EXAM DIAGNOSTIC QUESTIONS

Saturday, October 22, 2022 6:50 PM

Networking in Google Cloud: Defining and Implementing Networks

Saturday, October 22, 2022 6:50 PM

In auto mode network vpc it is expandable upto /16 but defaults to /20 but in custom mode expandable upto any RFC 1918 size
Rfc 1918 SUBNETS 0.0 – 10.255. 255.255 (10/8 prefix)
The ephemeral external IP addresses will go off when we stop the vm and a new ip address is created when we start the vm again
When creating multiple nic's in vm the internal dns only associates to nic0 and can attach upto 8 NIC's per VM depending on VM
Like if vm has <=2vCPU then only 2 NIC'S else >2vCPU then 1nic per VCPU (max 8)

Network Viewer: Read only access to all networking resources
Network Admin: Permissions to create,modify, and delete networking resources, except for firewall rules and SSL certificates
Security Admin: Permissions to create , modify, and delete firewall rules and SSL certificates

For provisioning shared VPC we need following roles:

1) Organization admin(nominates a shared vpc admin(compute.xpnAdmin))
2) Shared VPC admin (enables shared vpc for host projects,atatches service projects)
3) Service project admin(should have network user,compute instance admin,project owner in the service project)

When setting up shared vpc we can share all subnets or selective subnets to the service projects(which should have compute engine api enabled before attaching them to this)
When creating a vm instance we can select the networks shared with me option to select a host projects network

When A peered to B and A peered to C - C,B cannot be peered(transitive peering)

Cloud armor:

- Protects against ddos attacks at L3,4
- Has a managed protection plus package (subscription) which has a names ip address lists to use in the security policy and premium response support
- Integrates with security command center to alert potential layer 7 attacks
- Allowed traffic spike,increasing deny ratio are some events we get with the integration

We can also use a non google cloud backends in external https load balancing by creatign a neg for on prem and connecting onprem using cloud vpn or interconnect

sudo apt-get -y install siege (for load test)
siege -c 250 http://$LB_IP

Cloud CDN:

- Cache modes control the factors that determine whether or not to cache our content
  - USE_ORIGIN_HEADERS
  - CACHE_All_STATIC
  - FORCE_CACHE_ALL

Cloud CDN content can originate from different types of backends:

- Compute Engine virtual machine (VM) instance groups
- Zonal network endpoint groups (NEGs)
- Internet network endpoint groups (NEGs), for endpoints that are outside of Google Cloud (also known as custom origins)
- Google Cloud Storage buckets
  Cache invalidation lets you remove an object from Cloud CDN caches before its normal expiration time. Formatting: host (optional) followed by a path. Host can't include *. Path must start with /, and can't include ? or #. Path can't include * except as the last character after /

- If we have a new object in place of an ob ject being cached then we need to do this invalidatio to replace the object I cache

Networking in Google Cloud Platform

Sunday, October 23, 2022 4:02 PM

Dedicated interconnect is a stable same cost but partner connect is not a stable cost or stable speed

Network connectivity center:

- hub
  - Global management resource
  - One hub per project
  - Function of the hub varies
- Spokes
  - 1/more network resource connected to a hub
  - Example:router appliance,vpn
- Router appliance
  - Enables the installation of a network virtual appliance as the backing resource for a spoke
    - Install virtual appliance image on gce vm
    - Establish bgp peering between the vm and a cloud router

The interconnect option gives direct access to RFC 1918 ips whereas peering access to google public ips only

Creating a ha vpn through gcloud sdk

- gcloud compute vpn-gateways create vpc-demo-vpn-gw1 --network vpc-demo --region us-central1
- gcloud compute vpn-gateways create on-prem-vpn-gw1 --network on-prem --region us-central1
- gcloud compute routers create vpc-demo-router1 \
  --region us-central1 \
  --network vpc-demo \
  --asn 65001
- gcloud compute routers create on-prem-router1 \
  --region us-central1 \
  --network on-prem \
  --asn 65002
- gcloud compute vpn-tunnels create vpc-demo-tunnel0 \
  --peer-gcp-gateway on-prem-vpn-gw1 \
  --region us-central1 \
  --ike-version 2 \
  --shared-secret [SHARED_SECRET] \
  --router vpc-demo-router1 \
  --vpn-gateway vpc-demo-vpn-gw1 \
  --interface 0
- gcloud compute vpn-tunnels create vpc-demo-tunnel1 \
  --peer-gcp-gateway on-prem-vpn-gw1 \
  --region us-central1 \
  --ike-version 2 \
  --shared-secret [SHARED_SECRET] \
  --router vpc-demo-router1 \
  --vpn-gateway vpc-demo-vpn-gw1 \
  --interface 1
- gcloud compute vpn-tunnels create on-prem-tunnel0 \
  --peer-gcp-gateway vpc-demo-vpn-gw1 \
  --region us-central1 \
  --ike-version 2 \
  --shared-secret [SHARED_SECRET] \
  --router on-prem-router1 \
  --vpn-gateway on-prem-vpn-gw1 \
  --interface 0
- gcloud compute vpn-tunnels create on-prem-tunnel1 \
  --peer-gcp-gateway vpc-demo-vpn-gw1 \
  --region us-central1 \
  --ike-version 2 \
  --shared-secret [SHARED_SECRET] \
  --router on-prem-router1 \
  --vpn-gateway on-prem-vpn-gw1 \
  --interface 1
- gcloud compute routers add-interface vpc-demo-router1 \
  --interface-name if-tunnel0-to-on-prem \
  --ip-address 169.254.0.1 \
  --mask-length 30 \
  --vpn-tunnel vpc-demo-tunnel0 \
  --region us-central1
- gcloud compute routers add-bgp-peer vpc-demo-router1 \
  --peer-name bgp-on-prem-tunnel0 \
  --interface if-tunnel0-to-on-prem \
  --peer-ip-address 169.254.0.2 \
  --peer-asn 65002 \
  --region us-central1
- gcloud compute routers add-interface vpc-demo-router1 \
  --interface-name if-tunnel1-to-on-prem \
  --ip-address 169.254.1.1 \
  --mask-length 30 \
  --vpn-tunnel vpc-demo-tunnel1 \
  --region us-central1
- gcloud compute routers add-bgp-peer vpc-demo-router1 \
  --peer-name bgp-on-prem-tunnel1 \
  --interface if-tunnel1-to-on-prem \
  --peer-ip-address 169.254.1.2 \
  --peer-asn 65002 \
  --region us-central1
- gcloud compute routers add-interface on-prem-router1 \
  --interface-name if-tunnel0-to-vpc-demo \
  --ip-address 169.254.0.2 \
  --mask-length 30 \
  --vpn-tunnel on-prem-tunnel0 \
  --region us-central1
- gcloud compute routers add-bgp-peer on-prem-router1 \
  --peer-name bgp-vpc-demo-tunnel0 \
  --interface if-tunnel0-to-vpc-demo \
  --peer-ip-address 169.254.0.1 \
  --peer-asn 65001 \
  --region us-central1
- gcloud compute routers add-interface on-prem-router1 \
  --interface-name if-tunnel1-to-vpc-demo \
  --ip-address 169.254.1.2 \
  --mask-length 30 \
  --vpn-tunnel on-prem-tunnel1 \
  --region us-central1
- gcloud compute routers add-bgp-peer on-prem-router1 \
  --peer-name bgp-vpc-demo-tunnel1 \
  --interface if-tunnel1-to-vpc-demo \
  --peer-ip-address 169.254.1.1 \
  --peer-asn 65001 \
  --region us-central1

- gcloud compute networks update vpc-demo --bgp-routing-mode GLOBAL(for advertisiing across regions for connectivity since ha vpn is regional

CLOUD CDN

Tuesday, October 25, 2022 8:50 AM

Cloud CDN's content can be sourced from various types of backends:

- Instance groups
- Zonal NEG
- Serverless NEG
- Interet NEG
- Buckets

Cache hit means that GFE(google front end) will serve the content from the cache when the requested by user
Partial hit means GFE serves both from the backend and the cache
Cache miss means GFE cant be able to serve the content from the cache
Cache egress is the traffic going from the cache to client and cache fill is the traffic from the backend to cache
Cache hit ratio is the percenttage of times the content is served from the cache

Cache invalidation is used to remove the cache content
As with large scale caches content can be evicted unpredictebly so no particular request is guranteed to be served from the cache
Cloud armor can be incurred to cdn caches for edge security and backend server

We have fine grained control over folowing:

- Custom cache keys(protocol,host,query string params)
- Include or exclude specific query string params
- Set and override client and cdn ttl's at the edge
- Programmatic cache invalidation
- Negative caching
- Per origin cache policies

HTTP/2 to clients and origins, HTTP/3 (based on IETF QUIC)(should be enabled for better performance), TLS version 1.3, TCP BBR, global anycast(ipv4,v6)

Signed request(singned url's, signed cookies) for content authentication

Compliance with HIPAA,PCI-DSS,SOC(1,2,3),ISO *, FEDRAMP

Best practices:

- Use --cache-mode=CACHE_ALL_STATIC for allowing cdn to automatically cache static content coming from the origin, USE_ORIGIN_HEADERS(requires the origin to set valid caching headers to cache content),FORCE_CACHE_ALL(caches all content)
- Don’t cache user specific content
- By default cdn uses the complete request url (protocol://host/path/to/object?querystring) to build the cache key so we can customize the key to improve performace(like ignoring any like protocol/host that has the same cachable content)
- Negative caching provides fine-grained control over caching for common errors or redirects. When Cloud CDN encounters specific response codes, it holds that response in cache for a set TTL. This can reduce the load on your origins and improve the end-user experience by reducing response latency.
- Choose correct ttl's and expiration times
- Use cache control header to control expiration fro gcs objects (Cache-Control: public,max-age=252900)
- Add versioning to update cache content

- Ensure logging is enabled for cdn

If we have a serverless backend for load balancing we don’t need to have health checks and firewall rules allowing it since we use a serverless NEG
We can use this NEG as a backend in load balacnign the app engine.functions,run

- The external backends(NEG'S)is supported only in classic load balaner

gcloud compute backend-services (create | update) BACKEND_SERVICE_NAME --cache-mode=CACHE_MODE

When updating or creating backedn services/buckets we can specify --enable-cdn for enabling cdn and

- --no-cache-key-include-protocol
- --no-cache-key-include-host
- --no-cache-key-include-query-string

These for disabling particular part of the cache key and
--cache-key-include-protocol \
--cache-key-include-host \
--cache-key-include-query-string
- These for including the string user to query and use blacklist for excluding
--cache-key-include-query-string \
--cache-key-query-string-whitelist user

We can specify ttl's for specific error codes using negative caching
Eg:
gcloud compute backend-services update BACKEND_SERVICE_NAME \
--negative-caching \
--cache-mode=CACHE_ALL_STATIC \
--default-ttl=86400 \
--negative-caching-policy='404=60,405=120'

(--no-negative-caching)
Serving stale content lets Google's global cache continue to serve content when your origin server is unreachable or is returning errors to Cloud CDN.
To enable this behavior, your backend can specify the stale-while-revalidate directive in the response Cache-Control header. Cloud CDN will then serve that content from cache (if available) for the specified number of seconds past the cache entry expiration time. Asynchronously, Cloud CDN will revalidate content with the origin.

Cloud CDN enables this on your behalf with the cdnPolicy.serveWhileStale setting

gcloud compute backend-services update my-backend-service --serve-while-stale=180s

For generating signed urls we need to add keys in the backend first
1) head -c 16 /dev/urandom | base64 | tr +/ -_ > KEY_FILE_NAME
2) gcloud compute backend-services \
add-signed-url-key BACKEND_NAME \
--key-name KEY_NAME \
--key-file KEY_FILE_NAME
3)--signed-url-cache-max-age 60 for age
If you use Cloud Storage and you have restricted who can read the objects, you must give Cloud CDN permission to read the objects by adding the Cloud CDN service account to Cloud Storage ACLs.
Cloud CDN cache fill service account is not created until you add one or more keys for the project and it is not appear in project because it is cdn's property like

gcloud compute sign-url \
"https://example.com/media/video.mp4" \
--key-name my-test-key \
--expires-in 30m \
--key-file sign-url-key-file

In the code we write we should generate url's like below
https: //example.com/foo?Expires=EXPIRATION&KeyName=KEY_NAME&Signature=SIGNATURE

- Use security headers like **Strict-Transport-Security (HSTS)** -to connect directly to https instaead of http to https ,**X-Frame-Options** - Indicate whether a browser can render a page in a <frame>, <iframe>, <embed>, or <object>. This helps to prevent click-jacking attacks, by not letting your content be embedded into other sites., **Content-Security-Policy -** Don't allow in-line scripts, and only load scripts over HTTPS: Content-Security-Policy: default-src https:
For redirecting the http traffic to https we have to create the http balancer and using url map redirect forwards to https load balaner both will have same IP's

gcloud beta compute backend-services update BACKEND_SERVICE_NAME \
--compression-mode=AUTOMATIC (compression content compresses around 80% most of the cases)

Origin server supports byte range requests	Origin server does not support byte range requests
5 TB (5,497,558,138,880 bytes)	10 MB (10,485,760 bytes)

If gcs have the Content-Encoding: gzip metadata then client request should have Accept- Encoding: gzip header for byte range requests larger than 10 mb

Cache bypass allows requests containing specific request headers to bypass the cache, even if the content was previously cached.
gcloud compute backend-buckets (create | update) BACKEND_BUCKET_NAME
--bypass-cache-on-request-headers=BYPASS_REQUEST_HEADER
gcloud compute backend-services (create | update) (BACKEND_SERVICE_NAME | BACKEND_BUCKET_NAME)
--no-bypass-cache-on-request-headers

CLOUD IDS

Tuesday, October 25, 2022 1:16 PM

Cloud IDS is an intrusion detection service that provides threat detection for intrusions, malware, spyware, and command-and-control attacks on your network. Cloud IDS works by creating a Google-managed peered network with mirrored VMs. Traffic in the peered network is mirrored, and then inspected by Palo Alto Networks threat protection technologies to provide advanced threat detection. You can mirror all traffic or you can mirror filtered traffic, based on protocol, IP address range, or ingress and egress.
While Cloud IDS includes all the functionality that helps you maintain compliance, Cloud IDS itself is still being audited and is not yet compliance certified

Cloud IDS detects and alerts on threats, but does not take action to prevent attacks or repair damage You can use products like Google Cloud Armor to take action on the threats that Cloud IDS detects.

Cloud ids uses IDS endpoints that run on a zone(any zone In a region) to inspect traffic for malware or other atacks
Each IDS endpoint receives mirrored traffic and performs threat detection analysis. Cloud IDS uses Google Cloud packet mirroring, which creates a copy of your network traffic. After creating an IDS endpoint, you must attach one or more packet mirroring policies to it. These policies send mirrored traffic to a single IDS endpoint for inspection.

During IDS endpoint creation, you must choose an alert severity level. For maximum visibility, we recommend the **informational** level.others include Critical,High,Medium,Low
Each IDS endpoint has a maximum inspection capacity of 5 Gbps. We recommend that you calculate the throughput of your VPC network and ensure that you create enough IDS endpoints to service all of your traffic

1) First we peer our vpc network with the servicenetworking.googleapis.com
2) Create multiple IDS endpoints per region where workloads are there
   gcloud ids endpoints create ENDPOINT_NAME\
   --network=VPC_NETWORK\
   --zone=ZONE\
   --severity=SEVERITY\
   [--no-async] \
   [GCLOUD_WIDE_FLAG...]
3) Attach a packet mirroring policy to the endpoint
4) Obtain the endpoint forwarding rule from the ids endpoint by gcloud ids endpoints describe ENDPOINT_NAME
   gcloud compute packet-mirrorings create POLICY_NAME\
   --region=REGION--collector-ilb=ENDPOINT_FORWARDING_RULE\
   --network=VPC_NETWORK--mirrored-subnets=SUBNET

   Ids.googleapis.com/Endpoint
   Logging:
   Threat logs
   You can view logs generated due to threats in your network in Cloud Logging. The logs use a JSON format with the following fields.

   threat_id - Unique Palo Alto Networks threat identifier.
   name - Threat name.
   alert_severity - Severity of the threat. One of "INFORMATIONAL", "LOW", "MEDIUM", "HIGH" or "CRITICAL".
   type - Type of the threat.
   category - Sub-type of the threat.
   alert_time - Time when the threat was discovered.
   network - Customer network in which the threat was discovered.
   source_ip_address - Suspected traffic's source IP Address.
   destination_ip_address - Suspected traffic's destination IP Address.
   source_port - Suspected traffic's source port.
   destination_port - Suspected traffic's destination port.
   ip_protocol - Suspected traffic's IP protocol.
   application - Suspected traffic's application type (e.g. SSH).
   direction - Suspected traffic's direction (client-to-server or server-to-client).
   session_id - An internal numerical identifier applied to each session.

repeat_count - Number of sessions with the same source IP, destination IP, application, and type seen within 5 seconds.
uri_or_filename - URI or filename of the relevant threat, if applicable.
details - Additional information on the type of threat, taken from Palo Alto Networks' ThreatVault.

-

Cloud VPC

Tuesday, October 25, 2022 1:53 PM

Subnets can be used for different purposes:

- PRIVATE: A subnet to use for VM instances. This is the default subnet type.
- PRIVATE_SERVICE_CONNECT: A subnet to use to publish a managed service by using Private Service Connect.
- REGIONAL_MANAGED_PROXY: A proxy-only subnet to use with regional Envoy-based load balancers.

In most cases, you cannot change the purpose of a subnet after it has been created

When creating a firewall policies we specify the collection of firewall rules where we can specify a gotonext action that goes to the next lower level of firewall rule and making it a flow and at last assign it to one or more networks

Hierarchical firewall policy rules are defined in a firewall policy resource that acts as a container for firewall rules. The rules defined in a firewall policy are not enforced until the policy is associated with a node
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.

With **Private Service Connect**, you can create private endpoints using global internal IP addresses within your VPC network. You can assign DNS names to these internal IP addresses with meaningful names like storage-vialink1.p.googleapis.com and bigtable-adsteam.p.googleapis.com. These names and IP addresses are internal to your VPC network and any on-premises networks that are connected to it using Cloud VPN tunnels or VLAN attachments. You can control which traffic goes to which endpoint, and can demonstrate that traffic stays within Google Cloud.

This option gives you access to all Google APIs and services that are included in the API bundles.

Private Service Connect

Private Service Connect allows private consumption of services across VPC networks that belong to different groups, teams, projects, or organizations. You can publish and consume services using IP addresses that you define and that are internal to your VPC network.

You can use Private Service Connect to access Google APIs and services, or managed services in another VPC network.

Use Private Service Connect to access Google APIs
By default, if you have an application that uses a Google service, such as Cloud Storage, your application connects to the default DNS name for that service, such as storage.googleapis.com. Even though the IP addresses for the default DNS names are publicly routable, traffic sent from Google Cloud resources remains within Google's network.

With Private Service Connect, you can create private endpoints using global internal IP addresses within your VPC network. You can assign DNS names to these internal IP addresses with meaningful names like storage-vialink1.p.googleapis.com and bigtable-adsteam.p.googleapis.com. These names and IP addresses are internal to your VPC network and any on-premises networks that are connected to it using Cloud VPN tunnels or VLAN attachments. You can control which traffic goes to which endpoint, and can demonstrate that

traffic stays within Google Cloud.

This option gives you access to all Google APIs and services that are included in the API bundles. If you need to restrict access to only certain APIs and services, Private Service Connect with consumer HTTP(S) service controls allows you to choose which APIs and services are made available, for supported regional service endpoints.

For more information about Private Service Connect configurations for accessing Google APIs, see use cases.

Private Service Connect lets you send
traffic to Google APIs using a Private Service Connect
endpoint that is private to your VPC network.

Use Private Service Connect to access Google APIs with consumer HTTP(S) service controls
You can create a Private Service Connect endpoint with consumer HTTP(S) service controls using an internal HTTP(S) load balancer. The internal HTTP(S) load balancer provides the following features:

You can choose which services are available using a URL map; filtering by path lets you do more fine-grained checks.
You can rename services, for example spanner.example.com, and map them to URLs of your choice.
You can configure the load balancer to log all requests to Cloud Logging.
You can use customer-managed TLS certificates.
You can enable data residency in-transit by connecting to regional endpoints for Google APIs from workloads in that same region.

Private Service Connect lets you send
traffic to supported regional Google APIs using a
Private Service Connect endpoint. Using a load balancer adds
consumer HTTP(S) service controls.
Use Private Service Connect to publish and consume managed services
Private Service Connect lets a service producer offer services to a service consumer. A service producer VPC network can support multiple service consumers.

There are two types of Private Service Connect endpoints that can connect to a published service:

Private Service Connect endpoint (based on a forwarding rule)

With this endpoint type, consumers connect to an internal IP address that they define. Private Service Connect performs network address translation (NAT) to route the request to the service producer.

A Private Service Connect endpoint based on a forwarding
rule lets service consumers send traffic from the consumer's VPC
network to services in the service producer's VPC network.
Private Service Connect endpoint with consumer HTTP(S) servicecontrols (based on a global external HTTP(S) load balancer)

With this endpoint type, consumers connect to an external IP address. Private Service Connect uses a network endpoint group to route the request to the service producer.
Using a global external HTTP(S) load balancer as a policy enforcement point has the following benefits:
You can rename services and map them to URLs of your choice.
You can configure the load balancer to log all requests to Cloud Logging.

You can use customer-managed TLS certificates. or Google-managed certificates.
If the service producer has made a service available in multiple regions, client traffic can be load balanced across those regions.
Using a global external HTTP(S) load balancer lets service consumers with internet access send traffic to services in the service producer's VPC network.
Figure 4. Using a global external HTTP(S) load balancer lets service consumers with internet access send traffic to services in the service producer's VPC network (click to enlarge).

Managed services in multiple regions
You can make a service available in multiple regions by creating the following configurations.

Producer configuration:

Deploy the service in each region. Each regional instance of the service must be configured on a load balancer that supports access by a Private Service Connect endpoint with consumer HTTP(S) service controls.

Create a service attachment to publish each regional instance of the service.

Consumer configuration:
Create a Private Service Connect endpoint with consumer HTTP(S) service controls. The endpoint is based on a global external HTTP(S) load balancer and includes the following configurations:
A Private Service Connect NEG in each region that points to that region's service attachment.
A backend service that contains the NEG backends.
Endpoint type Supported targets Accessible by
Private Service Connect endpoint to access Google APIs
global internal IP address

An API bundle:
All APIs (all-apis): most Google APIs
(same as private.googleapis.com).
VPC-SC (vpc-sc): APIs that VPC Service Controls supports
(same as restricted.googleapis.com

Private google access for on prem hosts:

- The on-premises DNS configuration maps *.googleapis.com requests to restricted.googleapis.com, which resolves to the 199.36.153.4/30.
- Cloud Router has been configured to advertise the 199.36.153.4/30 IP address range through the Cloud VPN tunnel by using a custom route advertisement. Traffic going to Google APIs is routed through the tunnel to the VPC network.
- A custom static route was added to the VPC network that directs traffic with the destination 199.36.153.4/30 to the default internet gateway (as the next hop). Google then routes traffic to the appropriate API or service.
- If you created a Cloud DNS managed private zone for *.googleapis.com that maps to 199.36.153.4/30 and have authorized that zone for use by your VPC network, requests to anything in the googleapis.com domain are sent to the IP addresses that are used by restricted.googleapis.com. Only the supported APIs are accessible with this configuration, which might cause other services to be unreachable. Cloud DNS doesn't support partial overrides

Private services connect and private services access means it uses the producer consumer endpoint concepts

The service producer will create a producer service attachment and service consumer will create a service endpoint and connect to the producer

The producer must publish a service(load balancer) which enables the consumer to connect privately

Peering to the service networking .googleapis.com through vpc peering is called private services access
In this connection private only particular services are available to connect privately through this approach

CLOUD DNS

Tuesday, October 25, 2022 8:09 PM

- **A:** Address record, which maps host names to their IPv4 address.
- **AAAA:** IPv6 Address record, which maps host names to their IPv6 address.
- **CNAME:** Canonical name record, which specifies alias names.
- **MX:** Mail exchange record, which is used in routing requests to mail servers.
- **NS:** Name server record, which delegates a DNS zone to an authoritative server.
- **PTR:** Pointer record, which defines a name associated with an IP address.
- **SOA:** Start of authority, used to designate the primary name server and administrator responsible for a zone. Each zone hosted on a DNS server must have an SOA (start of authority) record. You can modify the record as needed (for example, you can change the serial number to an arbitrary number to support date-based versioning).

First we need to register a domain using google domains or other registrar and us cloud dns to create a zone and get the nameserver records and put them in the domain purchased

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

We can export the dns zone file from one dns provider and import them into cloud dns for use it in gcp

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

DNSSEC:

- authenticates responses to domain name lookups. It does not provide privacy protections for those lookups, but prevents attackers from manipulating or poisoning the responses to DNS requests.
- The DNS zone for your domain must serve special DNSSEC records for public keys (DNSKEY), signatures (RRSIG), and non-existence (NSEC, or NSEC3 and NSEC3PARAM) to authenticate your zone's contents

gcloud dns managed-zones update EXAMPLE_ZONE \
  --dnssec-state on

- To enable or disable logging use

gcloud dns policies create POLICY_NAME \
  --networks=NETWORK \
  --enable-logging \
  --description=DESCRIPTION
gcloud dns policies update POLICY_NAME\
  --networks=NETWORK\
  --no-enable-logging

gcloud beta container clusters create private-cluster \
  --enable-private-nodes \
  --master-ipv4-cidr 172.16.0.16/28 \
  --enable-ip-alias \
  --create-subnetwork ""

It will create automatically a subnet with secondary ip range with secondary ip range for pods,services and private google acess on

gcloud container clusters update private-cluster \
  --enable-master-authorized-networks \
  --master-authorized-networks [natIp/32 from the describe the vm instance]

Best practices of networking

Wednesday, October 26, 2022 12:46 PM

VPC-native clusters are required for private GKE clusters and for creating clusters on Shared VPCs. For clusters created in the Autopilot mode, VPC-native mode is always on and cannot be turned off.

Note: You cannot migrate between routes-based and VPC-native cluster types.

If we have used all the rfc 1918 address space we can use either RFC6598,5735 address space these are called non rfc 1918 address spaces

Install istio or define network policy as a manifest to control pod to pod connectivity security
Disable kubernetes UI dashboard, abac instead use rbac

RBAC example
- rules:
  apiGroups: ["apps","extensions"]
  resources: ["deployments"]
  verbs: ["get","list","watch"]

In routes in vpc the next hop can be an instance,ip address, forwarding rulle og lb,default internet gateway,vpn tunnel

Routes,byoip, Third-party device insertion (NGFW) into VPC using multi-NIC and internal load balancer as a next hop or equal-cost multi-path (ECMP) routes,vpc service controls,NAT,cloud identity,
2FA, access context manager, pseudonymization, format preserving substitution,
Automating security scanning for Common Vulnerabilities and Exposures (CVEs) through a CI/CD pipeline

- Automating virtual machine image creation, hardening, and maintenance

- Automating container image creation, verification, hardening, maintenance, and patch management

CLOUD ROUTER

Tuesday, November 22, 2022 2:45 PM

- https://cloud.google.com/network-connectivity/docs/router/concepts/overview#ilbs
- BGP sessions for the network connectivity products use link-local IPv4 addresses in the 169.254.0.0/16 range as BGP IP addresses
- ASN number will be used from each routers side for the bgp sessions
- Cloud Router supports only regional routing of IPv6 traffic in HA VPN tunnels. IPv6 traffic is routed within the region assigned to the HA VPN gateway. Global routing for IPv6 traffic in HA VPN tunnels is not supported
- During maintenance, new software tasks are provisioned. Your on-premises router logs indicate that BGP sessions managed by those software tasks went down and came back up (a BGP flap).
- Cloud Router maintenance is an automatic process, and it is designed so that it does not interrupt routing. Maintenance events are expected to take no more than 60 seconds. Before maintenance, the Cloud Router sends a graceful restart notification (a TCP FIN packet) to the on-premises router.
- Cloud Router advertises IP address ranges to your on-premises network. This allows clients in your on-premises network to send packets to and receive packets from resources in your VPC network
    - Default route advertisement
        - Regional dynamic routing mode allows cloud router to advertise primary/secondary rangesin same region
        - Global dynamic routing mode allows to advertise from all regions in the vpc network

      If you advertise privately used public IPv4 addresses, on-premises systems might be unable to access internet resources, which use those public IPv4 addresses.
    - Custom route advertisement
        - Only custom ipv4,ipv6
        - Custom ipv4,v6 in addition to subnet routes
    - Effective advertised prefixed
        - We can configure modes for cloud router basis or each bgp session basis
        - Like default mode for cloud router and custom mode for bgo session and vice versa also
- In order to advertise ipv6 we need to confire to use the dual stack ip addresses or the bgp session should be only enaled with ipv6

When you configure a BGP session on a Cloud Router, you can specify a base advertised priority for the BGP session. The base advertised priority applies to all prefixes (destinations) advertised by that BGP session.

Base priorities are whole numbers from 0 to 65535. The highest possible base priority is 0. The default base priority is 100. If you don't specify a base priority, the default priority is used.

Base priorities let you control which Cloud VPN tunnels or VLAN attachments on-premises systems use to send packets to your VPC network
Base priorities for BGP sessions among Cloud Routers in a region should be between 0 and 200, inclusive

- In classic vpn we cant use a peer gateway like on prem vpn or other vpn provider outside of gcp

In static routing, a network administrator uses static tables to manually configure and select network routes. Static routing is helpful in situations where the network design or parameters are expected to remain constant.

In dynamic routing, routers create and update routing tables at runtime based on actual network conditions. They attempt to find the fastest path from the source to the destination by using a dynamic routing protocol, which is a set of rules that create, maintain, and update the dynamic routing table

- First we create an vpn gateway in gcp side that has two interfaces because of ha vpn and use that one of the gateway address in the aws site to site vpn connection dialog
- In aws side create a virtual private gateway and give it a asn number of your choice that shoulod be used while bgp setup
- In aws side after creating the site site vpn connection download the file and inside it it will be given the gateway ip for each tunnel we can use those ip's to create a peer gateway with two interfaces (tunnel outside ip addresses)
- When configuring the tunnels we use the shared secret given in the file in the gcp dialog for tunnel creation in that use the interface ip from gcp (one ip addres congirured in the aws side) and in the other ip section give the tunnel ip that is given in the file

(we can use the cloud router from google side with a asn and a transit gateway from aws side with a asn and use them in a vpn connection and specify those asn's)

- We need to aslos use the bgp ip addresses to establish the connection like give the correct bgp addresses from aws cidr or gcp cidr (given in the file downloaded) (manually give the bgp ip's from the cidr's found in the aws side tunnels inside ip's)
- Add the cidr of the gcp network into the aws route table to test connection(set as main route table if creating another route table for the vpc)
- Create an internet gateway seperately for the aws side to attach it to vpc
- And also configure necessary security group in aws and firewall rules in gcp side for allowing/denying internal ip addresses of both(like rule for allowing the aws subnet range in gcp,same in aws side to allow gcp subnet ip)
  (for ping configure security group for allow custom ICP and give the gcp sibnet cidr)

gcloud compute config-ssh --ssh-key-file=~/.ssh/vm-ssh-key(for registering the ssh key with gcp)

The asn number should be (64512-65534, 4200000000-4294967294)

- By default there is not authentication for bgp sessions if we want to authenticate we can use md5 authentication where both sides need to use the same authentication key
- **Keepalive timer** - indicates whether a router is reachable to bgp peer
- **Hold timer** - keepalivetimer *3 (length of time that a cloud router or your onprem router should wait
- **Graceful restart timer -** timer values that defines the amount of time that the other router should wait after receiviing a graceful restart notification by the router
- **Stalepath timer -** how long a router waits before deleting learned routes after receving the EOR(endofrecord)(usually after the reinitialization after graceful restart) message from other router

We can use MD5 authentication only in HA VPN dedicated interconnect,partner interconnect
We provide the shared secret key while configuring cloud router or peer router

BFD (RFC 5880, RFC 5881) is a forwarding path outage detection protocol that is supported by most commercial routers. With BFD for Cloud Router, you can enable BFD functionality inside a BGP session to detect forwarding path outages such as link down events. This capability makes hybrid networks more resilient.

When you peer with Google Cloud from your on-premises network by using Dedicated Interconnect or Partner Interconnect, you can enable BFD for fast detection of link failure and

failover of traffic to an alternate link that has a backup BGP session. In this way, BFD provides a high-availability network connectivity path that can respond quickly to link faults.

BFD configured with default settings detects failure in 5 seconds, compared to 60 seconds for BGP-based failure detection. With BFD implemented on Cloud Router, end-to-end detection time can be as short as 5 seconds.*UDP based protocol)

Not supported for ha vpn only for dedicated,partner interconnect

BFD can only be enabled on provisioned VLAN attachments that use Dataplane version 2.
BFD MULITPLIER=>The number of consecutive BFD control packets that must be missed before BFD declares that a peer is unavailable.

There are three BFD mode settings, Active, Passive, and Disabled. If you don't set this mode, it defaults to a setting of Disabled, using non-echo mode (control packets only).

Disabled (default): BFD is disabled for this BGP peer.
Passive: The Cloud Router waits for the peer router to initiate the BFD session for this BGP peer.
Active: The Cloud Router initiates the BFD session for this BGP peer.
You must set the router on at least one side of a connection—either the Cloud Router or the peer router—to Active. When configuring a BGP session between two Cloud Routers, set one router's BFD session initialization mode to Active.

A router appliance instance uses its internal IP address to peer with the Cloud Router. The instance does not use a link-local address (such as 169.254.x.x) for BGP peering.

Similarly, the Cloud Router uses two RFC 1918 internal IP addresses to peer with the router appliance instance, one address for each interface. These IP addresses can be manually or automatically assigned from the subnet that contains both the router appliance instance interface and the Cloud Router interface.

If we use network connectivity center we have to create a hub and attach spokes accordingly (if we use router appliance vm instances we needd to add them via adding spokes)

Other networking products

Thursday, November 24, 2022 2:10 PM