

## CHAPTER 13



# Google Cloud DNS

The domain name system (DNS) is a global hierarchical distributed database that stores the mapping between a domain name and an IP address or other associated details such as the hostname. This mapping is called a *DNS record*, and associated records are stored in a *DNS zone*. From this perspective, the DNS zone can be seen as a container for DNS records that have the same suffix in a domain name. For example, [db.cloudplatformbook.com](https://db.cloudplatformbook.com) and [web.cloudplatformbook.com](https://web.cloudplatformbook.com) are two DNS records in the [cloudplatformbook.com](https://cloudplatformbook.com) DNS Zone.

Google Cloud DNS is a globally distributed, low-latency, high-performance, resilient, authoritative domain name service. As with other Google Cloud Platform products, Cloud DNS is also available via a REST API, the `gcloud` command-line tool, and the web-based graphical Developers Console.

Cloud DNS consists of managed zones and a resource record sets collection; these serve as a one-to-one mapping between DNS zones and records. All records in a managed zone are served by the same set of Cloud DNS servers. Cloud DNS uses anycast to serve the managed zones from many redundant locations around the world. The benefits of this architecture are high availability and low latency for applications and users alike.

Cloud DNS supports the following types of DNS records:

- *SOA*: Start of authority record, which specifies authoritative information on a DNS zone. Cloud DNS automatically creates an SOA resource record for every managed zone.
- *NS*: Name server record, which delegates a DNS zone to an authoritative DNS server.
- *A*: IPv4 address record, which maps host names to their IPv4 address.
- *AAAA*: IPv6 address record, which maps host names to their IPv6 address.
- *PTR*: Pointer record, used to create reverse DNS lookups—that is, to map IP addresses to host names.
- *CNAME*: Canonical name record, used to create alias names.
- *MX*: Mail exchange record, used to identify mail servers for the domain.
- *SPF*: Sender policy framework record, used to state the authoritative e-mail servers for a domain.
- *TXT*: Text records, which can be used to publish arbitrary human- and machine-readable information.
- *SRV*: Service location record, used by some VOIP and instant-messaging protocols and applications.
- *NAPTR*: Name authority pointer records, used in Internet telephony applications and often combined with SRV records in a chained fashion.

Each record can be seen as a key-value pair. A hostname, when used as the value of a record, should be a fully qualified DNS name (terminated with a period [.]). Cloud DNS supports wildcard records for all record types except NS.

Two entities play a key role in determining how long a change takes to reach a user: an authoritative DNS server and a DNS resolver. Cloud DNS is an authoritative DNS server and publishes the DNS zones and records. Change requests are usually applied within a minute and consist of two transparent steps:

1. Updates to the central control plane are applied synchronously (the client sends a write and then can read it back immediately in an API call).
2. Updates are replicated to the global serving plane asynchronously, normally in less than a minute.

The user is usually connected to a DNS resolver that caches the DNS zones and records of various domains. The cache validity is controlled by a parameter called time to live (TTL) that is associated with each DNS record.

TTL is a DNS record attribute that specifies how long a DNS record should be cached by the querying system, such as the end user system or intermediate DNS resolver. TTL is represented in seconds, and all systems outside of the authoritative name server should use it as a countdown timer. The DNS standard further states that a DNS resolver should discard an expired cached DNS record and retrieve it again from the DNS server. Setting a high TTL value means there is less traffic to the DNS server because the DNS resolver caches records longer. The negative aspect of a high TTL value is that an update to a DNS record takes much longer to propagate to the user. In addition, some DNS resolvers do not honor TTL values and use a fixed refresh interval or time to fetch updates. It is therefore recommended that during the system design and development stages, you keep TTL values low so that changes propagate faster (ideally). Once the system is in production, you can send a change request to Cloud DNS and set a high TTL value to reduce queries to the DNS server and save costs.

If you or your customers observe that the current DNS resolver is not following the standard and is fetching DNS records in an arbitrary fashion, you may want to consider using the DNS resolver from Google: Google Public DNS (<https://developers.google.com/speed/public-dns>). Google Public DNS is a fast, free, global DNS resolution service that can be used in place of your current DNS resolver (typically your ISP). The change is easy to make: all you have to do is set your primary and secondary DNS resolver IPs to 8.8.8.8 and 8.8.4.4. One downside of using the Google Public DNS resolver in an enterprise is that you may not be able to resolve some corporate web sites if they are served by an internal DNS server.

## Publishing Your Internet Identity

This section walks through the complete process of setting up DNS for a web site using Cloud DNS. You use a real domain—the domain for this book, [cloudplatformbook.com](http://cloudplatformbook.com)—in this walkthrough. The examples use the `gcloud` command-line tool.

The first step in setting up DNS for a domain using Cloud DNS is to create a managed zone for the DNS zone that you would like to host in Cloud DNS. The DNS zone in this case is [cloudplatformbook.com](http://cloudplatformbook.com). Listing 13-1 shows the `gcloud` command to create this managed zone.

**Listing 13-1.** `gcloud` DNS Command to Create a Managed Zone

```
$ gcloud dns managed-zones create cloudplatformbook \
  --description='Managed Zone for Google Cloud Platform Book' --dns-name='cloudplatformbook.com'
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/
managedZones/cloudplatformbook].
```

NAME	DNS_NAME	DESCRIPTION
cloudplatformbook	cloudplatformbook.com.	Managed Zone for Google Cloud Platform Book

Let's dissect this command and the response from Cloud DNS to understand the transaction. Just like other cloud platform products, Cloud DNS commands in `gcloud` are housed as subcommands under `gcloud dns <options>`. You can at any time use the `--help` option to get a list of the commands available at that hierarchy. For example, if you run the command `gcloud dns --help`, you get all the options available at the highest hierarchy level.

To create a managed zone, you must provide the following parameters:

- *DNS name:* The DNS zone suffix that this managed zone will publish on the Internet. You should own this domain or at least have administrative access to a domain registrar panel, because you make changes later to make Cloud DNS the official DNS server.
- *Description:* A Cloud DNS-specific field for your reference only. It is recommended that you provide meaningful data to help you remember what this managed zone is associated with.
- *Zone Name:* A Cloud DNS-specific field that must be unique within a project. The name cannot contain any spaces and is a required field in subsequent API requests. By convention, it usually matches the DNS zone name, as in this example.

Once the command in Listing 13-1 has been issued, Cloud DNS creates a managed-zone object for the DNS zone specified. It returns a formatted result, which restates parts of the creation request. You can retrieve the same information anytime by using the `list` command to list all the DNS zones managed by Cloud DNS in your project. Listing 13-2 shows the command and the corresponding result.

**Listing 13-2.** `gcloud` DNS Command to List Managed Zones

```
$ gcloud dns managed-zones list
```

```
NAME                DNS_NAME                DESCRIPTION
cloudplatformbook   cloudplatformbook.com.  Managed Zone for Google Cloud Platform Book
```

In addition to this overview information, you can get detailed information about a single managed zone by using the `describe` command. Listing 13-3 uses the `describe` command to get more details about the `cloudplatformbook` managed DNS zone.

**Listing 13-3.** `gcloud` DNS Command to Describe One Managed Zone

```
$ gcloud dns managed-zones describe cloudplatformbook
```

```
NAME                DNS_NAME                DESCRIPTION
creationTime: '2015-03-15T04:27:13.299Z'
description: Managed Zone for Google Cloud Platform Book
dnsName: cloudplatformbook.com.
id: '8730191890933023381'
kind: dns#managedZone
name: cloudplatformbook
nameServers:
- ns-cloud1.googledomains.com.
- ns-cloud2.googledomains.com.
- ns-cloud3.googledomains.com.
- ns-cloud4.googledomains.com.
```

Two items in the returned result merit special mention:

- `id`: Cloud DNS automatically assigns a numeric number as the identifier for this managed-zone object. This is how this managed zone is identified internally within Cloud DNS. You don't need to remember this or write it down, because you can always retrieve all the managed zones using the command `$ gcloud dns managed-zone list`.
- `nameServers`: For every managed zone, Cloud DNS automatically assigns multiple DNS servers that are located at different Google-operated data centers. You must delegate your domain to all of these name servers to enjoy the high availability of Cloud DNS and update the domain registrar accordingly. Only then are Cloud DNS name servers reachable by the client for DNS queries. You don't have any records yet, so you can postpone this step until later.

As part of DNS zone creation, Cloud DNS automatically adds two resource records: SOA and NS. Although NS is listed as part of the `describe` command, key information such as TTL values are not present. Listing 13-4 uses the `record-sets` command to list all the details of these two resource records.

**Listing 13-4.** `gcloud` DNS Command to List All Record Sets in a Managed Zone

```
$ gcloud dns record-sets list --zone=cloudplatformbook
```

```
NAME                TYPE  TTL  DATA
cloudplatformbook.com. NS    21600 ns-cloud1.googledomains.com., ns-cloud2.googledomains.com.,
                                     ns-cloud3.googledomains.com., ns-cloud4.googledomains.com.
cloudplatformbook.com. SOA    21600 ns-cloud1.googledomains.com. dns-admin.google.com. \
0 21600 3600 1209600 300
```

Let's start by adding new DNS records to the DNS zone `cloudplatformbook.com`. The first step is to build a transaction object. Cloud DNS introduces the concept of *transactions* to satisfy the requirement that a resource record-change operation (delete or add) should be an atomic operation. This means Cloud DNS serves either the old record or the new record but never returns a void result. The transaction object state is maintained using the `transaction.yaml` file in the current directory. There can be only one pending transaction at a time. Listing 13-5 starts the transaction, and Listing 13-6 lists the contents of `transaction.yaml`.

**Listing 13-5.** `gcloud` Command to Start a Transaction Object

```
$ gcloud dns record-sets transaction start --zone=cloudplatformbook
```

```
Transaction started [transaction.yaml].
```

**Listing 13-6.** Listing the Contents of the Transactions Objects File

```
$ pwd
```

```
/Users/cloudplatformbook
```

```
$ ls -la transaction.yaml
```

```
-rw-r--r--  1 cloudplatformbook staff  380 Mar 15 14:52 transaction.yaml
```

```
$ cat transaction.yaml
```

```
---
additions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
```

```

- ns-cloud1.googledomains.com. dns-admin.google.com. 1 21600 3600 1209600 300
ttl: 21600
type: SOA
deletions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
- ns-cloud1.googledomains.com. dns-admin.google.com. 0 21600 3600 1209600 300
ttl: 21600
type: SOA.

```

You have created a new compute engine instance to run a Wordpress-powered blog that will hold the errata for this book. This section skips the steps to set up this virtual machine instance, install the Wordpress software; see Chapter 4 “Google Compute Engine” and Chapter 7 “Google Cloud SQL” for instructions. Next, Listing 13-7 uses the `gcloud` command to list the details of the compute engine VM instance.

**Listing 13-7.** `gcloud` Compute Command to List VM Instances

```
$ gcloud compute instances list
```

```

NAME                ZONE          MACHINE_TYPE  INTERNAL_IP    EXTERNAL_IP    STATUS
cloudplatformbook  us-central1-f  n1-standard-2  10.240.203.206  146.148.64.211  RUNNING

```

Now, let’s add a few resource records: the public IPv4 address of the compute engine VM instance as a type A (address) record and a CNAME record for the domain’s `www` prefix. You then list the transaction object and commit the transaction; see Listing 13-8.

**Listing 13-8.** `gcloud` Command to Add a DNS Record of Type A in Cloud DNS

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='cloudplatformbook.' \
--ttl=3600 --type=A '146.148.64.211'
```

```
Record addition appended to transaction at [transaction.yaml].
```

For any web site, in addition to resolving the naked domain (that is, one without a prefix in its domain name), the DNS system should also resolve the `www` alias. In this case, `cloudplatformbook.com` is the naked domain, and `www.cloudplatformbook.com` is the fully qualified domain name (FQDN). Web servers are usually set up to serve both variations of the domain with identical content. In the DNS setup, the `www` variation is created as an alias to the naked domain. This is achieved by using the CNAME DNS record type (see Listing 13-9). Then, the “*transaction describe*” command in Listing 13-10 displays the contents of the `transaction.yaml` file.

**Listing 13-9.** `gcloud` Command to Add a DNS Record of Type CNAME in Cloud DNS

```
$ gcloud dns record-sets -zone= cloudplatformbook transaction add \
-name='www.cloudplatformbook.com.' -ttl=3600 -type=CNAME 'cloudplatformbook.com.'
```

```
Record addition appended to transaction at [transaction.yaml].
```

**Listing 13-10.** gcloud DNS Command to Describe the State of the Transaction Object, Pending Commit

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction describe
additions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - ns-cloud1.googledomains.com. dns-admin.google.com. 1 21600 3600 1209600 300
  ttl: 21600
  type: SOA
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - 146.148.64.211
  ttl: 3600
  type: A
- kind: dns#resourceRecordSet
  name: www.cloudplatformbook.com.
  rrdatas:
  - cloudplatformbook.com.
  ttl: 3600
  type: CNAME
deletions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - ns-cloud1.googledomains.com. dns-admin.google.com. 0 21600 3600 1209600 300
  ttl: 21600
  type: SOA
```

The output shows that the two records—A and CNAME—have been successfully added as part of the pending transaction. In addition, the transaction contains an SOA record in both the additions and deletions sections. On closer inspection, you can see that the difference between the two records is the serial number, which has been incremented by one. Listing 13-11 executes this transaction.

**Listing 13-11.** gcloud DNS Command to Execute a Transaction

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction execute
Executed transaction [transaction.yaml] for managed-zone [cloudplatformbook].
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/managedZones/cloudplatformbook/changes/1].
ID START_TIME STATUS
1 2015-03-15T07:49:59.935Z pending
```

Cloud DNS applies the executed transaction quickly—typically within a minute. You can see whether the transaction is successful by listing the changes to the DNS zone and by describing an individual change. Listing 13-12 and Listing 13-13 show the commands and corresponding responses.

**Listing 13-12.** gcloud DNS Command to List All Changes Committed to a DNS Zone

```
$ gcloud dns record-sets --zone=cloudplatformbook changes list
```

```
ID  START_TIME          STATUS
1   2015-03-15T07:49:59.935Z  done
0   2015-03-15T04:27:13.300Z  done
```

**Listing 13-13.** gcloud DNS Command to Describe the Changes Committed in a Transaction ID

```
$ gcloud dns record-sets --zone=cloudplatformbook changes describe 1
```

```
additions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - ns-cloud1.googledomains.com. dns-admin.google.com. 1 21600 3600 1209600 300
  ttl: 21600
  type: SOA
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - 146.148.64.211
  ttl: 3600
  type: A
- kind: dns#resourceRecordSet
  name: www.cloudplatformbook.com.
  rrdatas:
  - cloudplatformbook.com.
  ttl: 3600
  type: CNAME
deletions:
- kind: dns#resourceRecordSet
  name: cloudplatformbook.com.
  rrdatas:
  - ns-cloud1.googledomains.com. dns-admin.google.com. 0 21600 3600 1209600 300
  ttl: 21600
  type: SOA
id: '1'
kind: dns#change
startTime: '2015-03-15T07:49:59.935Z'
status: done
```

You can also list the resource records to see the current list of DNS records that is served for your DNS zone, as shown in Listing 13-14.

**Listing 13-14.** gcloud DNS Command to List the DNS Records for a DNS Zone

```
$ gcloud dns record-sets list --zone=cloudplatformbook
NAME                               TYPE TTL  DATA
cloudplatformbook.com.            A    3600  146.148.64.211
cloudplatformbook.com.            NS    21600 ns-cloud1.googledomains.com., ns-cloud2.googledomains.com.,
                                     ns-cloud3.googledomains.com., ns-cloud4.googledomains.com.
cloudplatformbook.com.            SOA   21600 ns-cloud1.googledomains.com. dns-admin.google.com.
1 21600 3600 1209600 300
www.cloudplatformbook.com.        CNAME 3600  cloudplatformbook.com.
```

Let's add some other types of DNS records required for the Wordpress-powered blog setup. You need a MySQL database as the storage back end. This example uses the Cloud SQL product from the Google Cloud Platform for this purpose. You have created a database instance and assigned an IPv4 address to it. Cloud SQL automatically assigns an IPv6 address to each Cloud SQL instance. Again, this section skips the details of setting up a Cloud SQL instance; see Chapter 7 "Google Cloud SQL" for setup details. The commands in Listing 13-15 and Listing 13-16 list the details of the Cloud SQL instance.

**Listing 13-15.** gcloud SQL Command to List Cloud SQL Instances

```
$ gcloud sql instances list
NAME                REGION    TIER  ADDRESS                STATUS
cloudplatformbook  us-central D2    173.194.242.117  RUNNABLE
```

**Listing 13-16.** gcloud SQL Command to Describe an Individual Cloud SQL Instance

```
$ gcloud sql instances describe cloudplatformbook
<snip>
ipAddresses:
- ipAddress: 173.194.242.117
ipv6Address: 2001:4860:4864:1:199b:32da:e624:20e2
<snip>
state: RUNNABLE
```

Based on these details, you can add the two IP addresses to the Cloud DNS setup. The gcloud DNS commands in Listing 13-17, Listing 13-18, and Listing 13-19 start a new transaction and add the IPv4 and IPv6 addresses.

**Listing 13-17.** gcloud DNS Command to Start a New Transaction

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction start
Transaction started [transaction.yaml].
```

**Listing 13-18.** gcloud DNS Command to Add a DNS Record of Type A to a Transaction

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='sql.cloudplatformbook.com.' --ttl=3600 --type=A '173.194.242.117'
Record addition appended to transaction at [transaction.yaml].
```

**Listing 13-19.** gcloud DNS Command to Add a DNS Record of Type AAAA to a Transaction

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='sql6.cloudplatformbook.com.' --ttl=3600 --type=AAAA
'2001:4860:4864:1:199b:32da:e624:20e2'
Record addition appended to transaction at [transaction.yaml].
```



Suppose you made an error in a previous record, and you want to fix it. For example, the A record for your domain should point to 146.148.64.212 instead of 146.148.64.211. You can use the transaction process to remove the old record and add the new one as a single atomic operation. Listing 13-20 shows the update of the A record and continues using the open transaction.

**Listing 13-20.** `gcloud` DNS Commands to Replace an Existing Record by Removing the Old Record and Adding a New One

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction remove \
--name='cloudplatformbook.com.' --ttl=3600 --type=A '146.148.64.211'
Record removal appended to transaction at [transaction.yaml].
```

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='cloudplatformbook.com.' --ttl=3600 --type=A '146.148.64.212'
Record addition appended to transaction at [transaction.yaml].
```

After a while, when you list the DNS records in the managed zone, you should see the updated A record showing the correct IP address.

Let's create a few more commonly used/required DNS records for domains. E-mail is the predominant form of communication on the Internet today. E-mails are sent and received by software system called *mail servers*. The locations of the mail servers need to be defined in the DNS setup. You do so by declaring *MX records*. Just like name servers, there is usually more than one mail server handling a domain; and unlike name servers, there is a priority among mail servers. Hence, when declaring an MX record, you need to include the priority values as part of the declaration. For the domain `cloudplatformbook.com`, Google Apps is handling e-mail. Listing 13-21 and Listing 13-22 add MX records to the DNS zone and point them to the Google Apps mail servers. This example also shows that you can add multiple values for a single DNS record.

**Listing 13-21.** `gcloud` DNS Command to Add Multiple Record Sets of the MX Record Type

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='mx.cloudplatformbook.com.' --ttl=3600 --type=MX '1 aspmx.l.google.com.' \
'5 alt1.aspmx.l.google.com.' '5 alt2.aspmx.l.google.com.' '10 alt3.aspmx.l.google.com.' \
'10 alt4.aspmx.l.google.com.'
Record addition appended to transaction at [transaction.yaml].
```

**Listing 13-22.** `gcloud` DNS Command to Commit the Transaction You Have Built So Far

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction execute
Executed transaction [transaction.yaml] for managed-zone [cloudplatformbook].
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/managedZones/cloudplatformbook/changes/6].
ID   START_TIME           STATUS
6    2015-03-15T11:12:15.621Z pending
```

Although e-mail has made it possible for people to communicate almost instantaneously, it has its own annoyances in the form of spam and phishing, which reduce the credibility of the system. In order to counter this, a new standard called the Sender Policy Framework (SPF) was created. SPF records are part of the DNS setup and list the authorized e-mail servers for a particular domain. By using SPF records, you can whitelist hosts that are permitted to send e-mails on behalf of the domain. It is a good practice to declare SPF records for a domain; usually, MX and SPF records are declared together. If you prefer to use a wizard-style system to construct a valid SPF record for a domain, see [www.spfwizard.net](http://www.spfwizard.net).

Listing 13-23 and Listing 13-24 add the SPF record for the domain `cloudplatformbook.com`. Outgoing e-mail is permitted from servers listed in MX records and from the server hosting the domain. You need to start a new transaction, because as you have committed (and closed) the previous one.

**Listing 13-23.** `gcloud` DNS Command to Start a New Transaction

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction start
Transaction started [transaction.yaml].
```

**Listing 13-24.** `gcloud` DNS Command to Add a DNS Record of Type SPF to the Open Transaction Object

```
$ $ gcloud dns record-sets --zone=cloudplatformbook transaction add
--name='cloudplatformbook.com.' --ttl=60 --type=SPF 'v=spf1 mx a'
Record addition appended to transaction at [transaction.yaml].
```

The last record type you can add to a DNS domain is a TXT record. As the name implies, it is used to publish arbitrary information about the domain. In this case, Listing 13-25 publishes a teaser about this book.

**Listing 13-25.** Cloud DNS Command to Add a TXT Record

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction add \
--name='cloudplatformbook.com.' --ttl=60 --type=TXT \
"Building Your Next Big Thing with Google Cloud Platform by Dr. S. P. T. Krishnan and
Mr. Jose L Ugia Gonzalez (Apress). ETA May 28 2015, 978-1-4842-1005-5."
Record addition appended to transaction at [transaction.yaml].
```

```
$ gcloud dns record-sets --zone=cloudplatformbook transaction execute
Executed transaction [transaction.yaml] for managed-zone [cloudplatformbook].
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/managedZones/cloudplatformbook/changes/7].
ID START_TIME STATUS
7 2015-03-15T15:41:37.441Z pending
```

After a bit, you should be able to list the DNS record you just added. The console output in Listing 13-26 shows the TXT records for the domain, extracted using `dig` tool running on a Mac OS X laptop.

**Listing 13-26.** Listing the TXT Records from a Desktop Using the `dig` Tool

```
$ dig txt cloudplatformbook.com
; <<>> DiG 9.8.3-P1 <<>> txt cloudplatformbook.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 37865
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;cloudplatformbook.com. IN TXT

;; ANSWER SECTION:
cloudplatformbook.com. 60 IN TXT "Building Your Next Big Thing with Google
Cloud Platform
by Dr. S. P. T. Krishnan and Mr. Jose L Ugia Gonzalez (Apress). ETA May 28 2015,
978-1-4842-1005-5."
```

```
;; Query time: 31 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Sun Mar 15 21:36:16 2015
;; MSG SIZE rcvd: 207
```

Let's look at an optional but useful DNS record type called a *pointer record* (PTR). A pointer record maps an IP address to a domain. This process is called *reverse DNS lookup* / *reverse DNS resolution* (rDNS), and it is the inverse of determining the IP address of a domain name (sometimes known as *forward DNS resolution*). The root of the Internet reverse DNS database is the Address and Routing Parameter Area (arpa) top-level domain. The second-level domains of IPv4 and IPv6 are `in-addr.arpa` and `ip6.arpa`, respectively. However, you determine the domain name associated with a given IP address using the Internet's Domain Name System (DNS).

You want to add a PTR record for the compute engine VM instance because it is designed to send e-mail in addition to the Google Apps system. Following are the high-level steps required to create a PTR record. Listing 13-27 through Listing 13-32 give the Cloud DNS commands:

1. Determine the target IP address.
2. Reverse the IP address, and split it into two octets.
3. Create a new managed zone following a special convention.
4. Add a new DNS record of type PTR.

**Listing 13-27.** Step 1: Determining the Target IP Address (IPv4 in This Case)

```
$ gcloud compute instances list
```

NAME	ZONE	MACHINE_TYPE	INTERNAL_IP	EXTERNAL_IP	STATUS
cloudplatformbook	us-central1-f	n1-standard-2	10.240.203.206	146.148.64.211	RUNNING

**Listing 13-28.** Step 2: Reversing the IPv4 Address and Splitting It into Two Parts

```
Reverse_External_IP = 211.64.148.146
1st, 2nd parts = 211, 64.148.146
```

**Listing 13-29.** Step 3: Creating a New Managed Zone

```
$ gcloud dns managed-zones create rdns-64-148-146 --description='Reverse DNS Zone for Google Cloud Platform Book' --dns-name='64.148.146.in-addr.arpa.'
```

```
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/managedZones/rdns-64-148-146].
```

NAME	DNS_NAME	DESCRIPTION
rdns-64-148-146	64.148.146.in-addr.arpa.	Reverse DNS Zone for Google Cloud Platform Book

**Listing 13-30.** Step 4.1: gcloud DNS Command to Start a New Transaction

```
$ gcloud dns record-sets --zone=rdns-64-148-146 transaction start
```

```
Transaction started [transaction.yaml].
```

**Listing 13-31.** Step 4.2: gcloud DNS Command to Add a DNS Record of Type PTR to a Transaction

```
$ gcloud dns record-sets --zone=rdns-64-148-146 transaction add \
--name='211.64.148.146.in-addr.arpa.' --ttl=60 --type=PTR 'cloudplatformbook.com.'
```

```
Record addition appended to transaction at [transaction.yaml].
```

**Listing 13-32.** Step 4.3: gcloud DNS Command to Commit the Transaction

```
$ gcloud dns record-sets --zone=rdns-64-148-146 transaction execute
Executed transaction [transaction.yaml] for managed-zone [rdns-64-148-146].
Created [https://www.googleapis.com/dns/v1beta1/projects/cloud-platform-book/managedZones/rdns-64-148-146/changes/1].
ID  START_TIME                STATUS
1   2015-03-16T13:06:35.210Z  pending
```

You can verify whether the DNS records have been successfully added to the two managed zones by listing the current records using the commands in Listing 13-33 and Listing 13-34.

**Listing 13-33.** Listing DNS Records Using the Cloud DNS gcloud Tool

```
$ gcloud dns record-sets --zone='cloudplatformbook' list
NAME                                TYPE  TTL  DATA
cloudplatformbook.com.             A     60   146.148.64.211
cloudplatformbook.com.             NS    21600 ns-cloud-c1.googledomains.com.,
                                     ns-cloud-c2.googledomains.com.,
                                     ns-cloud-c3.googledomains.com.,
                                     ns-cloud-c4.googledomains.com.
cloudplatformbook.com.             SOA   21600 ns-cloud-c1.googledomains.com. dns-admin.google.com.
19 21600 3600 1209600 300
cloudplatformbook.com.             SPF   60   "v=spf1" "mx" "a"
cloudplatformbook.com.             TXT   60   "Building Your Next Big Thing with Google Cloud
                                     Platform by
                                     Dr. S. P. T. Krishnan and Mr. Jose L Ugia Gonzalez (Apress). ETA May 28 2015, 978-1-4842-1005-5."
mx.cloudplatformbook.com.         MX    60   1 aspmx.l.google.com., 5 alt1.aspmx.l.google.com.,
5 alt2.aspmx.l.google.com., 10 alt3.aspmx.l.google.com., 10 alt4.aspmx.l.google.com.
sql.cloudplatformbook.com.         A     60   173.194.242.117
sql6.cloudplatformbook.com.        AAAA  60   2001:4860:4864:1:199b:32da:e624:20e2
www.cloudplatformbook.com.         CNAME 60   cloudplatformbook.com.
```

**Listing 13-34.** Listing DNS Records Using the Cloud DNS gcloud Tool

```
$ gcloud dns record-sets --zone='rdns-64-148-146' list
NAME                                TYPE  TTL  DATA
64.148.146.in-addr.arpa.          NS    21600 ns-cloud1.googledomains.com.,
                                     ns-cloud2.googledomains.com.,
                                     ns-cloud3.googledomains.com.,
                                     ns-cloud4.googledomains.com.
64.148.146.in-addr.arpa.          SOA   21600 ns-cloud1.googledomains.com. dns-admin.google.com.
1 21600 3600 1209600 300
211.64.148.146.in-addr.arpa.      PTR   60   cloudplatformbook.com.
```

The final step in the process of setting up a DNS server for your domain using Cloud DNS is to declare the Cloud DNS name servers for the domain in the domain registrar's panel. Depending on the domain registrar, the web interface will vary. Once the changes are made, you should be able to resolve the domain using standard tools.

## Summary

In this chapter, you learned about a fundamental Internet technology: DNS. You also learned about the Cloud DNS service from Google Cloud Platform that enables you to publish various domains. The chapter began by examining the building blocks of Cloud DNS, after which you deployed a real-life domain (the domain for this book). Finally, you verified that the domain you set up is accessible over the Internet by querying its record. This chapter should have provided you with sufficient information to get started using Cloud DNS.