

This page shows how to configure the DNS settings for a domain using Cloud DNS and Cloud Tools for PowerShell. It walks through a simple example of creating a managed zone to govern a domain and its subdomains, and then adding resource records to the zone to provide information that determines the behavior of the DNS server when handling requests to the zone's domains.

This document assumes you have a domain name and an IP address that you can point the domain name to. If you do not, you can register a domain name through [Google Domains](https://domains.google/) (https://domains.google/) or another domain registrar of your choice.

Read the [Cloud Tools for PowerShell cmdlet reference](http://googlecloudplatform.github.io/google-cloud-powershell/)

(http://googlecloudplatform.github.io/google-cloud-powershell/) to learn more about Cloud DNS cmdlets. To learn more about Cloud DNS in general, read the [Overview of Cloud DNS](/dns/overview) (/dns/overview).

The entirety of the DNS namespace is composed of many domains, which will soon include your own domain name. Managed zones in Cloud DNS model DNS zones and serve as containers to organize DNS records (such as A, CNAME, or TXT entries) for the same DNS name suffix. For example, the records for "example.com." and subdomains such as "first.example.com." could be in the same zone since they share the "example.com." suffix. Note the trailing dot, which is necessary and signifies an absolute DNS name.

To get started, set up a managed zone to organize the DNS records that you will create for your domain name. You can create a new managed zone and add it to your Cloud Console project by using the `Add-GcdManagedZone` cmdlet:

This creates a new zone with the specified details and adds it to the current project of the active [Cloud SDK configuration](/sdk/docs/configurations) (/sdk/docs/configurations), though you also have the option to specify a different project ID if desired. This additionally creates [default NS and SOA records](/dns/records/json-record) (/dns/records/json-record) in the zone for you.

However, to publish your new records in the zone to the internet, you also need to [update your domain's name servers to use Cloud DNS](/dns/update-name-servers) (/dns/update-name-servers). Even if your domain is registered with Google Domains, you still need to [update its name servers](https://support.google.com/domains/answer/3290309) (https://support.google.com/domains/answer/3290309).

You can find the Cloud DNS name servers assigned to your domain by using the `Get-GcdManagedZone` cmdlet on the managed zone governing the domain to return information about the zone:

[DNS resource records](/dns/records/json-record) (/dns/records/json-record) provide information that dictates the behavior of the DNS server when it is handling requests sent to a domain. For example, DNS records can be used to tell the server which IP address a domain resolves to, indicate the usable mail exchange servers for a domain, and much more.

In Cloud DNS, you can add or remove DNS records in a zone to configure such behavior. To add or remove immutable resource record sets, you do not operate on the records in a zone directly. Rather, you create independent resource records or retrieve existing ones using the `New-GcdResourceRecordSet` and `Get-GcdResourceRecordSet` cmdlets respectively, and then send change requests with these records to a specific zone by using the `Add-GcdChange` cmdlet.

You can use the helper cmdlet `New-GcdResourceRecordSet` to create a resource record set that you can put within a change and then add to or remove from a managed zone.

For example, if you wanted to create an A record to point your domain to an external IPv4 address in the format `###.###` (if you have an IPv6 address, use an AAAA record), you could use the following command:

Similarly, to create a CNAME record for the `www` subdomain such that `"www.example.com."` resolves to the same IP and behaves the same as `"example.com."` you could use the following command:

The supported resource record types that you can create and include in changes to a zone are A, AAAA, CNAME, MX, NAPTR, NS, PTR, SOA, SPF, SRV, and TXT. For help deciding which records you need and how to create them, see [supported resource record formats for Cloud DNS \(/dns/records/json-record\)](#).

If you want to retrieve an existing resource record set in a zone, you can use the `Get-GcdResourceRecordSet` cmdlet to return all records in a zone and then index the results:

If you only want records of a specific type, you can filter the results accordingly:

The cmdlets `New-GcdResourceRecordSet` and `Get-GcdResourceRecordSet` both return resource record sets, but they do not add records to or remove them from anything. For that, use the `Add-GcdChange` cmdlet:

For example, to add the A and CNAME records that you created above to our zone, do the following:

If you later wanted to change the IPv4 address that your domain resolves to, you could create a new A-type record and then add it to the managed zone while deleting the old A record:

Each call to the `Add-GcdChange` cmdlet returns the newly executed change request object (`/dns/api/v1/changes`). You can also pass the `Add-GcdChange` cmdlet a change request object to execute directly instead of lists of resource record sets:

Creating change request objects manually is not recommended, but they are useful if you want to re-apply a change made earlier or one made in a different zone. You can retrieve all past change request objects that were applied to a zone by using the `Get-GcdChange` cmdlet:

You can choose a specific change by indexing the previous result, or specify which change to retrieve with a `ChangeId`. Change requests are usually numbered from 0 onwards based on the order they were sent to the zone:

Sometimes, you may want to remove a managed zone altogether. This could be for a variety of reasons. Perhaps you have managed zone "user1-zone" for subdomain "user1.example.com." but user1 deletes their account, in which case you want to remove this subdomain and all associated DNS records.

To remove a managed zone from your project, use the `Remove-GcdManagedZone` cmdlet:

If successful, the command returns nothing.

However, this cmdlet does not immediately work on what Cloud DNS considers "non-empty" managed zones, or zones that contain non-NS or non-SOA type records (non-default records). For example, you would not be able to delete "my-new-zone," to which you added A and CNAME type records, without either granting permission to the cmdlet during processing or using the `-Force` switch:

The `Remove-GcdManagedZone` cmdlet also accepts pipeline input for the zone(s) to delete. For example, the following command forcibly deletes all the managed zones in the current project. This might be useful if you are no longer maintaining any of the domains set up in a project: