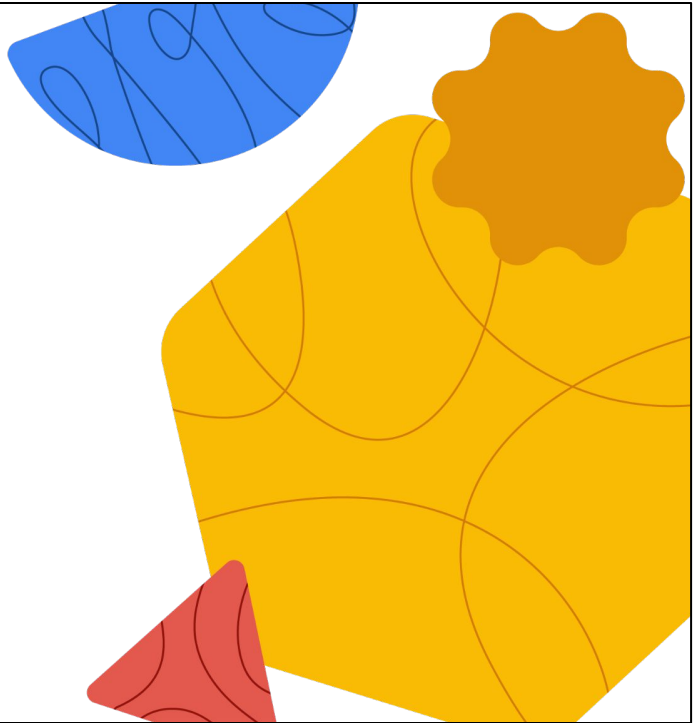





Networking in Google Cloud


Network Routing and Addressing



Welcome to the Network Routing and Addressing module.



Today's agenda



01 [Routes and route preferences](#)

02 [IPv6](#)

03 [BYOIP \(bring your own IP\)](#)

04 [Cloud DNS](#)

05 [Lab: Traffic Steering Using Geolocation](#)

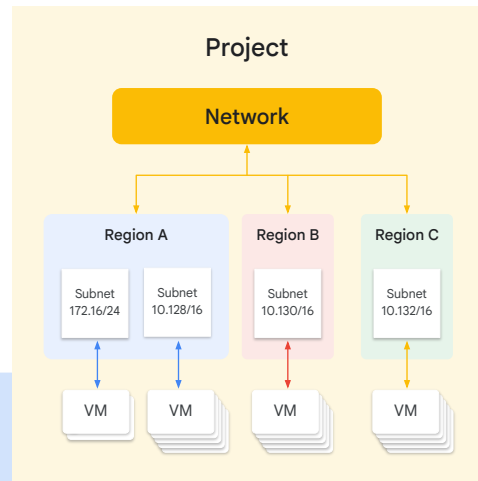
06 [Quiz](#)

In this module, we will explore foundational elements that guide traffic through the Google Cloud network. We'll start with the building blocks: routing, IPv4, and BYOIP. We'll also explore Cloud DNS.

Subnet and IP address

- ✓ Cannot overlap with other subnets.
- ✓ IP range must be a unique valid CIDR block.
- ✓ New subnet IP ranges have to fall within valid IP ranges.
- ✓ Can expand but not shrink.
- ✓ Auto mode can be expanded from /20 to /16.

Destination: 172.16.0.0/24
Next hop: Subnet B router
Destination: 0.0.0.0/0(all other),
Next hop: Internet gateway



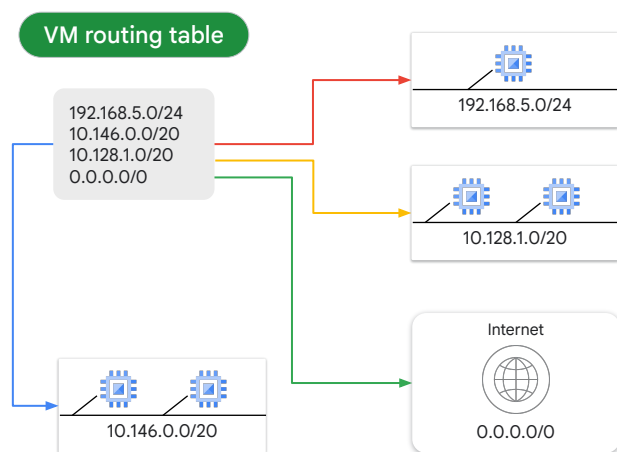
Before we cover routes, let's cover IP addresses. Speaking of IP addresses of a subnet, Google Cloud VPCs let you increase the IP address space of any subnets without any workload shutdown or downtime. This diagram illustrates a network with subnets that have different subnet masks, allowing for more instances in some subnets than others. This gives you flexibility and growth options to meet your needs, but there are some things to remember:

- The new subnet must not overlap with other subnets in the same VPC network in any region.
- Each IP range for all subnets in a VPC network must be a unique valid CIDR block.
- Also, the new subnet IP address ranges are regional internal IP addresses and have to fall within [valid IP ranges](#).
 - Subnet ranges cannot match, be narrower, or be broader than a restricted range.
 - Subnet ranges cannot span a valid RFC range and a privately used public IP address range.
 - Subnet ranges cannot span multiple RFC ranges.
- The new network range must be larger than the original, which means the prefix length value must be a smaller number. In other words, you cannot undo an expansion.

- Now, auto mode subnets start with a /20 IP range. They can be expanded to a /16 IP range, but no larger. Alternatively, you can convert the auto mode subnetwork to a custom mode subnetwork to increase the IP range further.

Routes

- ✓ Define the paths that network traffic takes from a virtual machine (VM) instance to other destinations.
- ✓ Apply to traffic that egresses a VM.
- ✓ Forward traffic to highest priority or specific route.
- ✓ Deliver traffic only if it also matches a firewall rule.
- ✓ Can be fine-tuned using network tags.



A route is created when a network or subnet is created, enabling traffic delivery from anywhere.

Routes define the paths that network traffic takes from a virtual machine (VM) instance to other destinations. These destinations can be inside your Google Cloud Virtual Private Cloud (VPC) network (for example, in another VM) or outside it. Routes match packets by destination IP address. Forward traffic to highest priority or specific route.

However, no traffic will flow without also matching a firewall rule.

A route is created when a network is created, which enables traffic delivery from anywhere. Also, a route is created when a subnet is created. This is what allows VMs on the same network to communicate.

Network tags fine-tune which route is picked. If a route has a network tag, it can be applied only to instances that have the same network tag. Routes without network tags can apply to all instances in the network.

This slide shows a simplified routing table.

Route types

Routes can be:

System-generated routes

These are default and subnet routes that are automatically created.

Custom routes

These routes are used to route traffic between subnets through a virtual appliance

VPC Network Peering routes

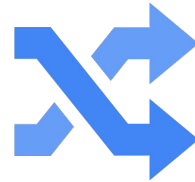
These are routes in a different VPC network connected using peering.

NCC routes

These are routes that represents a subnet IP range in a VPC spoke.

Policy-based routes

These are routes that apply to packets based on source IP, destination IP, protocol, or a combination thereof.



A route can be of many types:

there are system-generated, custom, peering, NCC, and policy based routes.

System-generated routes are simple and can be used by default. When they do not provide the desired granularity, create custom routes. For example, custom routes can be used to route traffic between subnets through a network virtual appliance. Peering routes are used for network peering.

VPC Network Peering routes in a different VPC network connected using Peering.

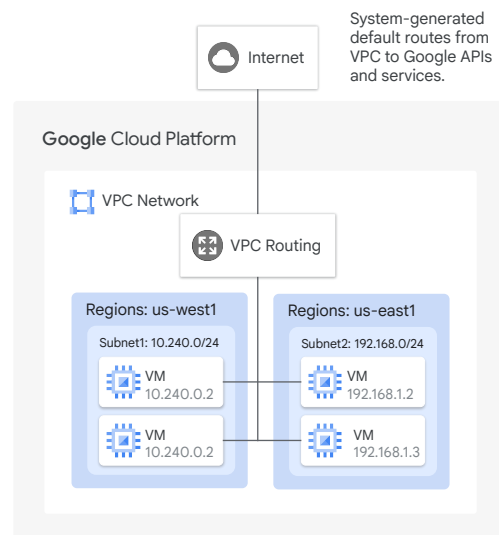
NCC routes that represents a subnet IP range in a VPC spoke.

Policy based routes apply to packets based on source IP, destination IP, protocol, or a combination thereof.

Next, you'll learn more about system-generated and custom route types. Peering, Network Connectivity Center routes, and policy-based routes are covered in another module.

Overview of system-generated default routes

- ✓ When you create a VPC network, it includes a system-generated IPv4 default route (0.0.0.0/0).
- ✓ When you create a dual-stack subnet with an external IPv6 address range, a system-generated IPv6 default route (:::0) is added to the VPC network.
- ✓ The IPv4 and IPv6 default routes define a path to external IP addresses.
- ✓ Default system-generated routes can serve as a path to Google APIs and services.



When you create a VPC network, it includes a system-generated IPv4 default route (0.0.0.0/0).

When you create a dual-stack subnet with an external IPv6 address range in a VPC network, a system-generated IPv6 default route (:::0) is added. If the default route doesn't exist, it isn't added.

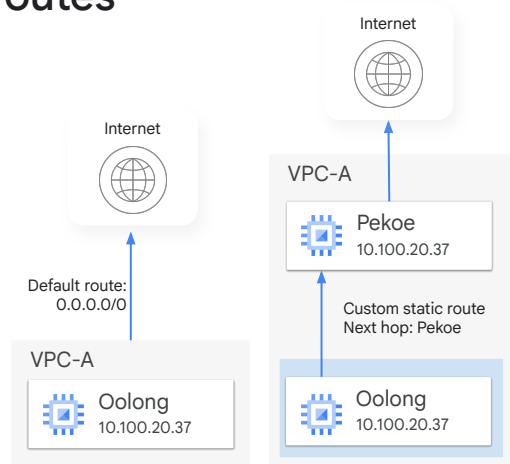
The IPv4 and IPv6 default routes that serve these purposes define a path out of the VPC network to external IP addresses on the internet.

If you access Google APIs and services without using a Private Service Connect endpoint, the default system-generated route can serve as the path to Google APIs and services. Private Service Connect enables you to publish and consume services by using the internal IP addresses that you define.

You'll learn more about Private Service Connect later in this course. For more information, in the Google Cloud documentation, refer to [Configuring Private Google Access](#) and [Accessing APIs from VMs with external IP addresses](#).

Using system-generated default routes

- ✓ A default route is used only if a route with a more specific destination does not apply to a packet.
- ✓ To completely isolate a network, delete the default route:
 - IPv4 only: to route internet traffic to a different next hop, replace the default route with a custom static or dynamic route.



Google Cloud only uses a default route if a route with a more specific destination does not apply to a packet. For information about how destination specificity and route priority influence route selection, see [Routing order](#) in the Google Cloud documentation.

To completely isolate your network from the internet or to replace the default route with a custom route, you can delete the default route.

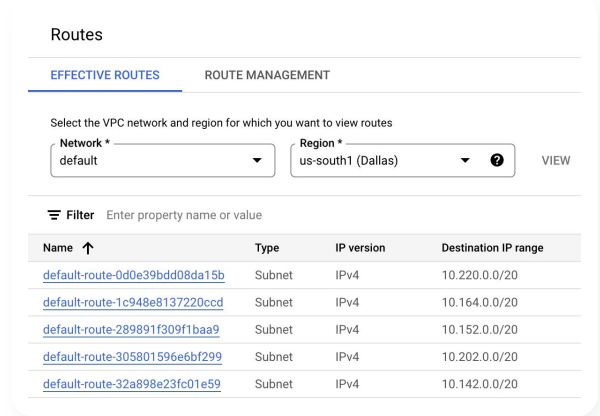
For IPv4 only, to route internet traffic to a different next hop, you can replace the default route with a custom static or dynamic route. For example, you could replace it with a custom static route whose next hop is a proxy VM. If you delete the default route and do not replace it, packets to IP ranges not covered by other routes are dropped.

If you don't have custom static routes that meet the routing requirements for Private Google Access, deleting the default route might disable Private Google Access.

Some organizations do not want a default route pointing to the internet; instead, they want the default route to point to an on-premises network. To do that, you can create a custom route. You will learn about custom routes later in this module.

Overview of system-generated subnet routes

- When you create a subnet, system-generated subnet routes are automatically created and cannot be modified or deleted.
- Subnet routes:
 - Are highest priority routes after policy-based routes.
 - Cannot be overridden by higher priority routes (lower number equals higher priority).
- Each subnet has at least one subnet route whose destination matches the subnet's primary IP range.
- If the subnet has secondary IP ranges, each secondary IP address range has a corresponding subnet route.



The screenshot shows the 'Routes' page in the AWS Management Console. It has two tabs: 'EFFECTIVE ROUTES' (selected) and 'ROUTE MANAGEMENT'. Below the tabs, there are dropdown menus for 'Network *' (set to 'default') and 'Region *' (set to 'us-south1 (Dallas)'), with a 'VIEW' button. A filter bar is present with the text 'Filter Enter property name or value'. Below this is a table of routes.

Name ↑	Type	IP version	Destination IP range
default-route-0d0e39bdd08da15b	Subnet	IPv4	10.220.0.0/20
default-route-1c948e8137220ccd	Subnet	IPv4	10.164.0.0/20
default-route-289891f309f1baa9	Subnet	IPv4	10.152.0.0/20
default-route-305801596e6bf299	Subnet	IPv4	10.202.0.0/20
default-route-32a898e23fc01e59	Subnet	IPv4	10.142.0.0/20

When you create a subnet, system-generated subnet routes are automatically created.

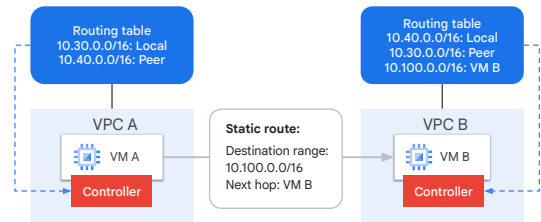
Subnet routes always have the most specific destination and cannot be overridden by higher priority routes. Recall that lower priority number indicate higher priority, so 1 would have a higher priority than 10.

Each subnet has at least one subnet route whose destination matches the primary IP range of the subnet.

If the subnet has secondary IP ranges, each secondary IP address range has a corresponding subnet route.

Overview of custom static routes

- ✓ Custom static routes forward packets to a static route next hop and are useful for all topologies.
- ✓ Benefits over dynamic routing include:
 - Quicker routing performance (lower processing overhead).
 - More security (no route advertisement).
- ✓ It has its limitations:
 - Cannot point to a VLAN attachment.
 - Require more maintenance, because routes are not dynamically updated.

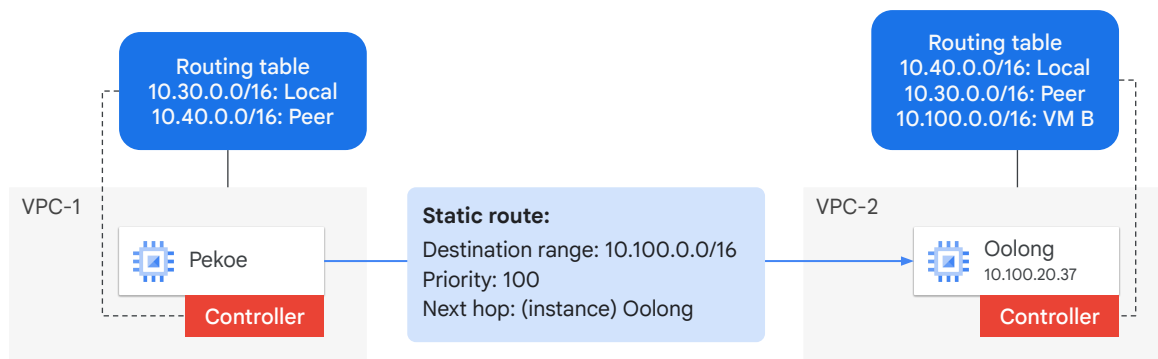


Custom static routes forward packets to a static route next hop and are useful for all topologies.

Dynamic routing generally provides quicker routing performance. Unlike dynamic routing, no processing power is devoted to maintaining and modifying the routes (hence the quicker performance). Custom static routing is more secure than dynamic routing, because there's no route advertisement.

Note these custom static routing limitations. A custom static route cannot point to a VLAN attachment. It also requires more maintenance, because routes are not dynamically updated. For example, a topology change on either network requires you to update static routes. Also, if a link fails, static routes can't reroute traffic automatically. Manually configured routes which are called [custom learned routes](#) can be used to overcome this limitation. For small, stable topologies, this is not always a significant concern.

Use case: Forward packets to a static route next hop



Here we have two VMs, Pekoe and Oolong, set up in two different VPC, VPC1, and VPC2.

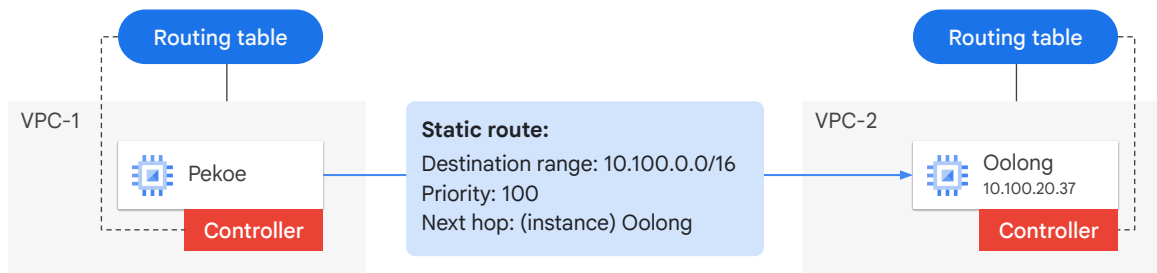
A VPN gateway is set up between these two VPCs and a two IPSEC tunnels has been created.

A static route has to be created for Pekoe to be able to ping Oolong and enable traffic to flow through the tunnel.

A static route forwards packets to a static route next hop and supports various destinations. The supported static route next hop are instances, internal passthrough Network Load Balancers, and Classic VPN tunnel next hops.

Controller

- The controller is kept informed of all routes from the network's routing table.
- Route changes are propagated to the VM controllers.

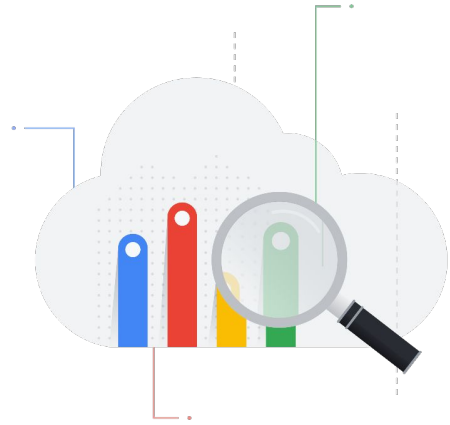


Each VM instance has a controller that is kept informed of all routes from the network's routing table. Route changes are propagated to the VM controllers. When you add or delete a route, the set of changes is propagated to the VM controllers. In this example, if you change any of the routes to the Oolong VM, Pekoe can still route packets to Oolong.

Different ways to create custom static routes

Create custom static routes:

- ✓ Manually, by using either the Google Cloud console, gcloud CLI compute routes create command, or the routes.insert API.
- ✓ When creating a Classic VPN tunnel without dynamic routing in the Google Cloud console, Cloud VPN may automatically generate static routes.
- ✓ You can also use an IaC system such as Terraform.



You can create custom static routes either manually or automatically.

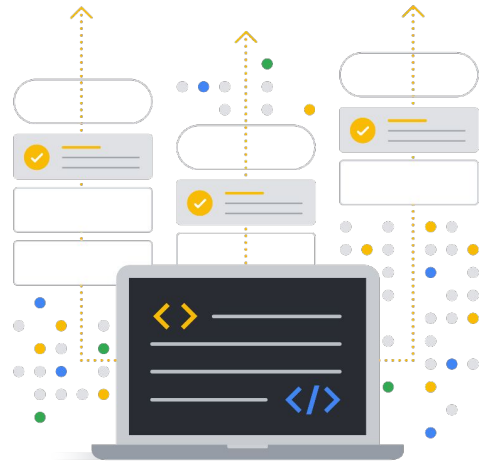
To create custom static routes manually, use the Google Cloud app, the gcloud CLI compute routes create command, or the routes.insert API.

When creating a Classic VPN tunnel without dynamic routing in the Google Cloud console, Cloud VPN may automatically generate static routes. To create the routes, you can also use the Google Cloud app to create a Classic VPN tunnel with policy-based routing or as a route-based VPN. For more information, see Cloud VPN networks and tunnel routing.

Another code-based approach would be to use an IaC system such as Terraform.

Dynamic routes

- ✓ Are managed by Cloud Routers.
- ✓ Typically represent IP address ranges outside your VPC network, which are advertised from a BGP peer.
- ✓ Dynamic routes are used by:
 - Dedicated Interconnect
 - Partner Interconnect
 - Cross-Cloud Interconnect
 - HA VPN tunnels
 - Classic VPN tunnels that use dynamic routing
 - NCC Router appliances



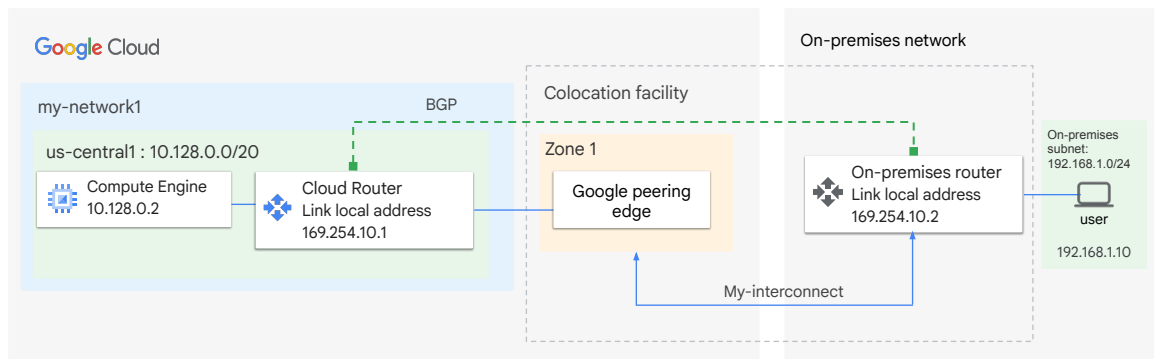
Dynamic routes are managed by Cloud Routers in the VPC network. Their destinations always represent IP address ranges outside your VPC network, which are advertised from a BGP peer router. BGP peer routers are typically outside the Google network (like on-premises or on another cloud provider).

Dynamic routes are used by:

- Dedicated Interconnect
- Partner Interconnect
- HA VPN tunnels
- Classic VPN tunnels that use dynamic routing
- NCC Router appliances

A dynamic routing example

- Routes are added and removed automatically by Cloud Routers in your VPC network.
- Routes apply to VMs according to the VPC network's dynamic routing mode.



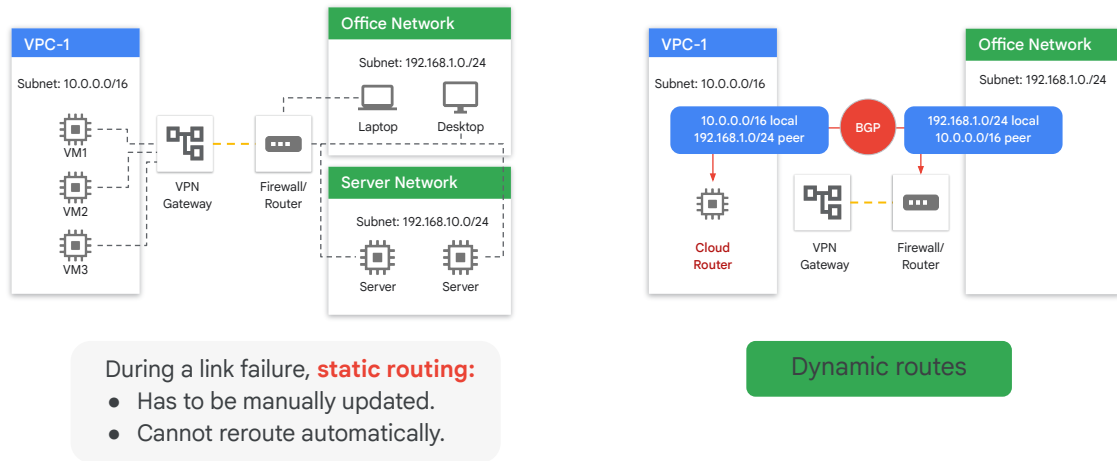
Routes are added and removed automatically by Cloud Routers in your VPC network.

The routes apply to VMs according to the VPC network's dynamic routing mode.

This example shows a VPC network connected to an on-premises network that uses Dedicated Interconnect.

Cloud Router handles the BGP advertisements and adds them as custom routes. Cloud Router creates a BGP session for the VLAN attachment and its corresponding on-premises peer router. The Cloud Router receives the routes that your on-premises router advertises. These routes are added as custom dynamic routes in your VPC network. The Cloud Router also advertises routes for Google Cloud resources to the on-premises peer router.

Use case: Automatically reroute traffic




For large organizations that turn up several networks and test them, static routes can be painful. In the above topology, VMs in the Google VPC route traffic to the VPN gateway through static routes. The VPN gateway encrypts traffic to and from the on-premises network. The on-premises environment has a firewall and a router that knows how to route traffic to the Cloud VPN gateway. As the on-premises network expands to the server network, routes have to be manually configured to route traffic to the resource in the server network.


A topology change on either network requires you to manually update static routes. Also, static routes cannot automatically reroute traffic when there is a link failure.

A solution is for a network to automatically and rapidly discover topology changes and then route traffic accordingly to minimize disruption. This is exactly the function of Cloud Router.

Whenever a link fails, Cloud Router will automatically reroute traffic if another path is possible. Cloud Router peers with an on-premises VPN gateway or router. The router exchanges topology information through a border gateway protocol (BGP). Cloud Router advertises subnets from its VPC network to the on-premises gateway via BGP. Then, topology changes automatically between your VPC and on-premises network.



Today's agenda



01 Routes and route preferences

02 [IPv6](#)

03 BYOIP (Bring your own IP)

04 Cloud DNS

05 Lab: Traffic Steering Using Geolocation

06 Quiz

Subnets and IPv6 support

- VPC networks now support IPv6 addresses.
- Support for IPv6 addresses can vary per subnet.
- To support IPv6, Google Cloud has introduced the concept of a subnet stack.
 - Single-stack and dual-stack subnets support IPv4 and IPv6.
- IPv6 addresses can be assigned to objects in a subnet that supports IPv6.



SUBNETS					
STATIC INTERNAL IP ADDRESSES FIREWALLS ROUTES VPC NETWORK PEERING					
ADD SUBNET FLOW LOGS					
Filter Enter property name or value					
<input type="checkbox"/>	Name	Region	Stack Type	Internal IP ranges ↓	External IP ranges
<input type="checkbox"/>	sn-india-mumbai	asia-south1	IPv4 and IPv6	192.168.6.0/24, fd20:476:674b	None
<input type="checkbox"/>	sn-eu-us-central	us-central1	IPv4 and IPv6	192.168.1.0/24	2600:1900:4000:4a23:0:0:0:0/64

VPC networks now support IPv6 addresses.

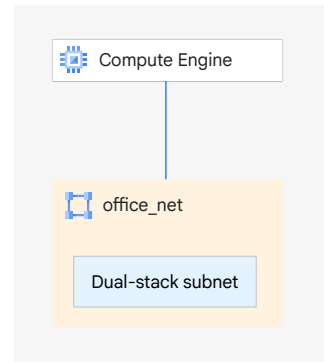
Support for IPv6 addresses can vary per subnet. To support IPv6, Google Cloud has introduced the concept of a subnet stack. The subnet stack defines the type of address that can be assigned to objects in the subnet.

Single and dual-stack subnets support IPv4 and IPv6.

IPv6 addresses can be assigned to objects in a subnet that supports IPv6. In other words, you can only assign IPv6 addresses to objects in a dual-stack subnet.

To use IPv6, set up a dual-stack subnet

- You can configure the IPv6 access type as internal or external.
- Internal IPv6 addresses are used for communication between VMs within VPC networks.
- External IPv6 addresses:
 - Can be used for communication between VMs within VPC networks.
 - Are also routable on the internet.
- Connected VMs inherit the IPv6 access type from the subnet.



You can configure the IPv6 access type to be internal or external.

Internal IPv6 addresses are used for VM to VM communication within VPC networks. These use unique local addresses (ULAs), which can only be routed within VPC networks and cannot be routed to the internet.

External IPv6 addresses can be used for communication between VMs within VPC networks. These use global unicast addresses (GUAs) and are also routable on the internet.


Connected VMs inherit the IPv6 access type from the subnet.

IPv6 caveat




Dual-stack subnets are not supported on auto mode VPC networks or legacy networks.

When configuring your VPC networks and subnets to use an IPv6 address, consider the caveat that dual-stack subnets are not supported on auto mode VPC networks or legacy networks. If you have an auto mode VPC network that you want to add dual-stack subnets to, you can convert the auto mode VPC network to custom mode.



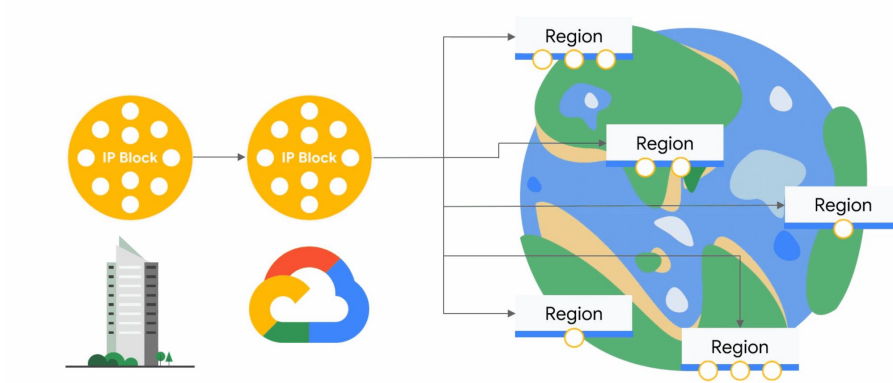
Today's agenda



- 01 Routes and route preferences
- 02 IPv6
- 03 [BYOIP \(bring your own IP\)](#)
- 04 Cloud DNS
- 05 Lab: Traffic Steering Using Geolocation
- 06 Quiz

The next topic we will be discussing is BYOIP.

Use case: Bring your own IP to Google's network

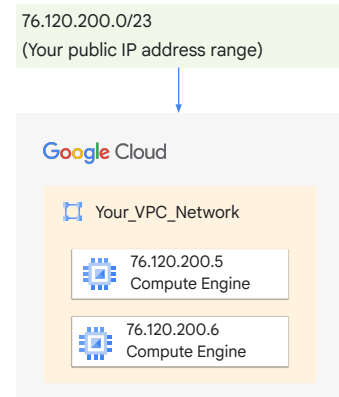


Enterprises want to move their applications to the cloud, but worry about having to swap their IP addresses for ones from their cloud provider. We hear from our customers that managing the migration of IP addresses can be one of the most challenging aspects of a cloud migration for network administrators.

Here at Google Cloud, we now allow you to Bring Your Own IP (BYOIP) addresses to Google's network infrastructure across all our 24 regions. By bringing over your own IP addresses, you can accelerate your migration while minimizing downtime, as well as significantly reduce your networking infrastructure costs. With Google Cloud, your BYOIP prefixes can be broken into blocks as small as 16 addresses (/28), can be distributed to any region, and can also be used for global load balancers, creating more flexibility with the resources you already have. You can also advertise the IP addresses you bring to Google Cloud globally to all peers.

Introduction to BYOIP (bring your own IP address)

- BYOIP enables customers to:
 - Assign IP addresses from a public IP range that they own to Google Cloud resources.
 - Route traffic directly from the internet to their VMs.
- Google Cloud manages these BYOIP addresses in the same way as Google-provided IP addresses, except that:
 - The IP addresses are available only to the customer who brought them.
 - Idle or in-use IP addresses incur no charges.



BYOIP enables customers to assign IP addresses from a public IP range that they own to Google Cloud resources. With BYOIP, customers can route traffic directly from the internet to their VMs without having to go through their own physical networks.

After the IP addresses are imported, Google Cloud manages them in the same way as Google-provided IP addresses, with these exceptions:

- The IP addresses are available only to the customer who brought them.
- Idle or in-use IP addresses incur no charges.

BYOIP guidelines

The object that the IP address is assigned to:

- Can have a regional scope or a global scope and must support an external address type.

BYOIP can be used

As the peer IP address of a Classic VPN gateway.

To create external forwarding rules in GKE ingress for external Application Load Balancers.

To configure static IP addresses on VM creation in a MIG.

In Shared VPC host projects and use the host project IP addresses in the service projects.

BYOIP is not supported

As the peer IP address of a HA VPN gateway and as the external IP address of Classic VPN or HA VPN gateway.

In Google Kubernetes Engine nodes and Pods.

MIGs that automatically allocate IP addresses to VMs.

Shared VPC does not support creating BYOIP addresses in service projects.

The object that the IP address is assigned to can have a regional scope, like a VM or the forwarding rule of a network load balancer. It can also have a global scope, like the forwarding rule of a global external Application Load Balancer.

It must support an external address type, because BYOIP ranges will be advertised by Google to the public internet.

BYOIP, bring your own IP, addresses are a way to incorporate your existing static external IP addresses into cloud environments. They are compatible with most resources that support static external IP addresses.

You can leverage BYOIP addresses for Classic VPN gateway tunnels (as peer IP addresses), external Google Kubernetes Engine forwarding rules, and configuring static IP addresses for VMs within stateful managed instance groups.

However, there are certain limitations. BYOIP addresses cannot be used as peer IP addresses for HA VPN gateway tunnels or as external IP addresses for VPN gateway tunnels in general. Additionally, Shared VPC service projects, Google Kubernetes Engine nodes and Pods, and managed instance groups with automatic IP allocation do not support the use of BYOIP addresses.

BYOIP caveats

01

BYOIP prefixes cannot overlap with subnet or alias ranges in the VPC.

02

The IP address must be IPv4.

03

Overlapping BGP route announcements can be problematic.




BYOIP prefixes cannot overlap with subnet or alias ranges in the VPC used by the customer.


For BYOIP, the IP address must be IPv4. Importing IPv6 addresses is not supported.

Overlapping BGP route announcements can be problematic. BGP is a routing protocol that picks the most efficient route to send a packet. If Google and another network advertise the same route with matching or mismatched prefix lengths, BGP cannot work properly. You might experience unexpected routing and packet loss.

For example: suppose you're advertising a 203.0.112.0/20 address block and you're using BGP to route packets. You could bring a 203.0.112.0/23 address block that you own to Google using BYOIP and set it up to route externally. Because the /23 block is contained within the /20 block, BGP route announcements may overlap. If you're maintaining the routing registry correctly, BGP routing practices cause the more specific route to take precedence. Thus, the /23 block will take precedence over the /20 block. However, if the /23 route ever stopped being advertised, the /20 block could be used.



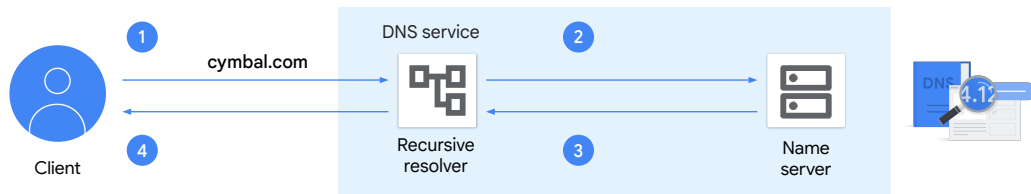
Today's agenda



- | | |
|----|---|
| 01 | Routes and route preferences |
| 02 | IPv6 |
| 03 | BYOIP (bring your own IP) |
| 04 | Cloud DNS |
| 05 | Lab: Traffic Steering Using Geolocation |
| 06 | Quiz |

Next, let's talk about Cloud DNS.

A simple DNS primer



- 1 A client makes a DNS request to find the IP address; the request is sent to a recursive resolver.
- 2 A recursive resolver requests the IP address from a name server.
- 3 The name server responds with the IP address.
- 4 The recursive resolver sends the IP to the client.

Before we talk about Google Cloud DNS, let's quickly review how DNS, or Domain Name System, works.

DNS provides a lookup for sites on the internet. You can think of it as a phone book, but instead of using the name of an organization to look up its phone number, you use the name of an organization to find an IP address. A DNS service is provided by your ISP, or internet service provider.

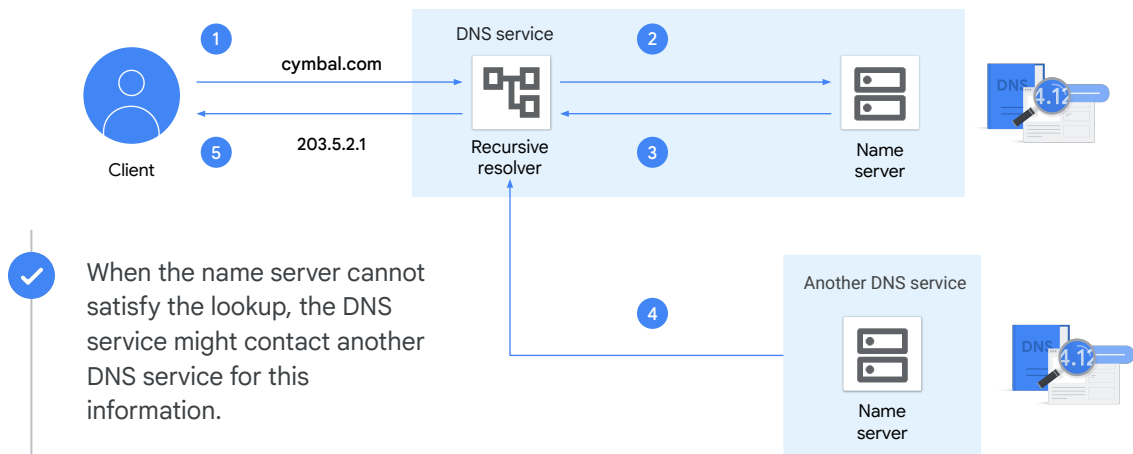
For example, suppose a request comes from a client computer to access cymbal.com.

To direct the client computer to the cymbal.com site, the internet service provider needs the IP address of cymbal.com.

The ISP connects to get this information from its DNS service.

The DNS service recursive resolver issues a request to look up the IP address of cymbal.com from one of its name servers. The name server responds with the IP and the recursive resolver sends the IP to the client.

A simple DNS primer



When the name server cannot satisfy the lookup, the DNS service might contact another DNS service for this information.

Some organizations don't rely on their ISP to provide DNS service, so they create and maintain their own DNS servers. Organizations sometimes do this to limit or customize the information that is returned, or because they can achieve better performance if they use their own DNS servers. Alternatively, they can purchase DNS services from another organization.

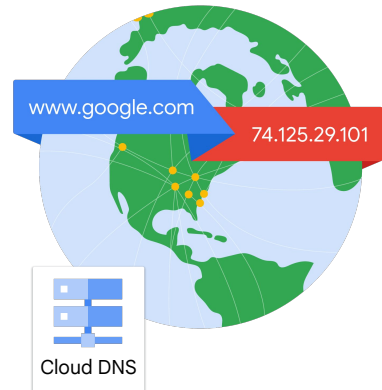
Obviously, there's a lot more that can be said about DNS and its components, but that's not covered in this course.

Various companies provide DNS services. Google Cloud is one of them.

Use Cloud DNS to host DNS zones

Cloud DNS can:

- ✓ Create and update millions of DNS records.
- ✓ Update by using the Google Cloud console, command line, or API.
- ✓ Google guarantees 100%* availability of its authoritative name servers.



Cloud DNS lets you create and update millions of DNS records without the burden of managing your own DNS servers and software.

Instead, you use a simple user interface, command-line interface, or API.

On Linux, by default, the VM's metadata server (169.254.169.254) resolves internal DNS names.

On Windows, by default, the subnet's default gateway resolves internal DNS names.

Google Cloud provides a monthly uptime percentage of serving DNS queries from at least one of the Google-managed authoritative name servers to customers of 100% SLO.

Important notes:

Exclusions: it's crucial to understand there are certain situations, such as maintenance, force majeure events, or actions on your part that can void this SLA.

Intermittent issues: the downtime definition specifically focuses on at least 60 consecutive seconds of unavailability. Intermittent issues less than a minute might not count towards the SLA.

Private and public DNS zones

Private DNS zones	Public DNS zones
Private zones are used to provide a namespace that is visible only inside the VPC or hybrid network environment.	Public zones are used to provide authoritative DNS resolution to clients on the public internet.
For example, an organization would use a private zone for a domain dev.gcp.example.com, which is reachable only from within the company intranet.	For example, a business would use a public zone for its external website, cymbal.com, which is directly accessible from the internet.

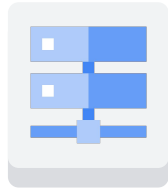
Private zones are used to provide a namespace that is visible only inside the VPC or hybrid network environment. For example, an organization would use a private zone for a domain dev.gcp.example.com, which is reachable only from within the company intranet.

Public zones are used to provide authoritative DNS resolution to clients on the public internet. For example, a business would use a public zone for its external website, cymbal.com, which is accessible directly from the internet.

Don't confuse the concept of a public zone with Google Public DNS (8.8.8.8). Google Public DNS is just a public recursive resolver.

For more information, refer to the Cloud DNS documentation, <https://cloud.google.com/dns/docs/>.

Introduction to Cloud DNS policies



Cloud DNS

Cloud DNS policies provide a flexible way to refine how your organization uses DNS.

After you create the DNS zone and artifacts needed for lookups, create Cloud DNS policies.

Cloud DNS policies provide a flexible way to define how your organization uses DNS.

After you create the DNS zones and artifacts needed for lookups, create Cloud DNS policies.

Supported Cloud DNS policies



Server policies apply private DNS configuration to a VPC network.



Response policies enable you to modify the behavior of the DNS resolver by using rules that you define.



Routing policies: steer traffic based on geolocation or round robin.

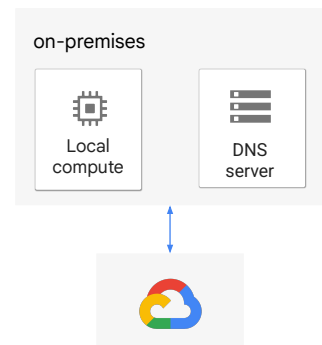
Cloud DNS supports different types of policies:

- Server policies apply private DNS configuration to a VPC network.
- Response policies enable you to modify the behavior of the DNS resolver by using rules that you define.
- Routing policies steer traffic based on geolocation or round robin.

Next, let's look at each of these types of policies.

Server policies

- Use server policies to set up hybrid deployments for DNS resolutions.
- Each VPC network can have one DNS server policy.
- You can set up an inbound server policy depending on the direction of DNS resolutions.
- For workloads that use an on-premises DNS resolver, use an outbound server policy to set up DNS forwarding.
- If you want on-premises workloads to resolve Cloud DNS Private zones, set up an inbound server policy.



Use server policies to set up hybrid deployments for DNS resolution.

You can configure one DNS server policy for each Virtual Private Cloud (VPC) network.

You can set up an inbound server policy depending on the direction of the DNS resolutions.

If your workloads plan to use an on-premises DNS resolver, you can set up DNS forwarding zones by using an outbound server policy.

If you want your on-premises workloads to resolve Cloud DNS private zones, you can set up an inbound server policy.

The policy can specify inbound DNS forwarding, outbound DNS forwarding, or both. In this section, inbound server policy refers to a policy that permits inbound DNS forwarding. Outbound server policy refers to one possible method for implementing outbound DNS forwarding. If a policy implements the features of both, it can be an inbound server policy and an outbound server policy.

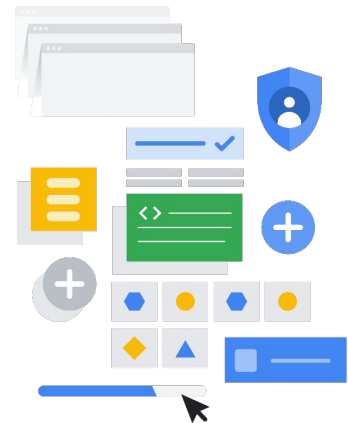
DNS server policies are not available for legacy networks. DNS server policies require VPC networks.

For detailed information about server policies, see [Server policies overview](#) in the

Google Cloud documentation. To configure and apply DNS server policies, see [Apply Cloud DNS server policies](#) in the Google Cloud documentation.

Response policies

- A response policy:
 - Is a Cloud DNS private zone concept that contains rules instead of records.
 - Lets you introduce customized rules in DNS servers within your network that the DNS resolver consults during lookups.
- If a rule in the response policy affects the incoming query, it's processed (otherwise, the lookup proceeds normally).
- Response policies are not DNS zones and are managed separately in the API



A response policy is a Cloud DNS private zone concept that contains rules instead of records. These rules can be used to achieve effects similar to the DNS response policy zone (RPZ) draft concept. In other words, you can use response policies to create a DNS firewall by returning modified DNS results to clients. For example, you can use response policies to block access to specified HTTP servers.

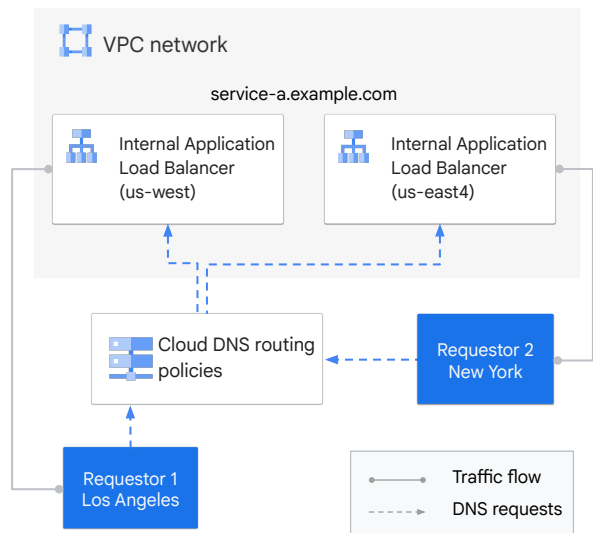
The response policy feature lets you introduce customized rules in DNS servers within your network that the DNS resolver consults during lookups.

If a rule in the response policy affects the incoming query, it's processed. Otherwise, the lookup proceeds normally. For more information, see [Manage response policies and rules](#) in the Google Cloud documentation.

A response policy is different from an RPZ (response policy zone). An RPZ is an otherwise normal DNS zone with specially formatted data that causes compatible resolvers to do special things. Response policies are not DNS zones and are managed separately in the API. To create and modify response policies in Cloud DNS, use the ResponsePolicies API. Response policies are separate from ManagedZones and cannot be managed by using either the ManagedZones API or the RRSet API.

Routing policies

- ✓ DNS routing policies steer your traffic based on specific criteria..
- ✓ Google Cloud supports three types of DNS routing policies:
 - Weighted round robin
 - Geolocation
 - Failover (private zones only)



DNS routing policies let you steer your traffic based on specific criteria. Google Cloud supports three types of DNS routing policies: weighted round robin, geolocation, geofencing and failover.

A weighted round robin routing policy lets you specify different weights per DNS target, and Cloud DNS ensures that your traffic is distributed according to the weights. You can use this policy to support manual active-active or active-passive configurations. You can also split traffic between production and experimental versions of software.

A geolocation routing policy lets you map traffic originating from source geographies (Google Cloud regions) to specific DNS targets. Use this policy to distribute incoming requests to different service instances based on the traffic's origin. You can use this feature with the internet, with external traffic, or with traffic originating within Google Cloud and bound for internal load balancers. Google Cloud uses the region where queries enter Google Cloud as the source geography.

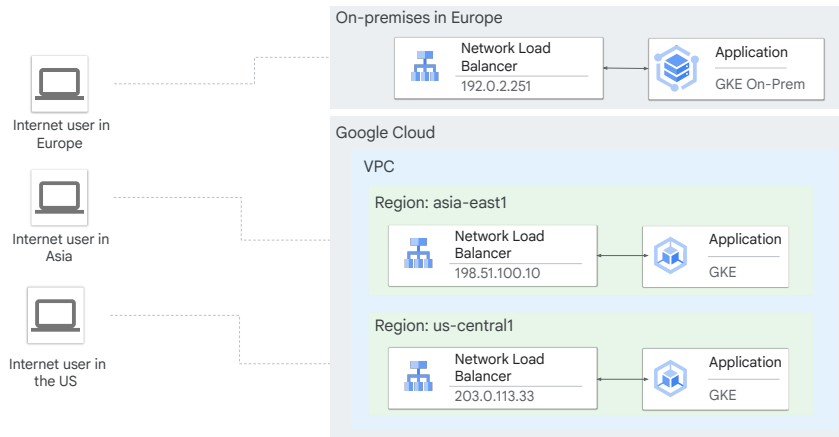
A failover routing policy lets you set up active backup configurations. This option is only available for private zones.

Next, you will implement a geolocation routing policy as part of a lab exercise. An example is shown on the screen; routing policies use geolocation to route requests to the closest load balancer.

In a lab exercise, you will configure a routing policy that uses geolocation.

To create, edit, or delete DNS routing policies, see [Manage DNS routing policies](#) in the Google Cloud documentation.

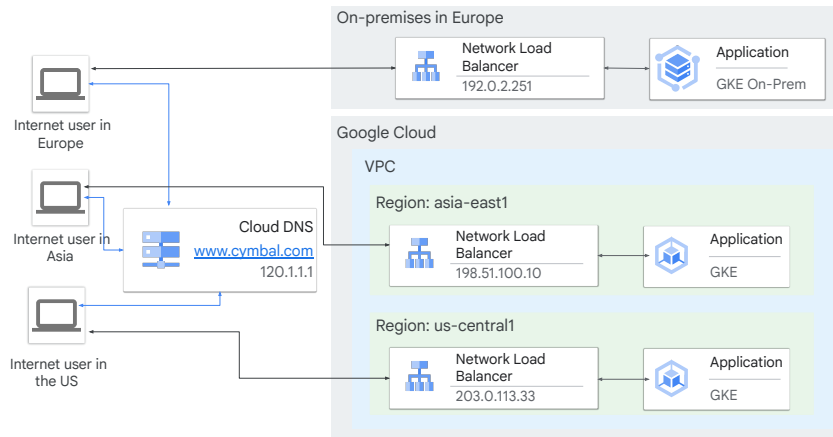
Use case: Centralize and simplify DNS management



Let us look at a scenario before we move to the lab exercise. Carl is a network engineer at Cymbal Corporation. Cymbal has a hybrid environment, with an on-premises datacenter in Europe and a Google Cloud environment with VPC subnets in Asia and the US. Carl does not want to juggle managing multiple DNS providers for their global web presence. Fragmented setup leads to inconsistencies, manual configuration overhead, and potential for human error.

Carl is looking for a solution that can provide a single endpoint for a hybrid application so that external clients in multiple regions resolve `www.cymbal.com` to the nearest region.

Use case: Carl can use Cloud DNS to centralize DNS management



Carl can leverage Cloud DNS for its global anycast network for low-latency DNS resolution and high availability. The intuitive interface simplifies record management, offering features like health checks and traffic routing.

Cloud DNS provides global traffic management for www.cymbal.com, routing users to the closest available server. Asia and US-based visitors are directed to Network load balancers within their respective regions, which connect them to GKE instances. For European users, a load balancer in a European data center handles traffic, forwarding to GKE on VMware. This setup demonstrates flexibility in application hosting.

Carl can create this configuration by using the following steps:

1. Carl creates the Network load balancers and the on-premises load balancer in each region.
2. Carl sets up a public Cloud DNS zone for his domain. Cloud DNS automatically assigns anycast name servers to handle domain record storage. These name servers are strategically located in a fully managed, shared environment.
3. Then, they create a DNS routing policy. In the policy, you set the type to GEO and you set the --routing-policy-data value to a list of target regions that are mapped to the corresponding Network load balancers.

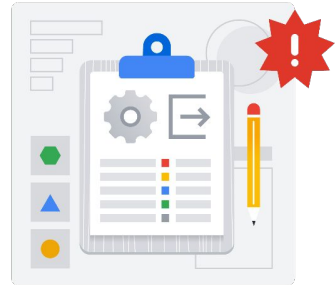
Routing policy caveats

01

Only one type of routing policy can be applied to a resource record set at a time.


02

Nesting or otherwise combining routing policies is not supported.




When configuring routing policies, consider these caveats:

- Only one type of routing policy can be applied to a resource record set at a time.
- Nesting or otherwise combining routing policies is not supported.



Today's agenda



- 01 Routes and route preferences
- 02 IPv6
- 03 BYOIP (bring your own IP)
- 04 Cloud DNS
- 05 [Lab: Traffic Steering Using Geolocation](#)
- 06 Quiz

Next, let's talk about Cloud DNS.


Lab Intro

Traffic Steering Using Geolocation


In this lab, you will configure and test the geolocation routing policy. The geolocation routing policy applies the nearest match for the source location when the traffic source location doesn't match any policy items exactly.

The lab tasks are to:

- Launch client VMs, one in each region.
- Launch server VMs, one in each region except asia-south1.
- Create a private zone, like example.com.
- Create a geolocation routing policy using gcloud commands.
- Test the configuration.



Today's agenda



- 01 Routes and route preferences
- 02 IPv6
- 03 BYOIP (bring your own IP)
- 04 Cloud DNS
- 05 Lab: Traffic Steering Using Geolocation
- 06 [Quiz](#)

Next, let's talk about Cloud DNS.

Quiz | Question 1

Question

You must create a VM that has an IPv6 address. How do you do it?

- A. Create an IPv6-only subnet, and create the VM with an IPv6 address.
- B. Create a dual-stack subnet, and create the VM with an IPv6 address.
- C. Create a single-stack network, and create the VM with an IPv6 address.
- D. Create a dual-stack network, and create the VM with an IPv6 address.

Quiz | Question 1

Answer

You must create a VM that has an IPv6 address. How do you do it?

- A. Create an IPv6-only subnet, and create the VM with an IPv6 address.
- B. Create a dual-stack subnet, and create the VM with an IPv6 address.
- C. Create a single-stack network, and create the VM with an IPv6 address.
- D. Create a dual-stack network, and create the VM with an IPv6 address.



Explanation:

- A. That's incorrect. There's no subnet that supports only IPv6.
- B. Correct. A dual-stack subnet supports IPv4 and IPv6 addresses. There's no subnet stack that supports only IPv6.
- C. That's incorrect. Stacks are created at the subnet level, not the network level.
- D. That's incorrect. Stacks are created at the subnet level, not the network level.

Quiz | Question 2

Question

To set up hybrid deployments for DNS resolution, which type of DNS policy should you use?

- A. Routing policy
- B. Response policy
- C. Server policy
- D. Traffic policy

Quiz | Question 2

Answer

To set up hybrid deployments for DNS resolution, which type of DNS policy should you use?

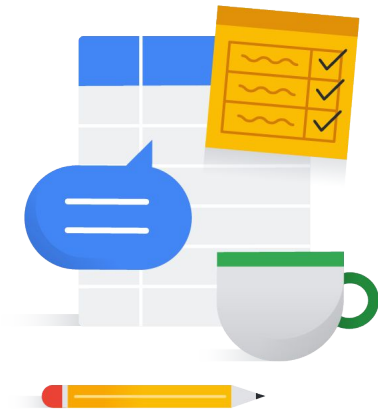
- A. Routing policy
- B. Response policy
- C. Server policy
- D. Traffic policy



Explanation:

- A. That's incorrect. A routing policy lets you steer traffic based on geolocation or weighted round robin.
- B. That's incorrect. A response policy is a Cloud DNS private zone concept that contains rules instead of records.
- C. Correct. Use one or more server policies to set up hybrid deployments for DNS resolution.
- D. That's incorrect. Cloud DNS does not include a traffic policy. Cloud DNS does include a routing policy, which lets you steer traffic based on geolocation or weighted round robin.

Debrief



In this module, you learned about some fundamental Google Cloud VPC networking concepts. We began with an overview of Google Cloud VPC networks. We then discussed how to use IPv6 addressing—and the configuration that must be done at the subnet level. After that, we discussed routes and route preferences, including system-generated routes, custom routes, and dynamic routes. We continued with information about bringing existing external IP addresses into Google Cloud, also known as BYOIP. Then, we discussed using multiple network interfaces on Compute Engine VMs, as well as some important caveats. After that, we used Cloud DNS policies to refine how an organization uses Cloud DNS. We concluded the module with a lab exercise and a brief quiz to test your knowledge of what you've learned.



THANK YOU