Overview

This page provides an overview of Cloud DNS features and capabilities. To get started using Cloud DNS, see the Quickstart (/dns/docs/quickstart).

Introduction

Cloud DNS is a high-performance, resilient, global Domain Name System (DNS) service that publishes your domain names to the global DNS in a cost-effective (/dns/pricing) way.

Concepts

DNS is a hierarchical distributed database that lets you store IP addresses and other data, and look them up by name. Cloud DNS lets you publish your zones and records in the DNS without the burden of managing your own DNS servers and software.

For a list of general DNS terminology, see the General DNS overview (/dns/docs/dns-overview).

Cloud DNS offers both public zones (#dns-public-zones) and private managed DNS zones (#dns-private-zones). A public zone is visible to the public internet, while a private zone is visible only from one or more VPC networks that you specify.

Cloud DNS terminology

The Cloud DNS API is built around projects (#dns-project), managed zones (#dns-managed-zones), record sets (#dns-record-sets), and changes to record sets.

Project

> A Google Cloud Console project is a container for resources, a domain for access control, and the place where billing (/dns/pricing) is configured and aggregated. For more details, see Creating and managing projects (/resource-manager/docs/creating-managing-projects).

Managed zones

The managed zone holds DNS records for the same DNS name suffix (`example.com`, for example). A project can have multiple managed zones, but they must each have a unique name. In Cloud DNS, the managed zone is the resource that models a DNS zone (http://wikipedia.org/wiki/DNS_zone). All records in a managed zone are hosted on the same Google-operated name servers. These name servers respond to DNS queries against your managed zone according to how you configure the zone. A project can contain multiple managed zones. Charges accrue (/dns/pricing) for each zone for each day that the managed zone exists. Managed zones support labels (https://cloudplatform.googleblog.com/2015/10/using-labels-to-organize-Google-Cloud-Platform-resources.html), which you can use to help organize your billing.

Public zones

A public zone is visible to the internet. Cloud DNS has public authoritative name servers that respond to queries about public zones regardless of where the queries originate. You can create DNS records in a public zone to publish your service on the internet. For example, you might create the following record in a public zone `example.com` for your public web site `www.example.com`.

DNS Name	Type	TTL (seconds)	Data
www.example.com	A	300	198.51.100.0

Cloud DNS assigns a set of name servers when a public zone is created. For the DNS records in a public zone to be resolvable over the internet, you must update the name server setting (/dns/docs/update-name-servers) of your domain registration at your registrar.

★ **Note:** Cloud DNS public zones are authoritative, and NS and SOA resource records are at the zone apex. You cannot delete these types of records. For details, see RFC 1034 (https://tools.ietf.org/html/rfc1034).

For more information on how to register and set up your domain, see Setting up a domain using Cloud DNS (/dns/docs/tutorials/create-domain-tutorial).

Private zones

Private zones enable you to manage custom domain names for your virtual machines, load balancers, and other Google Cloud resources without exposing the underlying DNS data to the public internet. A private zone is a container of DNS records that can only be queried by one or more VPC networks that you authorize.

A private zone can only be queried by resources in the same project where it is defined. The VPC networks that you authorize must be located in the same project as the private zone. If you need to query records hosted in managed private zones in other projects, use DNS peering (#dns-peering).

You must specify the list of authorized VPC networks that can query your private zone when you create or update it. Only authorized networks are allowed to query your private zone; if you do not specify any authorized networks, you cannot query the private zone at all.

You can use private zones with Shared VPC (/vpc/docs/shared-vpc). For important information about using private zones with Shared VPC, see Shared VPC considerations (#shared-vpc).

Private zones do not support DNS security extensions (DNSSEC) or custom resource record sets of type `NS`.

Requests for DNS records in private zones must be submitted through the metadata server, `169.254.169.254`, which is the default internal name server for VMs created from Google-supplied images (/compute/docs/images#os-compute-support). You can submit queries to this name server from any VM that uses an authorized VPC network.

For example, you can create a private zone for `dev.gcp.example.com` to host internal DNS records for experimental applications. The following table shows example records in that zone. Database clients can connect to the database server, `db-01.dev.gcp.example.com`, using its internal DNS name instead of its IP address. Database clients resolve this internal DNS name using the host resolver on the VM, which submits the DNS query to the metadata server, `169.254.169.254`. The metadata server acts as a recursive resolver to query your private zone.

DNS Name	Type	TTL (seconds)	Data
db-01.dev.gcp.example.com	A	5	10.128.1.35
instance-01.dev.gcp.example.com	A	50	10.128.1.10

Private zones enable you to create split horizon (#split-horizon) DNS configurations. This is because you can create a private zone with a different set of records, which override those in a public zone. You can then control which VPC networks query the records in the private zone. For example, see overlapping zones (#overlapping).

Service Directory

Beta

This product or feature is covered by the Pre-GA Offerings Terms (/terms/service-terms#1) **of the Google Cloud Platform Terms of Service. Pre-GA products and features may have limited support, and changes to pre-GA products and features may not be compatible with other pre-GA versions. For more information, see the launch stage descriptions** (/products#product-launch-stages)**.**

Service Directory is a managed service registry for Google Cloud. It enables you to register and discover services using HTTP or gRPC (using its Lookup API) in addition to using traditional DNS. You can register both Google Cloud and non-Google Cloud services using Service Directory.

Cloud DNS enables you to create Service Directory-backed zones, which are a type of private zone containing information about your services and endpoints. You don't add record sets to the zone; rather, they're inferred automatically based on the configuration of the Service Directory namespace associated with the zone. For detailed information about Service Directory, see the Service Directory overview (/service-directory/docs/overview).

Cloud DNS and non-RFC 1918 addresses

By default, Cloud DNS forwards to non-RFC 1918 addresses through the public internet. However, Cloud DNS also supports forwarding to non-RFC 1918 addresses for private zones.

Once you configure a VPC network to use non-RFC 1918 addresses, you must configure the Cloud DNS private zone as a managed reverse lookup zone (/dns/zones#create-mrl-zone). This configuration enables Cloud DNS to resolve non-RFC 1918 addresses locally instead of sending them over the internet.

Cloud DNS also supports outbound forwarding to non-RFC 1918 addresses by privately routing those addresses within Google Cloud. To enable this type of outbound forwarding, you must configure a forwarding zone with specific forwarding path arguments. For details, see Creating an outbound server policy (/dns/docs/policies#create-out).

Forwarding zones

A forwarding zone is a type of Cloud DNS managed private zone that sends requests for that zone to the IP addresses of its forwarding targets. For more information, see DNS forwarding methods (#dns-forwarding-methods).

Peering zones

A peering zone is a type of Cloud DNS managed private zone that follows the name resolution order (#vpc-name-resolution-order) of another VPC network and can be used to resolve the names that are defined in the other VPC network.

Zone operations

Any changes that you make to managed zones in Cloud DNS are recorded in the operations collection (/dns/docs/operations), which lists managed zone updates (modifying descriptions or DNSSEC state or configuration).

Internationalized Domain Names (IDN)

An internationalized domain name (IDN) (https://www.icann.org/resources/pages/idn-2012-02-25-en) is an internet domain name that allows people all over the world to use a language-specific script or alphabet, such as Arabic, Chinese, Cyrillic, Devanagari, Hebrew, or the Latin alphabet-based special characters in domain names. This conversion is implemented using Punycode (https://wikipedia.org/wiki/Punycode), which is a representation of

Unicode characters using ASCII. For example, an IDN representation of `.ελ` is `.xn--qxam`. Some browsers, email clients, and applications might recognize it and render it as `.ελ` on your behalf. Note that the Internationalizing Domain Names in Applications (IDNA) (https://tools.ietf.org/html/rfc3490) standard only allows for Unicode strings that are short enough to be represented as a valid DNS label. For information on how you can use IDN with Cloud DNS, see Creating zones with internationalized domain names (/dns/zones/international-domains).

Registrar

A domain name registrar (https://wikipedia.org/wiki/Domain_name_registrar) is an organization that manages the reservation of internet domain names. A registrar must be accredited by a generic top-level domain (gTLD) registry or a country code top-level domain (ccTLD) registry.

Internal DNS

Google Cloud creates internal DNS names for VMs automatically, even if you do not use Cloud DNS. For more information about internal DNS, see the internal DNS documentation (/compute/docs/internal-dns).

Delegated subzones

DNS allows the owner of a zone to delegate a subdomain to a different name server using NS records. Resolvers follow these records and send queries for the subdomain to the target name server specified in the delegation.

★ **Note:** Private zones do not support delegation of subdomains to different name servers using NS records.

Resource record sets collection

The resource record sets collection holds the current state of the DNS records that make up a managed zone. You can read this collection but you do not modify it directly. Rather, you edit the resource record sets in a managed zone by creating a `Change` request in the changes collection. The resource record sets collection reflects all your changes immediately. However, there is a delay

between when changes are made in the API and the time that they take effect at your authoritative DNS servers. Managing Records (/dns/records) explains how to manage records.

★ **Note:** In consistency with DNS industry practices, Cloud DNS name servers now randomize the order of the resource record sets. This is normal DNS behavior and applies to both public *and* private Cloud DNS zones.

Resource record changes

To make a change to the resource record sets collection, submit a `Change` request containing additions or deletions. Additions and deletions can be done in bulk or in a single atomic transaction, and take effect at the same time in each authoritative DNS server.

For example, if you have an A record that looks like this:

```
www  A  203.0.113.1 203.0.113.2
```

And you run a command that looks like this:

```
DEL  www  A  203.0.113.2
ADD  www  A  203.0.113.3
```

Your record looks like this after the bulk change:

```
www  A  203.0.113.1 203.0.113.3
```

The ADD and DEL happen simultaneously.

SOA serial number format

The serial numbers of SOA records created in Cloud DNS managed zones monotonically increase with each transactional change to a zone's record sets made using the `gcloud dns record-sets`

`transaction` command. You are free to manually change the serial number of an SOA record to an arbitrary number, however, including an ISO 8601-formatted date as recommended in RFC 1912. For example, in the following SOA record:

```
ns-gcp-private.googledomains.com. cloud-dns-hostmaster.google.com. [serial number] 21600 3600 259200 300
```

You can change the serial number directly from the Google Cloud Console by entering the desired value into the third space-delimited field of the record.

DNS Server Policy

A DNS server policy lets you access name resolution services provided by Google Cloud in a VPC network with inbound forwarding, or change the VPC name resolution order (#vpc-name-resolution-order) with outbound forwarding.

Domains, Subdomains, and Delegation

Most subdomains are just records in the managed zone for the parent domain. Subdomains that are **delegated** by creating NS (name server) records in their parent domain's zone need to have their own zones as well. *Create managed zones for parent domains in Cloud DNS* **before** creating any zones for their delegated subdomains. Do this even if you are hosting the parent domain on another DNS service. If you have several subdomain zones but don't create the parent zone, it can be complicated (/dns/quotas#nameserver_limits) to create the parent zone later if you decide to move it to Cloud DNS.

DNSSEC

The Domain Name System Security Extensions (DNSSEC) (/dns/docs/dnssec) is a suite of Internet Engineering Task Force (IETF) extensions to DNS which authenticate responses to domain name lookups. DNSSEC does not provide privacy protections for those lookups, but prevents attackers from manipulating or poisoning the responses to DNS requests.

DNSKEYs collection

The DNSKEYs collection holds the current state of the DNSKEY records used to sign a DNSSEC-enabled managed zone. You can only read this collection; all changes to the DNSKEYs are made by Cloud DNS. The DNSKEYs collection has all the information that domain registrars require to activate DNSSEC (/dns/docs/registrars#add-ds).

Shared VPC considerations

To use a Cloud DNS managed private zone, Cloud DNS forwarding zone, or Cloud DNS peering zone with Shared VPC, you must create the zone in the host project (/vpc/docs/shared-vpc#concepts_and_terminology), then add the appropriate Shared VPC network(s) to the list of authorized networks for that zone.

For more information, see Best practices for Cloud DNS private zones (/dns/docs/best-practices-dns#best_practices_for_private_zones).

VPC name resolution order

Each VPC network provides DNS name resolution services to the virtual machine (VM) instances that use it. When VMs use their metadata server, `169.254.169.254`, as their name server, Google Cloud searches for DNS records according to the following order:

- If your VPC network has an outbound server policy (#dns-server-policy-out), Google Cloud forwards **all** DNS queries to those alternative servers (#altns-targets). The VPC name resolution order consists only of this step.
- If your VPC network does *not* have an outbound server policy:
    1. Google Cloud tries to find a private zone that matches as much of the requested record as possible (longest suffix matching). This includes:
        - Searching records you created in private zones
        - Querying the forwarding targets for forwarding zones
        - Querying the name resolution order of another VPC network using peering zones
    2. Google Cloud searches the automatically created Compute Engine internal DNS (/compute/docs/internal-dns) records for the project.

3. Google Cloud queries publicly available zones, following the appropriately configured start-of-authority (SOA). This includes Cloud DNS public zones.

Cloud DNS does not fall back to the next form of name resolution if a given query fails. For example, if a query f
`le.com.` matches a private zone and results in **`NXDOMAIN`**, Cloud DNS does not try to query the public Internet
`le.com.`.

DNS forwarding methods

Google Cloud offers inbound and outbound DNS forwarding for private zones.

- *Inbound forwarding* means enabling an on-premises DNS client or server to send DNS requests to Cloud DNS. The DNS client or server can then resolve records according to a VPC network's name resolution order (#vpc-name-resolution-order). Inbound forwarding lets on-premises clients resolve records in private zones, forwarding zones, and peering zones for which the VPC network has been authorized. On-premises clients connect to the VPC network using Cloud VPN or Cloud Interconnect.
- *Outbound forwarding* means enabling VMs in Google Cloud to send DNS requests to DNS name servers of your choosing. The name servers can be located in the same VPC network, an on-premises network, or on the internet.

You configure DNS forwarding by creating a forwarding zone (#dns-forwarding-zones) or by creating a Cloud DNS server policy (#dns-server-policy). The two methods are summarized in the following table:

Forwarding	Cloud DNS methods
Inbound	On-premises systems can send requests to a VPC network in order to use that network's VPC name resolution order (#vpc-name-resolution-order) if you create an inbound server policy (#dns-server-policy-in) for that VPC network.
Outbound	You can configure VMs in a VPC network to: • Resolve records hosted on name servers configured as forwarding targets of a forwarding zone (#dns-forwarding-zones) authorized for use by your VPC network. For important information about how Google Cloud routes traffic to a forwarding target's IP address, see Forwarding target and routing methods (#fz-targets). • Send all DNS requests an alternative name server by creating an outbound server policy (#dns-server-policy-out) for the VPC network. When using an alternative name server, VMs in

your VPC network are no longer able to resolve records in Cloud DNS private zones, forwarding zones, or peering zones. Carefully review the VPC name resolution order (#vpc-name-resolution-order) for additional details.

You can simultaneously configure inbound and outbound forwarding for a VPC network. Bi-directional forwarding lets VMs in your VPC network resolve records in an on-premises network or in a network hosted by a different cloud provider. This type of forwarding also enables hosts in the on-premises network to resolve records for your Google Cloud resources.

Cloud DNS does *not* support DNS forwarding for public zones. Cloud DNS public zones must be authoritative.

The Cloud DNS control plane uses an algorithm to determine the responsiveness of a forwarding target. Your outbound forwarded queries may sometimes result in `SERVFAIL` errors. For information on how to work around this issue, see the related section in the Troubleshooting documentation (/dns/docs/troubleshooting#outbound-forwarded-queries-receive-servfail-errors).

PTR records in private zones

PTR records for RFC 1918 addresses

To perform reverse lookups with custom PTR records for RFC 1918 addresses (https://wikipedia.org/wiki/Private_network#Private_IPv4_addresses), you must create a Cloud DNS private zone that is at least as specific as one of the following example zones. This is a consequence of the longest-suffix matching described in VPC name resolution order (#vpc-name-resolution-order).

- `10.in-addr.arpa.`
- `168.192.in-addr.arpa.`
- `16.172.in-addr.arpa.`, `17.172.in-addr.arpa.`, ... through `31.172.in-addr.arpa.`

If you create a Cloud DNS private zone for RFC 1918 addresses, custom PTR records that you create for VMs in that zone are overridden by the PTR records that Compute Engine internal DNS (/compute/docs/internal-dns) creates automatically. This is because Compute Engine internal DNS PTR records are created in the previous list of more specific zones.

For example, suppose you create a managed private zone for `in-addr.arpa.` with the following PTR record for a VM whose IP address is `10.1.1.1`:

1.1.1.10.in-addr.arpa. -> test.example.domain

PTR queries for `1.1.1.10.in-addr.arpa.` are answered by the more specific `10.in-addr.arpa.` Compute Engine internal DNS zone. The PTR record in your Cloud DNS private zone for `test.example.domain` is ignored.

To override the automatically created Compute Engine internal DNS PTR records for VMs, make sure that you create your custom PTR records in a private zone that is at least as specific as one of the zones presented in this section. For example, if you create the following PTR record in a private zone for `10.in-addr.arpa.`, your record overrides the automatically generated one:

1.1.1.10.in-addr.arpa. -> test.example.domain

You can also create more specific reverse Cloud DNS private zones if you only need to override a portion of a network block. For example, a private zone for `5.10.in-addr.arpa.` can be used hold PTR records that override any Compute Engine internal DNS PTR records that are automatically created for VMs with IP addresses in the `10.5.0.0/16` address range.

Supported DNS record types

Cloud DNS supports the following types of records:

Record type	Description
`A`	Address record, which maps host names to their IPv4 address.
`AAAA`	IPv6 Address record, which maps host names to their IPv6 address.
`CAA`	Certificate Authority (CA) Authorization, which specifies which CAs are allowed to create certificates for a domain.
`CNAME`	Canonical name record, which specifies alias names. Cloud DNS does not support resolving CNAMEs recursively across different managed private zones (CNAME chasing). For details, see Troubleshooting (/dns/docs/troubleshooting#cname-in-private-zone-not-working).

Record type	Description
IPSECKEY	IPSEC tunnel gateway data and public keys for IPSEC-capable clients to enable opportunistic encryption (/dns/docs/dnssec#ipseckey).
MX	Mail exchange record, which routes requests to mail servers.
NAPTR	Naming authority pointer record, defined by RFC 3403 (http://tools.ietf.org/html/rfc3403).
NS	Name server record, which delegates a DNS zone to an authoritative server.
PTR	Pointer record, which is often used for reverse DNS lookups.
SOA	Start of authority record, which specifies authoritative information about a DNS zone. An SOA resource record is created for you when you create your managed zone. You can modify the record as needed (for example, you can change the serial number to an arbitrary number to support date-based versioning).
SPF	Sender Policy Framework record, a deprecated record type formerly used in e-mail validation systems (use a TXT record instead).
SRV	Service locator record, which is used by some voice over IP, instant messaging protocols, and other applications.
SSHFP	SSH fingerprint for SSH clients to validate the public keys of SSH servers (/dns/docs/dnssec#sshfp).
TLSA	TLS authentication record for TLS clients to validate X.509 server certificates (/dns/docs/dnssec-advanced#tlsa).
TXT	Text record, which can contain arbitrary text and can also be used to define machine-readable data, such as security or abuse prevention information. A TXT record may contain one or more text strings; the maximum length of each individual string is 255 characters (https://tools.ietf.org/html/rfc4408#section-3.1.3). Mail agents and other software agents concatenate multiple strings. Enclose each string in quotation marks. For example: `"Hello world" "Bye world"`

Managing Records (/dns/records) shows how to work with DNS records.

Wildcard DNS records

Wildcard records are supported for all record types, except for NS records.

DNSSEC

Cloud DNS supports managed DNSSEC, protecting your domains from spoofing and cache poisoning attacks. When you use a validating resolver like Google Public DNS (https://developers.google.com/speed/public-dns/), DNSSEC provides strong authentication (but not encryption) of domain lookups. For more information on DNSSEC, see Managing DNSSEC configuration (/dns/docs/dnssec-config).

Forwarding zones

Cloud DNS forwarding zones allow you to configure target name servers for specific private zones. Using a forwarding zone is one way to implement outbound DNS forwarding from your VPC network.

A Cloud DNS forwarding zone is a special type of Cloud DNS private zone. Instead of creating records within the zone, you specify a set of *forwarding targets*. Each forwarding target is an IP addresses of a DNS server, located in your VPC network, or in an on-premises network connected to your VPC network by Cloud VPN or Cloud Interconnect.

Forwarding targets and routing methods

Cloud DNS supports three types of targets and offers standard or private methods for routing traffic to them.

Forwarding target	Description	Standard routing	Private routing	Requests come from
Type 1	An internal IP address of a Google Cloud VM in **the same** VPC network that is authorized to use the forwarding zone	Must be RFC 1918 IP addresses — traffic always routed through an authorized VPC network	Any internal IP address — traffic always routed through an authorized VPC network	`35.199.192.0/19`
Type 2	An IP address of	Must be RFC	Any internal IP	`35.199.192.0/19`

	an on-premises system, connected to the VPC network authorized to query the forwarding zone, using Cloud VPN or Cloud Interconnect	1918 IP addresses — traffic always routed through an authorized VPC network	address — traffic always routed through an authorized VPC network	
Type 3	An external IP address of a DNS name server on the internet. Includes an external IP address of a Google Cloud resource.	Must be an internet routable address — traffic always routed to the internet	Private routing isn't supported.	Google Public DNS source ranges (https://developers.google.com/speed/public-dns/faq#locations)

You can choose one of the two following routing methods when you add the forwarding target to the forwarding zone:

- **Standard routing**: Routes traffic through an authorized VPC network or over the internet based on whether or not the forwarding target is an RFC 1918 IP address. If the forwarding target is an RFC 1918 IP address, Cloud DNS classifies the target as either a *Type 1* or *Type 2* target, and routes requests through an authorized VPC network. If the target is not an RFC 1918 IP address, Cloud DNS classifies the target as *Type 3*, and expects the target to be internet accessible.
- **Private routing**: Always routes traffic through an authorized VPC network, regardless of the target's IP address (RFC 1918 or not). Consequently, only *Type 1* and *Type 2* targets are supported.

To access a *Type 1* or a *Type 2* forwarding target, Cloud DNS uses routes in the authorized VPC network, where the DNS client is located. These routes define a secure path to the forwarding target:

- Cloud DNS uses automatically created subnet routes (/vpc/docs/routes#subnet-routes) to send traffic to *Type 1* targets. *Type 1* targets reply using a special return route for Cloud DNS responses (/vpc/docs/routes#cloud-dns).

- Cloud DNS can use either custom dynamic routes or custom static routes (/vpc/docs/routes#custom-routes), except for custom static routes with network tags, to send traffic to *Type 2* targets. *Type 2* forwarding targets reply using routes in your on-premises network.

For additional guidance about network requirements for *Type 1* and *Type 2* targets, see forwarding target network requirements (/dns/zones#private_targets).

tant: A *Type 1* forwarding target **cannot** be a VM located in another Google Cloud VPC network that's connect network authorized to query the forwarding zone. It doesn't matter *how* the two VPC networks are connected using VPC Network Peering or Cloud VPN. If you have a forwarding target in a second VPC network, create a rding zone and authorize that zone for the second network, then create a peering zone (#peering-zones) in the etwork.

Using forwarding zones

VMs in a VPC network can use a Cloud DNS forwarding zone in the following cases:

- The VPC network has been authorized to use the Cloud DNS forwarding zone. You can authorize multiple VPC networks in the same project to use the forwarding zone.
- The guest OS of each VM in the VPC network uses the VM's metadata server, `169.254.169.254` as its name server.

Overlapping forwarding zones

Because Cloud DNS forwarding zones are a type of Cloud DNS managed private zone, you can create multiple zones that overlap. VMs configured as described above resolve a record according to the VPC name resolution order (#vpc-name-resolution-order), using the zone with the longest suffix. For more information, see Overlapping zones (#overlapping).

Caching and forwarding zones

Cloud DNS caches responses for queries sent to Cloud DNS forwarding zones. Cloud DNS maintains a cache of responses from reachable forwarding targets for the *shorter* of the following time spans:

- 60 seconds
- The duration of the record's time-to-live (TTL)

When *all* of the forwarding targets for a forwarding zone become unreachable, Cloud DNS maintains its cache of the previously-requested records in that zone for the duration of each record's TTL.

When to use peering instead

Cloud DNS cannot connect to a forwarding target by means of transitive routing. To illustrate an invalid configuration, consider the following scenario:

- You've connected an on-premises network to a VPC network named `vpc-net-a` using Cloud VPN or Cloud Interconnect.
- You've connected VPC network `vpc-net-a` to `vpc-net-b` using VPC network peering. You've configured `vpc-net-a` to export custom routes, and `vpc-net-b` to import them.
- You've created a forwarding zone whose forwarding targets are located in the on-premises network to which `vpc-net-a` is connected. You've authorized `vpc-net-b` to use that forwarding zone.

Resolving a record in a zone served by the forwarding targets fails in this scenario, even though there is connectivity from `vpc-net-b` to your on-premises network. To demonstrate this failure, perform the following tests from a VM in `vpc-net-b`:

- Query the VM's metadata server, `169.254.169.254`, for a record defined in the forwarding zone. This query fails (expectedly) because Cloud DNS does not support transitive routing to forwarding targets. A VM must be configured to use its metadata server (#fz-using) in order to use a forwarding zone.
- Query the forwarding target directly for that same record. This query demonstrates that transitive connectivity succeeds, though Cloud DNS does not use this path.

You can fix this invalid scenario by using a Cloud DNS peering zone (#dns-peering):

1. Create a Cloud DNS peering zone authorized for `vpc-net-b` that targets `vpc-net-a`.
2. Create a forwarding zone authorized for `vpc-net-a` whose forwarding targets are on-premises name servers.

You can perform these steps in any order. After completing these steps, Compute Engine instances in both `vpc-net-a` and `vpc-net-b` can query the on-premises forwarding targets.

You can leave VPC networks **`vpc-net-a`** and **`vpc-net-b`** connected using VPC network peering; however, VP rk peering is not required for the Cloud DNS peering zone to operate. Peering zones do not depend on VPC net g.

DNS peering

DNS peering lets you send requests for records that come from one zone's namespace to another VPC network. For example, a SaaS provider can give a SaaS customer access to DNS records it manages.

To provide DNS peering, you must create a Cloud DNS peering zone and configure it to perform DNS lookups in a VPC network where the records for that zone's namespace are available. The VPC network where the DNS peering zone performs lookups is called the *DNS producer network*.

To use DNS peering, you must authorize a network to use a peering zone. The VPC network authorized to use the peering zone is called the *DNS consumer network*.

Once authorized, Google Cloud resources in the DNS consumer network can perform lookups for records in the peering zone's namespace as if they were in the DNS producer network. Lookups for records in the peering zone's namespace follow the DNS producer network's name resolution order (#vpc-name-resolution-order). Thus, Google Cloud resources in the DNS consumer network can look up records in the zone's namespace from the following sources available in the DNS producer network:

- Cloud DNS managed private zones authorized for use by the DNS producer network
- Cloud DNS managed forwarding zones authorized for use by the DNS producer network
- Compute Engine internal DNS names in the DNS producer network
- An alternative name server, if an outbound DNS policy has been configured in the DNS producer network

DNS peering limitations

Keep the following in mind when configuring DNS peering:

- DNS peering is a one-way relationship. It allows Google Cloud resources in the DNS consumer network to look up records in the peering zone's namespace as if the Google Cloud resources were in the DNS producer network.
- The DNS producer and consumer networks must be VPC networks.
- DNS peering and VPC network peering (/vpc/docs/vpc-peering) are different services. DNS peering can be used in conjunction with VPC network peering, but VPC network peering is *not* required for DNS peering.
- Transitive DNS peering is supported, but only through a single transitive hop. In other words, no more than three VPC networks (with the network in the middle being the transitive hop) can be involved. For example, you can create a peering zone in `vpc-net-a` which targets `vpc-net-b`, then a peering zone in `vpc-net-b`, which targets `vpc-net-c`.
- If you are using DNS peering to target a forwarding zone, the target VPC network with the forwarding zone must contain a VM, an interconnect attachment (VLAN), or a Cloud VPN tunnel located in the same region as the source VM that uses the DNS peering zone. For details on this limitation, see the Troubleshooting section (/dns/docs/troubleshooting#forwarding-queries-from-consumer-vpc-to-producer-vpc-not-working).

To create a peering zone, you must have the DNS Peer (/iam/docs/understanding-roles#dns-roles) IAM role for the project that contains the DNS producer network.

Overlapping zones

Two zones *overlap* with each other when the origin domain name of one zone is either identical to or is a subdomain of the origin of the other zone. As examples:

- A zone for `gcp.example.com` and another zone for `gcp.example.com` overlap because the domain names are identical.
- A zone for `dev.gcp.example.com` and a zone for `gcp.example.com` overlap because `dev.gcp.example.com` is a subdomain of `gcp.example.com`. See the next section (#overlapping-rules) for rules that apply to overlapping zones.

Rules for overlapping zones

Cloud DNS enforces the following rules for overlapping zones:

- Overlapping public zones are not allowed on the same Cloud DNS name servers. When you create overlapping zones, Cloud DNS attempts to put them on different name servers. If that is not possible, Cloud DNS fails to create the overlapping zone.
- A private zone can overlap with any public zone.
- Private zones scoped for different VPC networks can overlap with each other. For example, two VPC networks can each have a database VM named `database.gcp.example.com` in a zone `gcp.example.com`. Queries for `database.gcp.example.com` receive different answers according to the zone records defined in the zone authorized for each VPC network.
- Two private zones that have been authorized to be accessible from the same VPC network cannot have identical origins unless one zone is a subdomain of the other. The metadata server uses longest-suffix matching in determining which origin to query for records in a given zone.

Query resolution examples

Google Cloud resolves Cloud DNS zones as described in VPC name resolution order (#vpc-name-resolution-order). When determining the zone to query for a given record, Cloud DNS tries to find a zone that matches as much of the requested record as possible (longest suffix match).

Unless you have specified an alternative name server in an outbound server policy, Google Cloud first attempts to find a record in private zone (or forwarding zone or peering zone) authorized for your VPC network *before* it looks for the record in a public zone.

The following examples illustrate the order that the metadata server uses when querying DNS records. For each of these examples, suppose that you have created two private zones, `gcp.example.com` and `dev.gcp.example.com`, and authorized access to them from the same VPC network. Google Cloud handles the DNS queries from VMs in a VPC network in the following way:

- The metadata server uses public name servers to resolve a record for `myapp.example.com` because there is no private zone for `example.com` that has been authorized for the VPC network.

- The metadata server resolves the record `myapp.gcp.example.com` using the authorized private zone `gcp.example.com` because `gcp.example.com` is the longest common suffix between the requested record name and available authorized private zones. `NXDOMAIN` is returned if there's no record for `myapp.gcp.example.com` defined in the `gcp.example.com` private zone, *even if there is a record for* `myapp.gcp.example.com` *defined in a public zone*.
- Similarly, queries for `myapp.dev.gcp.example.com` are resolved according to records in the authorized private zone `dev.gcp.example.com`. `NXDOMAIN` is returned if there is no record for `myapp.dev.gcp.example.com` in the `dev.gcp.example.com` zone, *even if there is a record for* `myapp.dev.gcp.example.com` *in another private or public zone*.
- Queries for `myapp.prod.gcp.example.com` are resolved according to records in the private zone `gcp.example.com`, because `gcp.example.com` is the longest common suffix between the requested DNS record and the available private zones.

Split horizon DNS example

You can use a combination of public and private zones in a split horizon DNS configuration. Private zones enable you to define different responses to a query for the same record when the query originates from a VM within an authorized VPC network. Split horizon DNS is useful whenever you need to provide different records for the same DNS queries depending on the originating VPC network.

Consider the following split horizon example:

- You've created a public zone, `gcp.example.com`, and you've configured its registrar to use Cloud DNS name servers.
- You've created a private zone, `gcp.example.com`, and you've authorized your VPC network to access this zone.

In the private zone, you've created a single record:

Record	Type	TTL (seconds)	Data
foo.gcp.example.com	A	5	10.128.1.35

In the public zone, you've created two records:

Record	Type	TTL (seconds)	Data
foo.gcp.example.com	A	5	104.198.6.142
bar.gcp.example.com	A	50	104.198.7.145

The following queries are resolved as described:

- A query for `foo.gcp.example.com` from a VM in your VPC network returns `10.128.1.35`.
- A query for `foo.gcp.example.com` from the internet returns `104.198.6.142`.
- A query for `bar.gcp.example.com` from a VM in your VPC network returns an `NXDOMAIN` error because there's no record for `bar.gcp.example.com` in the private zone `gcp.example.com`.
- A query for `bar.gcp.example.com` from the internet returns `104.198.7.145`.

DNS server policies

You can configure one DNS server policy for each VPC network. The policy can specify inbound forwarding, outbound forwarding, or both. In this section, *inbound server policy* refers to a policy that permits inbound DNS forwarding, and *outbound server policy* refers to *one possible method* for implementing outbound DNS forwarding. It is possible for a policy to be both an *inbound server policy* and an *outbound server policy* if it implements the features of both.

DNS server policies are not available for legacy networks (/vpc/docs/legacy). They require VPC networks.

For more information, see Applying server policies (/dns/docs/policies).

Inbound server policy

Each VPC network provides DNS name resolution services to the VMs that use it. When a VM uses its metadata server, `169.254.169.254`, as its name server, Google Cloud searches for DNS records according to the VPC name resolution order (#vpc-name-resolution-order).

By default, a VPC network's name resolution services — through its name resolution order — are only available to that VPC network itself. You can make these name resolution services

available to an on-premises network connected using Cloud VPN or Cloud Interconnect by creating an inbound DNS policy in your VPC network.

When you create an inbound policy, Cloud DNS takes an internal IP address from the primary IP address range of a subnet in each region your VPC network uses. It uses these internal IP addresses as entry points for inbound DNS requests.

Inbound policy entry points

The regional internal IP addresses used by Cloud DNS for the inbound DNS policy serve as entry points into the name resolution services of the VPC network. To use the inbound DNS policy, you must configure your on-premises systems or name servers to forward DNS queries to the proxy IP address located *in the same region* as the Cloud VPN tunnel or Cloud Interconnect attachment (VLAN) that connects your on-premises network to your VPC network.

For information on how to create inbound server policies, see Creating an inbound server policy (/dns/docs/policies#create-in).

Outbound server policy

You can change the VPC name resolution order (#vpc-name-resolution-order) by creating an outbound DNS policy that specifies a list of alternative name servers. When you specify alternative name servers for a VPC network, those servers are the **only** name servers that Google Cloud queries when handling DNS requests from VMs in your VPC network that are configured to use their metadata servers (`169.254.169.254`).

ant: A DNS policy that enables outbound forwarding **disables** resolution of Compute Engine internal DNS and DNS managed private zones. An outbound policy is one of two methods for outbound forwarding -forwarding-methods).

For information on how to create outbound server policies, see Creating an outbound server policy (/dns/docs/policies#create-out).

Alternative name servers and routing methods

Cloud DNS supports four types of alternative name servers and offers standard and private routing methods for routing traffic to them.

Alternative name servers are defined in the following table:

Alternative name server	Description	Standard routing	Private routing	Requests come from
Type 1	An internal IP address of a Google Cloud VM in **the same** VPC network where the outbound server policy is defined	Must be RFC 1918 IP addresses — traffic always routed through an authorized VPC network	Any internal IP address — traffic always routed through an authorized VPC network	`35.199.192.0/19`
Type 2	An IP address of an on-premises system, connected to the VPC network with the outbound server policy, using Cloud VPN or Cloud Interconnect	Must be RFC 1918 IP addresses — traffic always routed through an authorized VPC network	Any internal IP address — traffic always routed through an authorized VPC network	`35.199.192.0/19`
Type 3	An external IP address of a DNS name server on the internet. Includes an external IP address of a Google Cloud resource.	Must be an internet routable address — traffic always routed to the internet	Private routing isn't supported.	Google Public DNS source ranges (https://developers.google.com/speed/public-dns/faq#locations)
Type 4	An external IP address of a Compute Engine VM in a different VPC network.	Must be a non-RFC 1918 IP address.	Private routing isn't supported — make sure private routing is unchecked.	Google Public DNS source ranges (https://developers.google.com/speed/public-dns/faq#locations)

You can choose one of the following two routing methods when you specify the alternative name server of an outbound forwarding policy.

- **Standard routing**: Routes traffic through an authorized VPC network or over the internet based on whether or not the alternative name server is an RFC 1918 IP address. If the alternative name server is an RFC 1918 IP address, Cloud DNS classifies the name server as either a *Type 1* or *Type 2* name server, and routes requests through an authorized VPC network. If the alternative name server is not an RFC 1918 IP address, Cloud DNS classifies the name server as *Type 3*, and expects the name server to be internet accessible.
- **Private routing**: Always routes traffic through an authorized VPC network, regardless of the alternative name server's IP address (RFC 1918 or not). Consequently, only *Type 1* and *Type 2* name servers are supported.

To access a *Type 1* or a *Type 2* alternative name server, Cloud DNS uses routes in the authorized VPC network, where the DNS client is located. These routes define a secure path to the name server:

- Cloud DNS uses automatically created subnet routes (/vpc/docs/routes#subnet-routes) to send traffic to *Type 1* alternative name servers. *Type 1* name servers reply using a special return route for Cloud DNS responses (/vpc/docs/routes#cloud-dns).
- Cloud DNS can use either custom dynamic routes or custom static routes (/vpc/docs/routes#custom-routes), except for custom static routes with network tags, to send traffic to *Type 2* alternative name servers. *Type 2* name servers reply using routes in your on-premises network.

For additional guidance about network requirements for *Type 1* and *Type 2* name servers, see alternative name server network requirements (/dns/docs/policies#private_alternative_name_servers).

ant: A *Type 1* alternative name server **cannot** be a VM located in another Google Cloud VPC network that's cted to a VPC network to which the DNS outbound server policy applies. It doesn't matter *how* the two VPC rks are connected — either using VPC Network Peering or Cloud VPN.

Access control

General access control

You can manage the users who are allowed to make changes to your DNS records on the IAM & Admin page in the Google Cloud Console (https://console.cloud.google.com/iam-admin). For users to be authorized to make changes, they must be listed as either an `editor` or `owner` in the Permissions section of the Cloud Console. The viewer permission level grants read-only access to the Cloud DNS records.

These permissions also apply to service accounts that you might use to manage your DNS services.

Access control for managed zones

Users with the Project Owner or Project editor role (/iam/docs/understanding-roles#primitive_roles) can manage or view the managed zones in the specific project that they are managing.

Users with the DNS Administrator or DNS Reader role (/dns/docs/access-control#roles) can manage or view the managed zones across all the projects that they have access to.

Project owners, editors, DNS admins, and readers can view the list of private zones applied to any VPC network in the current project.

Performance and timing

Cloud DNS uses anycast (http://wikipedia.org/wiki/Anycast) to serve your managed zones from multiple locations around the world for high availability. Requests are automatically routed to the nearest location, reducing latency and improving authoritative name lookup performance for your users.

Propagation of changes

Changes are propagated in two parts. First, the change that you send through the API or command-line tool must be pushed to Cloud DNS's authoritative DNS servers. Second, DNS resolvers must pick up this change when their cache of the records expires.

The DNS resolver's cache is controlled by the time-to-live (TTL) value that you set for your records, which is specified in seconds. For example, if you set a TTL value of 86400 (the number of seconds in 24 hours), the DNS resolvers are instructed to cache the records for 24

hours. Some DNS resolvers ignore the TTL value or use their own values that can delay the full propagation of records.

If you are planning for a change to services that requires a narrow window, you might want to change the TTL to a shorter value prior to making your change. This approach can help reduce the caching window and ensure a quicker change to your new record settings. After the change, you can change the value back to its previous TTL value to reduce load on the DNS resolvers.

Next steps

- To get started using Cloud DNS, see the Quickstart (/dns/docs/quickstart)
- For a step-by-step walk-through of how to register and set up your domain, see Setting up a domain using Cloud DNS (/dns/docs/tutorials/create-domain-tutorial)
- Learn about our API client libraries (/dns/docs/libraries)

Last updated 2020-08-21 UTC.