

# Migrating to Cloud DNS

Cloud DNS supports the migration of an existing DNS domain from another DNS provider to Cloud DNS. This procedure describes how to complete the necessary steps: creating a managed zone for your domain, importing your existing DNS configuration, and updating your registrar's name server records.

## Before you begin

If you have not used the `gcloud` command-line tool before, first [set up `gcloud` SDK](/compute/docs/gcloud-compute) (`/compute/docs/gcloud-compute`).

Next, run the following command to specify the project name and authenticate with the Cloud Console:

```
gcloud auth login
```

You can also specify the `--project` parameter for a command to operate against a different project for that invocation.

## Step 1: Create a managed zone

To migrate an existing domain, first create a managed zone to contain your DNS records. Note that when you create a zone, the new zone won't be used until you update your domain registration, explicitly point some resolver at it, or directly query one of your zone's name servers.

### `gcloud`

To create a zone, use the following command:

```
gcloud dns managed-zones create --dns-name=example.com.  
--description=A zone examplezonename
```

Replace the following command options:

- ***example.com.***: The DNS name
- ***A zone***: A description of the zone
- ***examplezonename***: The name to identify the DNS zone

## Step 2: Export your DNS configuration from your existing provider

Note that you must consult your provider's documentation to learn how to export your zone file ([https://wikipedia.org/wiki/Zone\\_file](https://wikipedia.org/wiki/Zone_file)). Cloud DNS supports the import of zone files in BIND or YAML records format.

For example:

- For Dyn (<https://www.oracle.com/corporate/acquisitions/dyn/>), go to Download Your Zone File (<https://help.dyn.com/dns-knowledge-base/download-your-zone-file/>).
- For AWS Route 53 (<https://aws.amazon.com/route53/>), which does not support export, you can use the open source cli53 (<https://github.com/barnybug/cli53>) tool.

## Step 3: Import the record set

Once you have the exported file from your other provider, you can use `gcloud` commands to import it into your managed zone.

To import record sets correctly, you must remove the apex records or use the flags described in the `gcloud` tab.

### `gcloud`

To import record-sets, use the `dns record-sets import` command. The `--zone-file-format` flag tells `import` to expect a BIND zone formatted file. If you omit this flag, `import` expects a YAML-formatted records file:

```
gcloud dns record-sets import -z=examplezonename  
--zone-file-format path-to-example-zone-file
```

Replace the following command option:

- **examplezonename**: The name of your DNS zone

**Caution:** Cloud DNS If your import file contains NS or SOA records for the apex of the zone, they will conflict with the pre-existing Cloud DNS records. In order to use the pre-existing Cloud DNS records (recommended), ensure that the NS or SOA records are removed from your import file. However, there are use cases for overriding this behavior; see the important information below.

**Caution:** If your authoritative DNS is split across multiple providers and you have a non-Cloud DNS primary name server, then you must replace the Cloud DNS SOA record with the one from the other provider. To do this, you must use the **--delete-all-existing** flag when importing record sets in order to replace the SOA records provided by Cloud DNS. Otherwise, the update fails, since the imported records conflict with the pre-existing Cloud DNS records. For similar reasons, you can specify that the NS records in the import file be used instead of the pre-existing Cloud DNS records by using the **--delete-all-existing** and **--replace-origin-ns** flags together. Note that specifying an NS record for the apex of a zone results in an error even if the **--replace-origin-ns** flag is not specified. Either remove these records from the import file or use both the **--delete-all-existing** and **--replace-origin-ns** flags together if appropriate.

For detailed information on using the `dns record-sets import` command, see the [command reference page](/sdk/gcloud/reference/dns/record-sets/import) (/sdk/gcloud/reference/dns/record-sets/import).

**Note:** Some DNS implementations and providers export BIND zone files without final periods on domain name data in CNAME, MX, PTR, and other records. In zone files Cloud DNS follows RFC standards and interprets all domain names without a final period as relative to the DNS name of the zone, so importing the following MX records into a zone with the DNS name **example.com** results in identical (and probably undesired) records for both:

```
in.smtp                IN MX 5 gmail-smtp-in.1.google.com  
in.smtp.example.com.  IN MX 5 gmail-smtp-in.1.google.com.example.com.
```

Check your zone files to ensure all names that need them have final periods before importing them.

## Step 4: Verify DNS propagation

You can use the Linux `watch` and `dig` commands to monitor and verify that your changes have been picked up by the Cloud DNS name servers.

The `watch` and `dig` commands are not `gcloud` commands and are not used with the `gcloud` prefix. On non-Linux systems, you might need to install the `watch` and `dig` commands.

1. Look up your zone's Cloud DNS name servers:

```
gcloud dns managed-zones describe examplezonename
```

Replace the following command option:

- ***examplezonename***: The name of your DNS zone

The output looks something like this:

```
nameServers:  
- ns-cloud-a1.googledomains.com.  
- ns-cloud-a2.googledomains.com.  
- ns-cloud-a3.googledomains.com.  
- ns-cloud-a4.googledomains.com.
```

In the output, the letter following the "ns-cloud-" part of the name is referred to as the name server *shard*. As documented [here](https://cloud.google.com/dns/quotas#nameserver_limits) ([https://cloud.google.com/dns/quotas#nameserver\\_limits](https://cloud.google.com/dns/quotas#nameserver_limits)), there are five such shards (letters A-E).

2. Check if the records are available on the name servers.

```
watch dig example.com @your_zone_nameserver
```

Replace **`your_zone_nameserver`** with one of the name servers returned when you ran the previous command.

3. Once you see your change, press `Ctrl-C` to exit.

The `watch` command runs the `dig` command every 2 seconds by default. You can use this command to determine when your authoritative name server picks up your change, which should happen within 120 seconds.

## Step 5: Update your registrar's name server records

Log into your registrar provider and change the authoritative name servers to point to the name servers you saw in step 4. At the same time, make a note of the time to live (TTL) your registrar has set on the records. That tells you how long you have to wait before the new name servers begin to be used.

## Step 6: Wait for changes, then verify

To get the authoritative name servers for your domain on the Internet, run the following Linux commands:

```
short NS example.com
```

If the output shows that all changes have propagated, you're done. If not, you can check intermittently or you can automatically run the command every 2 seconds while you wait for the name servers to change. To do that, run the following:

```
dig +short NS example.com
```

`Ctrl-C` exits the command.

If you're not using Linux, you can use the [nslookup command](https://wikipedia.org/wiki/Nslookup) (<https://wikipedia.org/wiki/Nslookup>)

## Next steps

- [Cloud DNS Overview \(/dns/overview\)](/dns/overview)
- [gcloud command-line tool \(/sdk/gcloud/reference/dns\)](/sdk/gcloud/reference/dns)
- [Managing Records \(/dns/records\)](/dns/records)
- [Records Format \(JSON\) \(/dns/records/json-record\)](/dns/records/json-record)

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (https://www.apache.org/licenses/LICENSE-2.0). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (https://developers.google.com/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2020-07-09 UTC.