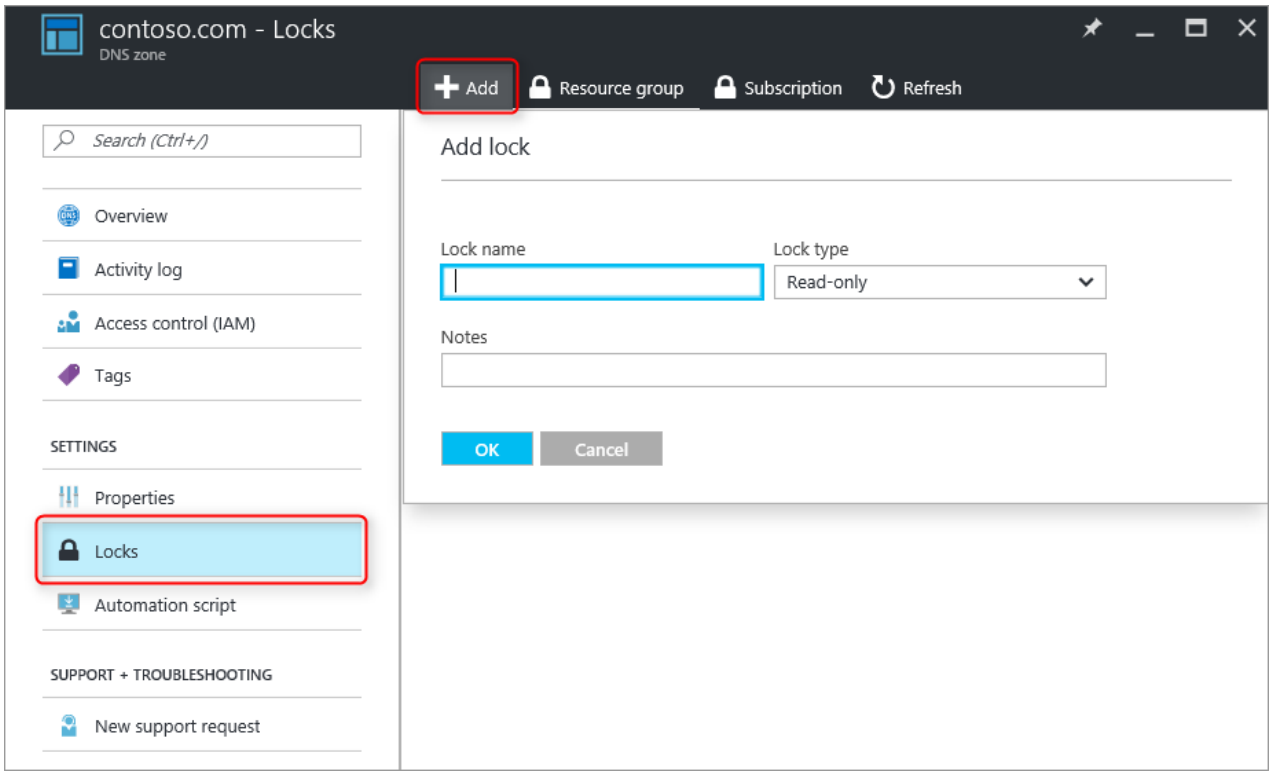
In addition to RBAC, Azure Resource Manager supports another type of security control, namely the ability to 'lock' resources. Where RBAC rules allow you to control the actions of specific users and groups, resource locks are applied to the resource, and are effective across all users and roles. For more information, see Lock resources with Azure Resource Manager.

There are two types of resource lock: **DoNotDelete** and **ReadOnly**. These can be applied either to a 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 any changes being made, apply a ReadOnly lock to the zone. This 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 blade, click 'Locks', then 'Add':



Zone-level resource locks can also be created via Azure PowerShell:

```
# Lock a DNS zone
New-AzureRmResourceLock -LockLevel <lock level> -LockName <lock name> -ResourceName <zone name> -ResourceType
Microsoft.Network/DNSZones -ResourceGroupName <resource group name>
```

Configuring Azure resource locks is not currently supported via the Azure CLI.

Protecting individual records

To prevent an existing DNS record set against modification, apply a ReadOnly lock to the record set.

> **NOTE**
> Applying a DoNotDelete 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 are not supported in the Azure portal or Azure CLI.

```
# Lock a DNS record set
New-AzureRmResourceLock -LockLevel <lock level> -LockName <lock name> -ResourceName <zone name>/<record set
name> -ResourceType Microsoft.Network/DNSZones/<record type> -ResourceGroupName <resource group name>
```

Protecting against zone deletion

When a zone is deleted in Azure DNS, all record sets in the zone are also deleted. This operation cannot be undone. Accidentally deleting a critical zone has the potential to have a significant business impact. It is therefore very important to protect against accidental zone deletion.

Applying a DoNotDelete lock to a zone prevents the zone from being deleted. However, since locks are inherited by child resources, it also prevents any record sets in the zone from being deleted, which may be undesirable. Furthermore, as described in the note above, it is also ineffective since records can still be removed from the existing record sets.

As an alternative, consider applying a DoNotDelete lock to a record set in the zone, such as the SOA record set. Since the zone cannot be deleted without also deleting the record sets, this 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 would also delete the SOA record set, and blocks the call because the SOA is locked. No record sets are deleted.

The following PowerShell command creates a DoNotDelete lock against the SOA record of the given zone:

```
# Protect against zone delete with DoNotDelete lock on the record set
New-AzureRmResourceLock -LockLevel DoNotDelete -LockName <lock name> -ResourceName <zone name>/@ -ResourceType
Microsoft.Network/DNSZones/SOA -ResourceGroupName <resource group name>
```

Another way to prevent accidental zone deletion is by using a custom role to ensure the operator and service accounts used to manage your zones do not have zone delete permissions. When you do need to delete a zone, you can enforce a two-step delete, first granting zone delete permissions (at the zone scope, to prevent deleting the wrong zone) and second to delete the zone.

This second approach has the advantage that it works for all zones accessed by those accounts, without having to remember to create any locks. It has the disadvantage that any accounts with zone delete permissions, such as the subscription owner, can still accidentally delete a critical zone.

It is 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.
- For more information about securing your Azure resources, see Security considerations for Azure Resource Manager.

Create DNS zones and record sets using the .NET SDK

1/17/2017 • 6 min to read • Edit on GitHub

You can automate operations to create, delete, or update DNS zones, record sets, and records by using DNS SDK with .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** in 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 recordsets. The following code logs in to 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 a 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 of how to use the Azure DNS .NET SDK, including examples for other DNS record types.