Google Cloud

Considerations for NGFW Appliances

Vijay Sekhri

Strategic Cloud Engineer

September 10, 2022

Introduction

Google Cloud offers multiple security solutions for many types of customer's use cases. To adopt defense in depth strategy, GCP has a plethora of managed services to choose from pertaining to different layers of the infrastructure stack. GCP offers native firewall capability without the need to provision any infrastructure components. However, many customers choose to supplement or replace GCP's native firewall with a third party next generation firewall (NGFW) using appliances. Many customers who are in the financial sector or the public sector choose NGFW for compliance reasons and others choose it for familiarity of the toolset. In this document we will describe the set of considerations to take into account when incorporating third party firewalls. This document is mainly meant for Security and Network operators who would also learn a few pros and cons of using those appliances.

Load Balancing

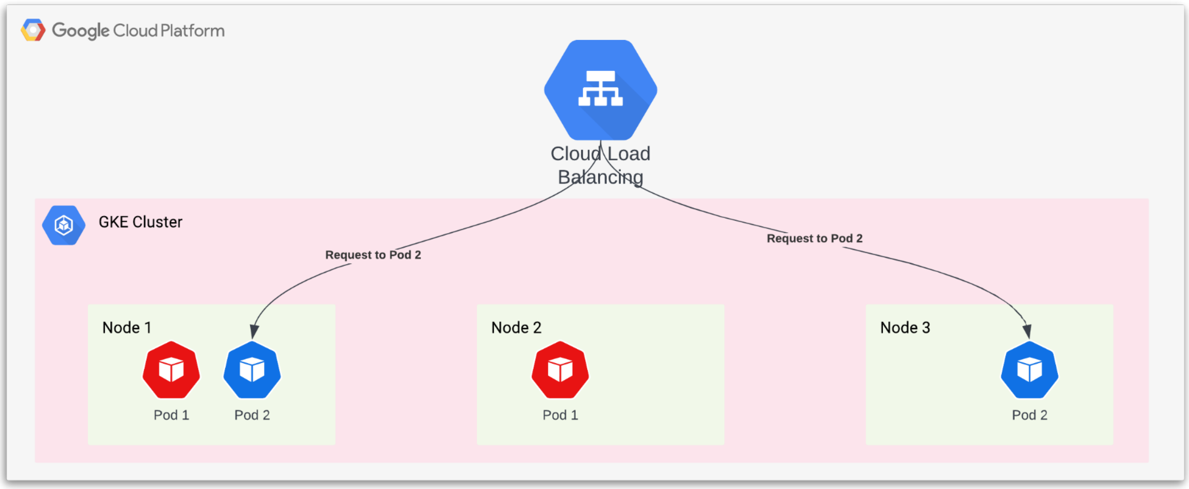
To control public ingress traffic, customers often add NGFW appliances between the public load balancer and the backend. To ensure HA of such design typically the NGFW is set up behind its own managed/unmanaged instance group. In this case the traffic flows from Google public load balancer to NGFW and from NGFW to the backends.

GCP offers fully managed HTTP and TCP load balancing services for both internal and external facing use cases. With respect to Load Balancing with Kubernetes/GKE, GCP also offers additional capabilities like

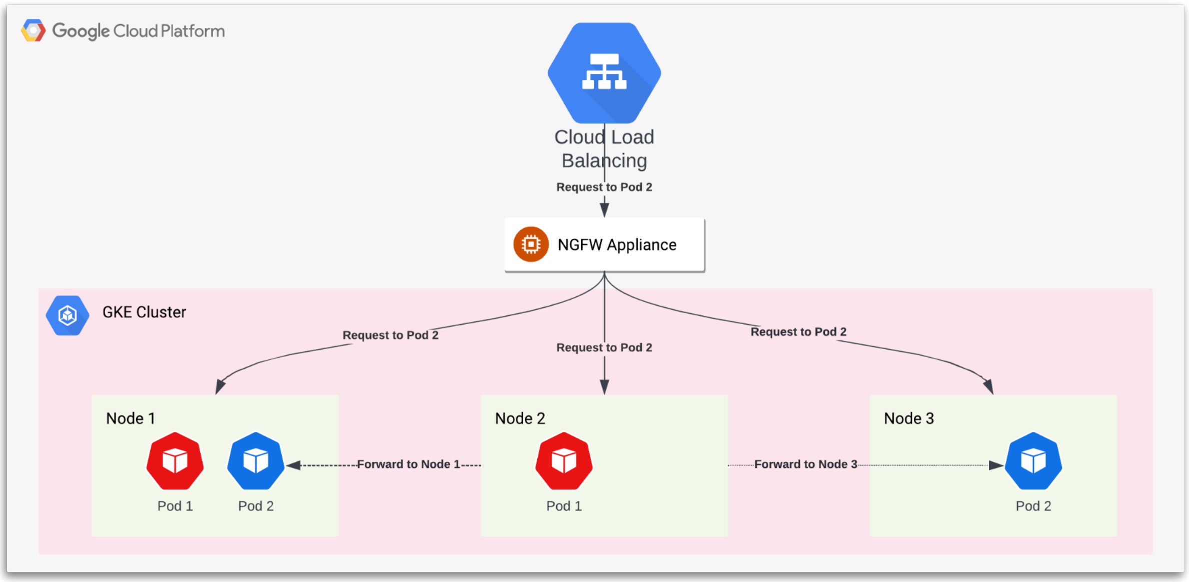
- Container native load balancing
- Kubernetes gateway controller
- Multi cluster ingress

Container native load balancing

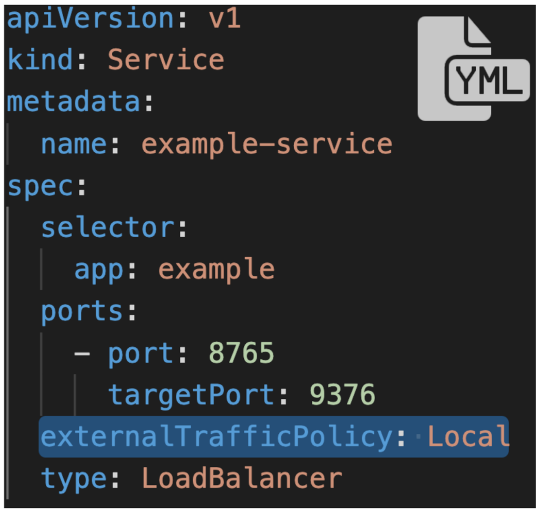
Container native load balancing on GKE helps avoid the double hop problem. The Load Balancer service can route traffic directly to the node where the containers are present, instead of any node in the cluster that would forward the message again to another node where the container would be present.



When a NGFW appliance is inserted in between the traffic from Cloud Load Balancer to the backend nodes, the former loses the knowledge of the pods. Hence the only endpoint the Load Balancer can forward the traffic to is of NGFW appliances. In this case, the capability of the GCLB to avoid unnecessary hops to other nodes to reach other containers is lost.



When traffic is sent from NGFW to the nodes, it is typically sent to the ILB using a Kubernetes service object that forwards it to the nodes. In this scenario, to avoid the double hop problem, the externalTrafficPolicy could be set to Local. Notice that the packets will still have to pass through the NGFW and then directly to the backend pods. However the packets will not hop again once reaching the node of the custer.

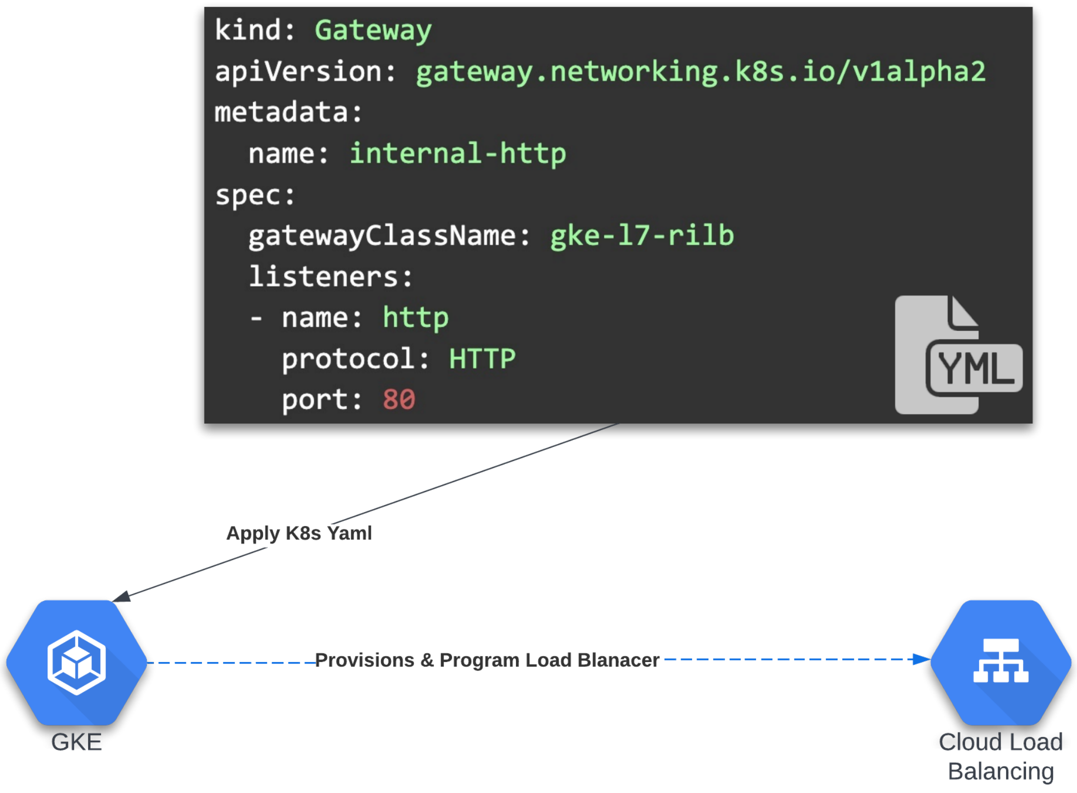


```yaml
apiVersion: v1
kind: Service
metadata:
  name: example-service
spec:
  selector:
    app: example
  ports:
    - port: 8765
      targetPort: 9376
  externalTrafficPolicy: Local
  type: LoadBalancer
```

Kubernetes Gateway Controller

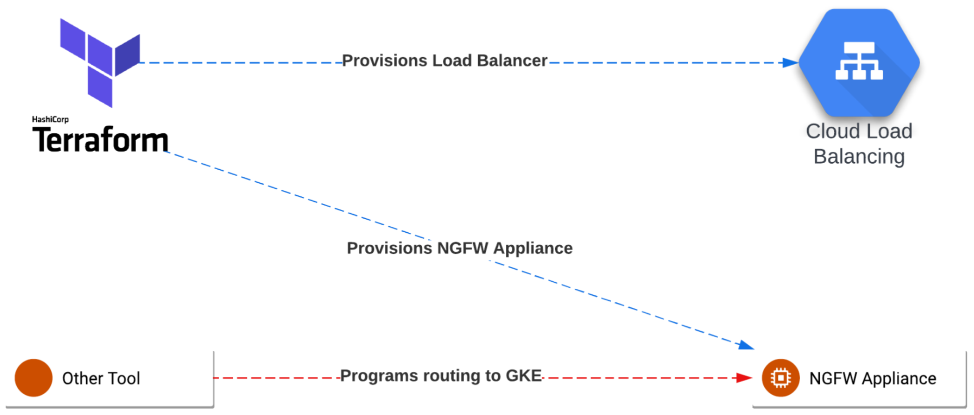
Kubernetes gateway controller is the next evolution of native Kubernetes ingress objects with features like traffic shaping, multi-tenant role oriented design, etc. Provisioning a Global Google Cloud Load Balancer is a multi-step process and proprietary to GCP. For promoting infrastructure portability across clouds, Kubernetes provides native constructs like Kubernetes gateway controller that can be declared once and deployed anywhere. Using simple Kubernetes objects as shown below, the controllers provision and configure the Load Balancer of the cloud providers.

This is possible because the Load Balancer of the cloud provider directly routes the traffic to the nodes/pods based on the Kubernetes native object declaration.



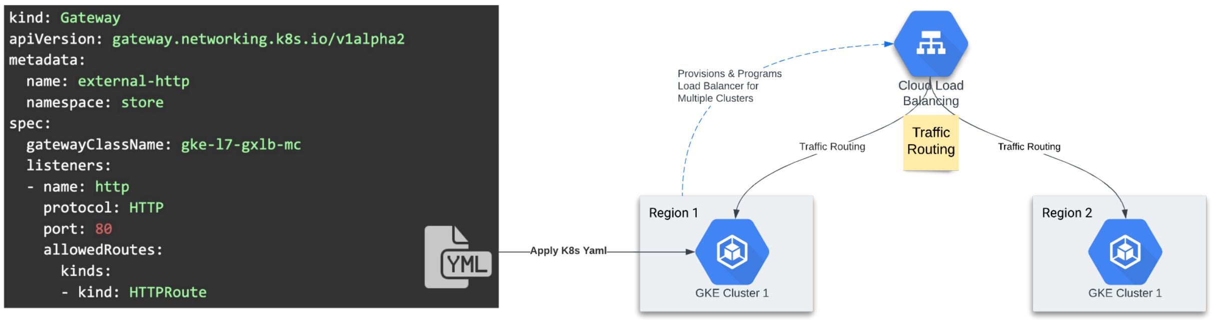
```yaml
kind: Gateway
apiVersion: gateway.networking.k8s.io/v1alpha2
metadata:
  name: internal-http
spec:
  gatewayClassName: gke-l7-rilb
  listeners:
  - name: http
    protocol: HTTP
    port: 80
```

When a NGFW appliance is inserted between the Load Balancer and the nodes, then the customers cannot simply use the native Kubernetes objects to program the load balancer. As the traffic from external Load Balancer has to always reach NGFW appliances and the appliances have to later route the traffic to appropriate nodes (or ILBs provisioned with Kubernetes service), multiple non-kubernetes-native tools are required to set up the flow correctly. One such way would be to first use Terraform or Ansible to provision the Load Balancer and the NGFW appliance and then use another tool to automate the routing for traffic from the appliance to nodes (or ILBs) of the GKE cluster. In some cases VM startup scripts can be used while provisioning NGFW appliances to automate the routing instead of using some other tool altogether. Nonetheless, the customers will have to absorb some additional operational complexity in this architecture.



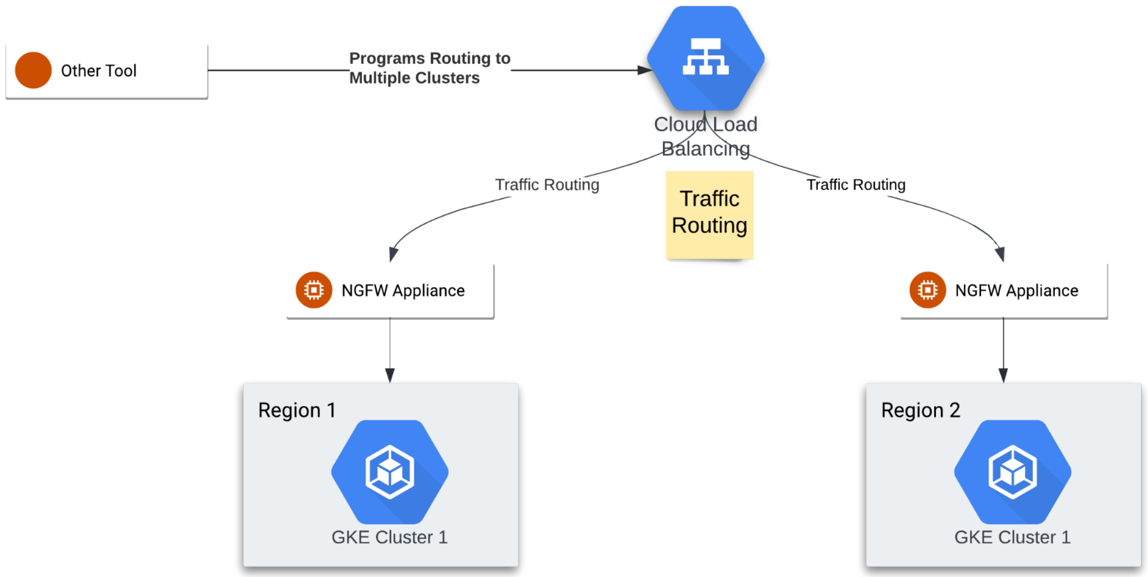
Multi Cluster Ingress/Gateway

GKE clusters should be deployed in multiple regions, in order to improve systems availability (automatic HA/DR). To manage north-south traffic to these clusters, [Multi Cluster Ingress](#) (MCI) objects offer a native Kubernetes solution. Without MCI, customers will have to manage routing of traffic from the load balancers to these clusters by manually programming the Load Balancers. As the number of workloads and clusters grow, this task of configuring Load Balancer becomes management overhead. With MCI a single declarative Kubernetes object can easily provision and program Google Load Balancer across multiple clusters.



```
kind: Gateway
apiVersion: gateway.networking.k8s.io/v1alpha2
metadata:
  name: external-http
  namespace: store
spec:
  gatewayClassName: gke-l7-gxlb-mc
  listeners:
  - name: http
    protocol: HTTP
    port: 80
    allowedRoutes:
      kinds:
      - kind: HTTPRoute
```

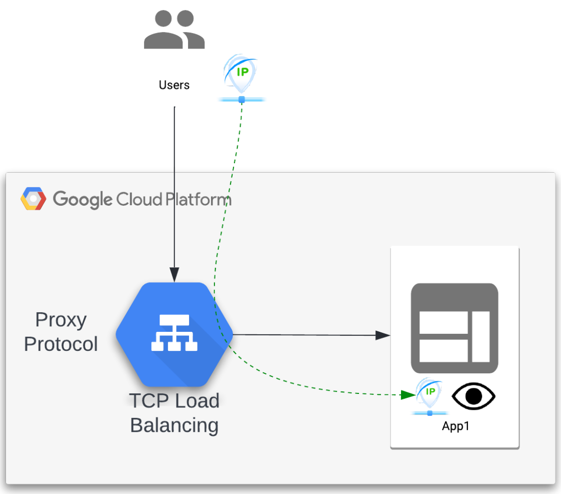
When an NGFW appliance is inserted between the traffic from the Load Balancer and the GKE clusters, routing from the Load Balancer has to be manually programmed by using some tool like [ansible](#) or [terraform](#). The appliances have to be deployed in multiple regions as Load Balancer backends. Finally, the routing still has to be configured in each backend appliance to reach the appropriate GKE cluster nodes or ILBs provisioned using [Kubernetes Services](#) .



Preserving Client IP

Proxy Load Balancers terminate the connection and establish a new one to the backend. In doing so, these load balancers would lose the client source IP address. When Layer7 load balancers are used, client source IP can be easily preserved by leveraging the x-forwarded-for http headers. With Layer4 load balancers, the client source IP can be preserved in 2 ways:

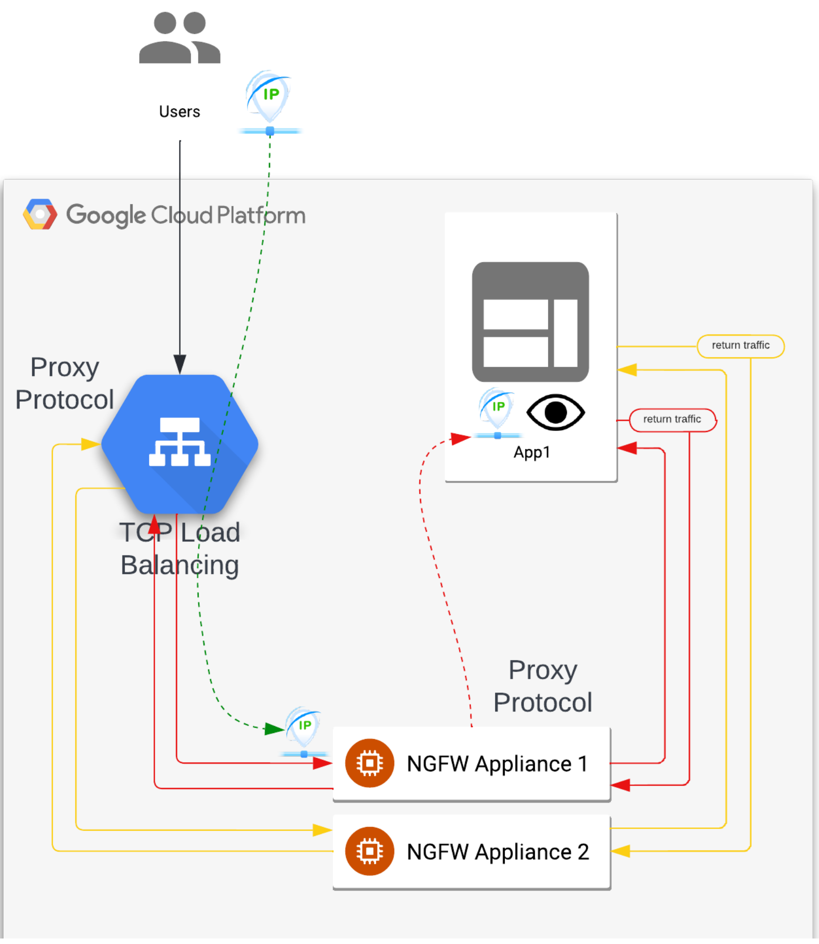
- By simply using Network Load Balancers (Non Proxy)
- By using proxy protocol and TCP Proxy / SSL Proxy Load Balancers



When NGFW appliances are used in HA setup, to ensure that the round trip traffic can be returned back correctly, it has to flow from the same NGFW appliance instance. To maintain this affinity, one of the following two options should be used:

1. Symmetric Hashing with ILBs
2. Source NAT

With Symmetric Hashing, the client source IP can be left unchanged by the NGFW and return traffic from the backend should route to the same NGFW. However, if customers choose to use Source NATing, Proxy Protocol has to be supported and enabled on the NGFW to retain the client source IP.

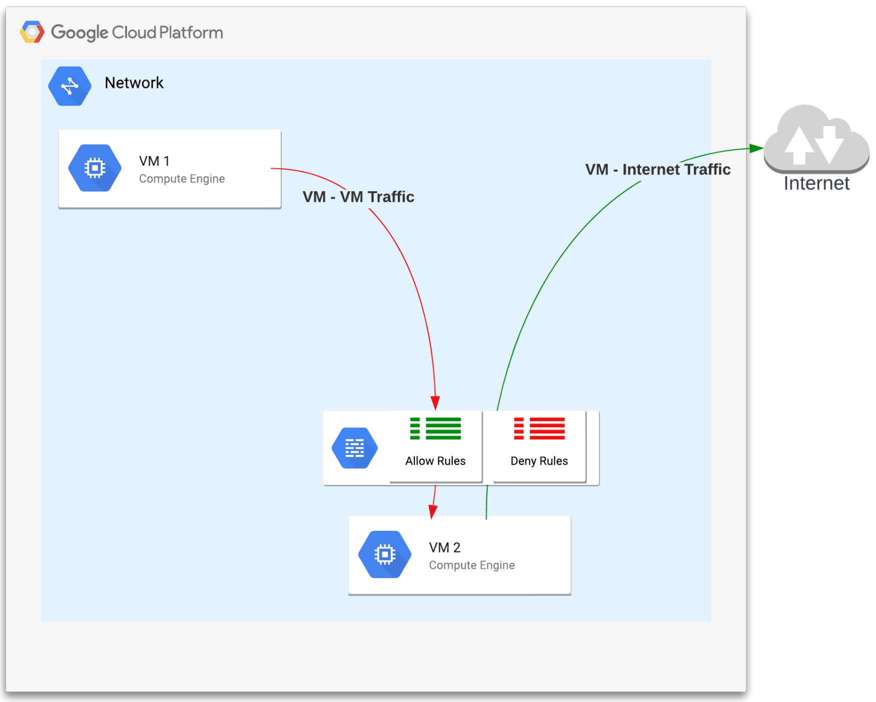


Performance

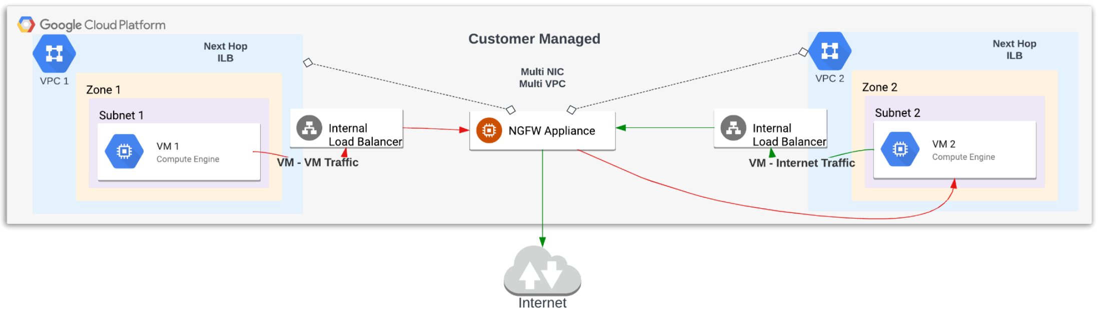
GCP native firewalls like other network functions are software defined and are distributed in nature. Firewall rules are defined at the underlying network level and are evaluated on each compute instance for ingress and egress traffic. This type or architecture promotes high performance and eliminates single choke points in the system.

Autoscaling

As the number of infrastructure resources grows, the firewall evaluation does not suffer from bottlenecks as they are spanned across the network. Scaling of underlying networking is Google's responsibility and customers don't have to manage any resource based on network traffic spikes.



When NGFW is set up between different VPCs for VM to VM traffic and established as the default route for external traffic, then the entire management of scaling this central point becomes customer responsibility. If not done carefully, this could lead to a single choke point in the system. As all traffic gets routed through the same set of appliances, the performance of network traffic can be impacted by this bottleneck design. Using Internal Load Balancer in front of the appliances assists with managing the health of the appliances, however it may not always be simple to autoscale these appliances based on traffic. The reason for difficult autoscaling is that the image for the appliance will have to be self-contained such that all the routing decisions are a part of the appliance image or pulled from some central system during startup to function correctly. Some NGFW appliance vendors may provide an easy way for auto scaling, however it may be more difficult for others.

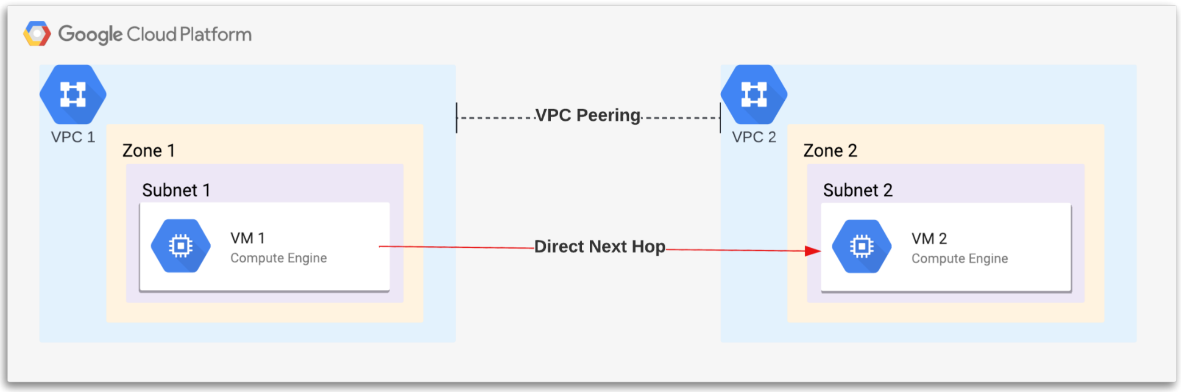


Metrics for autoscaling typically rely on CPU utilization , memory utilization or network bandwidth. To handle spikes in network traffic, enough room should be set aside, as it could take a while for the entire new instance of NGFW to be fully ready. The time for an NGFW instance to spin up may vary from vendor to vendor and therefore should be taken into consideration while configuring the resources of the instances.

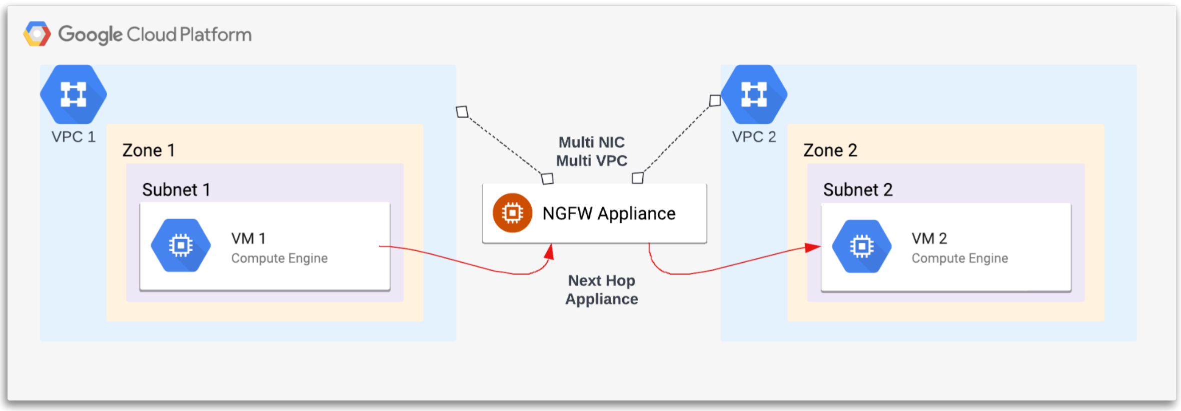
Oftentimes auto-scaling and high availability requires session failover so that any in-flight connections are not dropped. Vendor specific settings have to be configured to enable session failover. One such example for Palo Alto NGFW is to configure floating IP.

Next Hops

With native GCP firewalls, the packets leaving and entering the VM instances are evaluated at the networking level. It is fair to assume that the routing decisions and evaluation of firewall rules for the packets are done at each VM, therefore one VM can send traffic to another VM directly and still honor the firewall conditions. This implies that even in different VPCs or different subnets the communication among the services is happening with the least amount of network hops. Eliminating network hops in native GCP settings, leads to better network traffic performance.

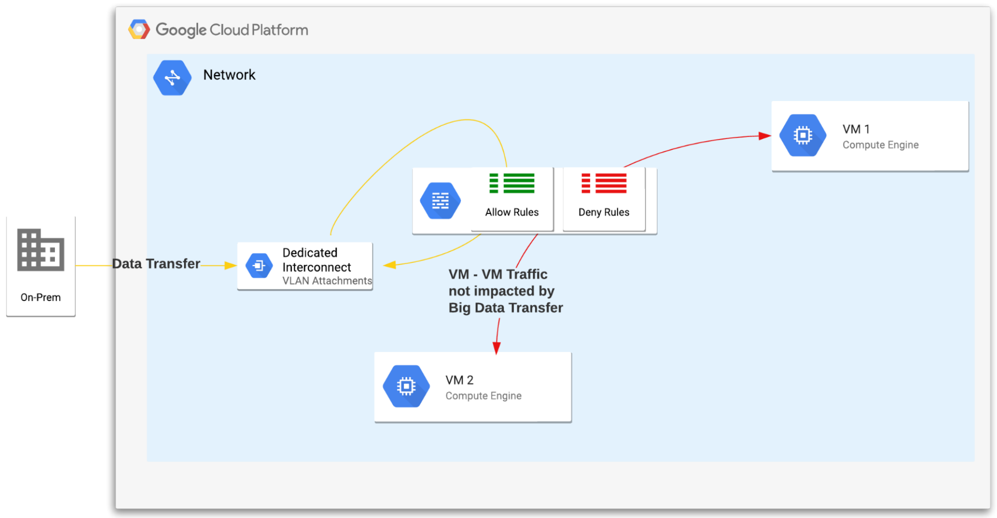


In contrast, when all traffic is required to be routed through the appliances, the next hop for the network packets are typically the forwarding rule off the ILB, an IP address or the appliance instance itself. This implies that for each packet to be delivered to its destination, it needs to traverse through another system thereby adding an additional network hop and impacting the performance to some extent.

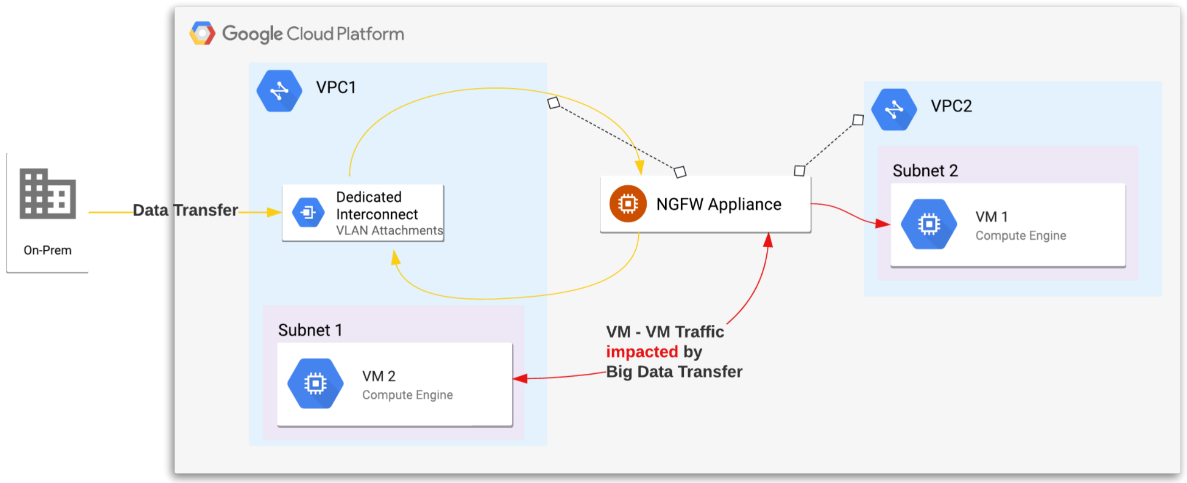


Data Transfers

Many customers have on-prem or other cloud providers from where the data of a certain system needs to be transferred to and from GCP on a regular cadence. Sometimes the size of data can vary from a few TBs to a few PBs. Using native GCP firewalls for on-prem to GCP data transfer, throughput is limited by the connection options to GCP data centers and the flavor of the compute instances. The high amount of data transfer does not impact other workloads/services running on GCP, even though they all share the same native firewall. This is due to the distributed nature of GCP native firewall which is fully managed and auto scaled by Google.



When NGFW appliances handle both on-prem to cloud traffic and east west (VPC to VPC) traffic, the workloads may show a negative performance impact during the data transfer. This is because the same appliance, responsible for transferring large amounts of data, is also used for VM to VM communication. Ideally, the data transfer from and to on-prem data center should not impact the workloads/service already running on GCP. One way to avoid such scenarios is to separate out a set of appliances for on-prem traffic , east-west traffic and north-south traffic. Isolating and dedicating appliances for each type of traffic is desired but may be more costly depending on the license agreements with appliance vendors.

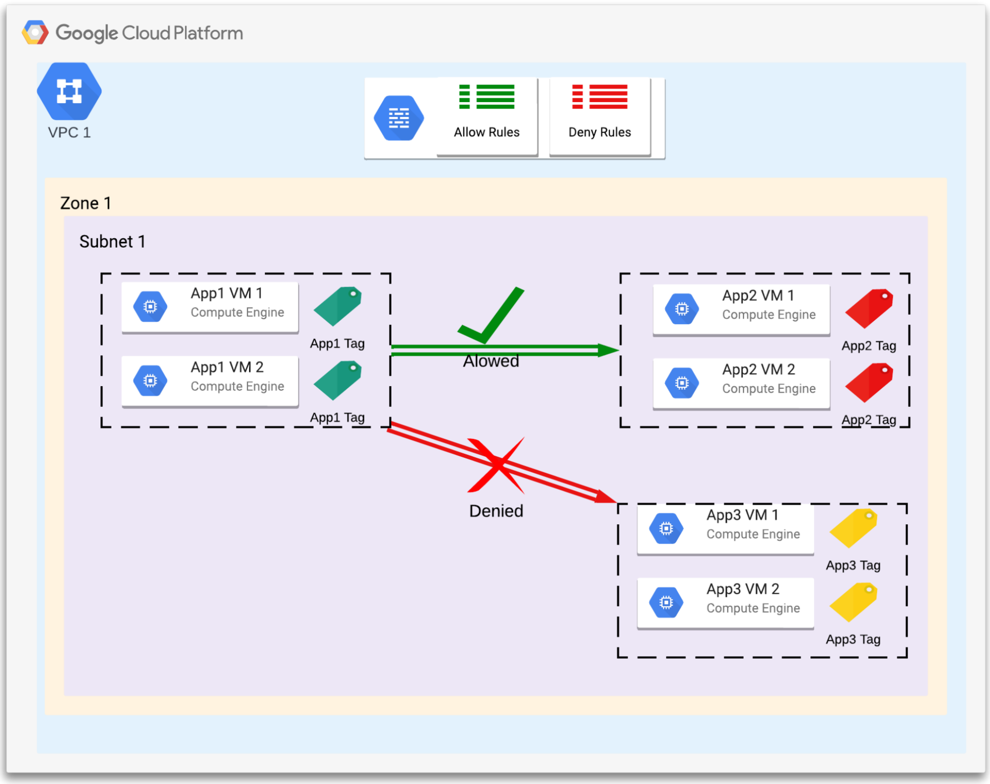

Google Cloud Platform
VPC1
VPC2
On-Prem
Data Transfer
Dedicated Interconnect
VLAN Attachments
NGFW Appliance
Subnet 2
VM 1
Compute Engine
Subnet 1
VM 2
Compute Engine
VM - VM Traffic
impacted by
Big Data Transfer

Micro segmentation

Google recommends creating large subnets and less number of VPCs in general. This is to avoid the toil of managing a large number of networking constructs. To adopt a service centric view, GCP also offers a Private Service Connect (PSC) option. However in many cases PSC may not be feasible and large VPC with less number of subnets and many services could be a design pattern of choice. In such scenarios customers require a way to achieve micro segmentation that can control communication among services running in the same subnets.

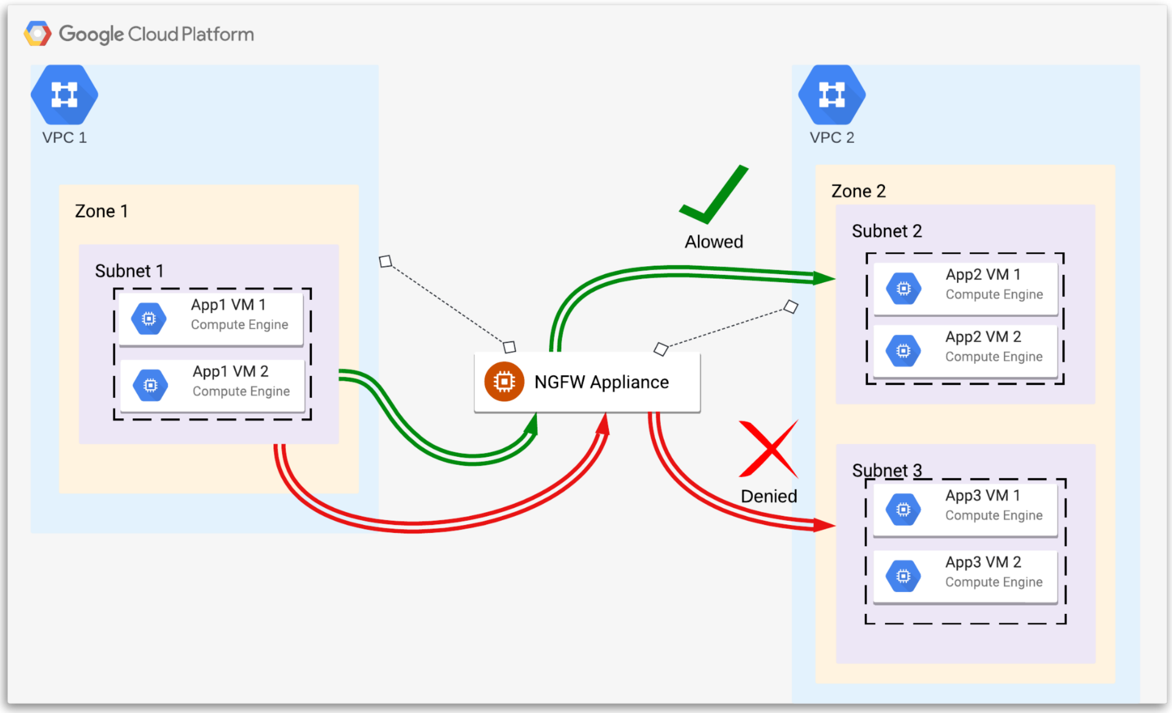
VM Based Services Micro Segmentation

GCP native firewall supports creating rules with secure tags and service accounts. Within the same subnets micro segmentation can be achieved by simply tagging different VMs with different resource manager tag values and authoring firewall rules that leverage those tag values. In the picture below, App1 VMs have tag App1, App2 VMs have tag App2 and App3 VMs have tag App3. A simple firewall rule leverages the tag App1 as source and App2 as target for allow rule and App1 as source and App3 as destination for deny rule. This lets successful connectivity between App1 VMs and App2 VMs but denies connectivity from App1 VMs to App3 VMs.



Appliances have to integrate with GCP networking so that VM tags can be leveraged, in order to provide microsegmentation. One such example would be to configure rules based on network tags in VM Series by Palo Alto. In cases when NGFW vendors do not support such integration, to achieve micro segmentation, customers have to again build smaller subnets per application. This creates the same toil again for managing multiple small subnets like in the on-prem environments.

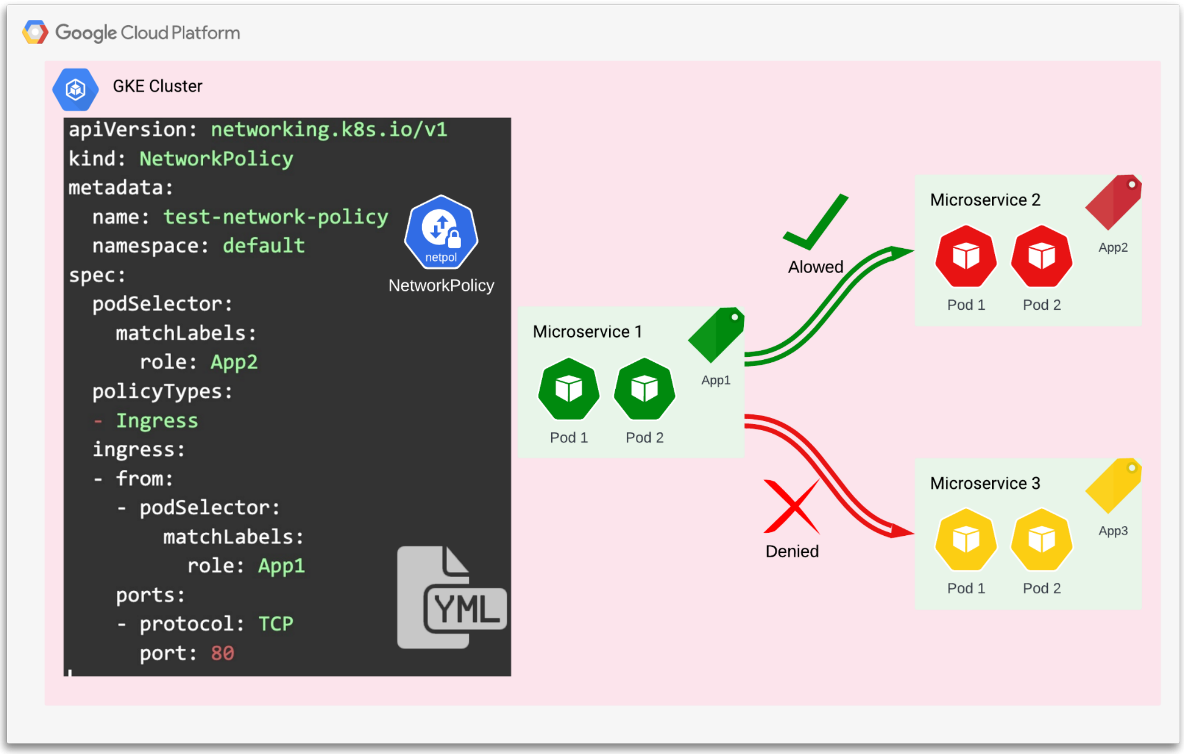
Furthermore, currently* appliances cannot be sandwiched between subnets within the same VPC. If customers want all inter-subnet traffic to be routed through NGFW appliances, then they have to be created in separate VPCs. In addition, firewall appliances can not filter or inspect traffic between hosts in the same subnet, whereas GCP Cloud Firewall rules are able to do so.



Container Based Services Micro Segmentation

When multiple services run in multi-tenant environments like GKE, microsegmentation is achieved using native Kubernetes constructs. Kubernetes network policy provides a declarative approach to achieve micro segmentation using the labels and selectors, which are similar to tags for VM based micro segmentation. GCP also supports dataplanev2 for providing more features like logging of firewall rules that uses eBPF. In the picture below , Microservice1 has a label App1, Micorservice2 has a label App2 and Microservice3 has a label App3. To allow traffic from Microservice1 to Microservice2 the network policy simply uses labels App1 and App2 .

Similarly, to disallow traffic from App1 to App3, a deny rule can be added by leveraging labels only.



```
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: test-network-policy
  namespace: default
spec:
  podSelector:
    matchLabels:
      role: App2
  policyTypes:
  - Ingress
  ingress:
  - from:
    - podSelector:
        matchLabels:
          role: App1
    ports:
    - protocol: TCP
      port: 80
```

All NGFW vendors may not always support container based micro segmentation. Typically the NGFW will run as another container inside the Kubernetes environment as a daemonset. One such example would be Container Series by Palo Alto.

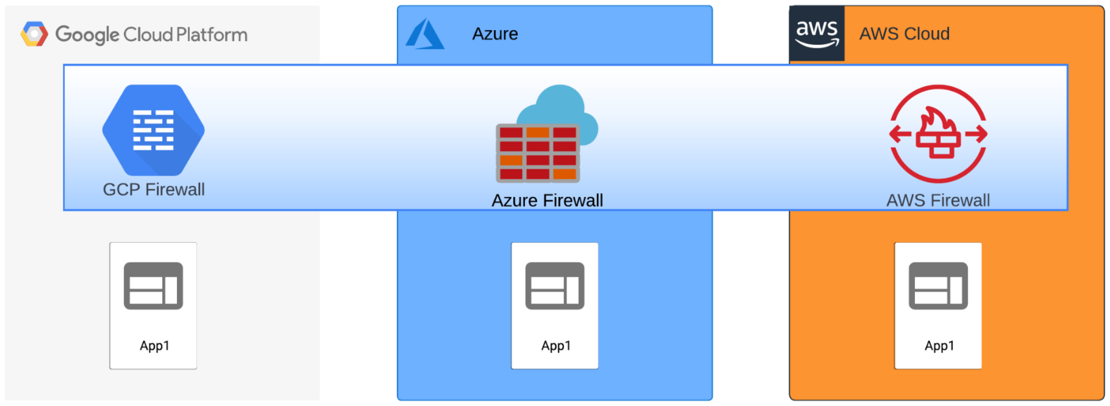
Multi Cloud

Many customers strive towards using a multi cloud strategy. To build a resilient system capable of dealing with any cloud provider outages, customers tend to follow architecture in which infrastructure gets mirrored on multiple data centers (multi cloud and on-prem). To operate with this multi cloud architecture, a set of new challenges could emerge, one of which is how to design cloud agnostic systems.

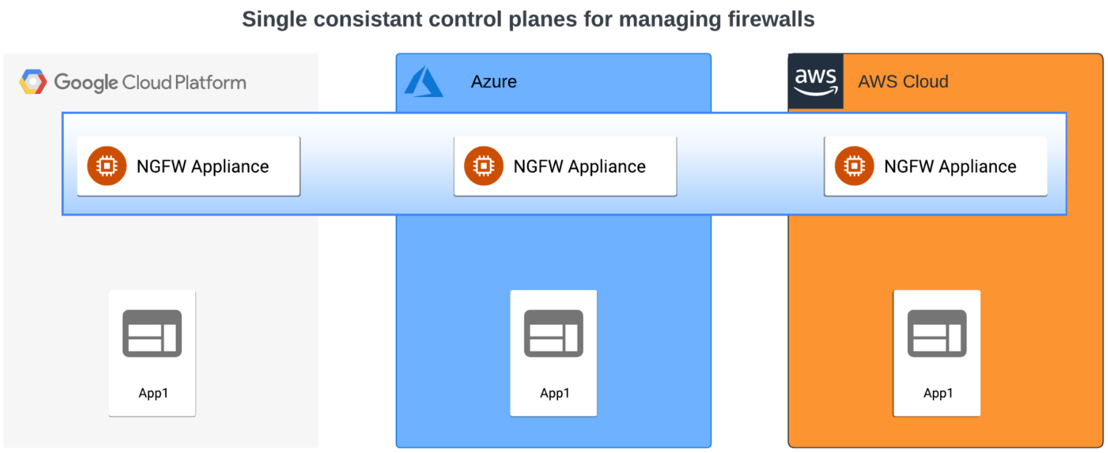
Cloud Agnostic Design

Cloud providers offer native capabilities like firewall, message systems, databases, etc that are proprietary to that provider. Customers often wish to not tightly integrate their system with cloud specific service especially with a multi cloud system. For consistency the components of the system should be able to operate on all cloud providers and/or follow open source standards. To leverage GCP native firewalls would mean that the same cannot be used for different cloud providers. This will lead to an undesirable design as inconsistent sets of services are used on each cloud provider. Customers have to manage the control plane for the firewall differently for each of the cloud providers.

Different control planes for managing firewalls



In contrast using a third party NGFW provider leads to a more consistent control plane experience across clouds. Using a consistent tool across the cloud providers leads to overall less management overhead and familiar experience.

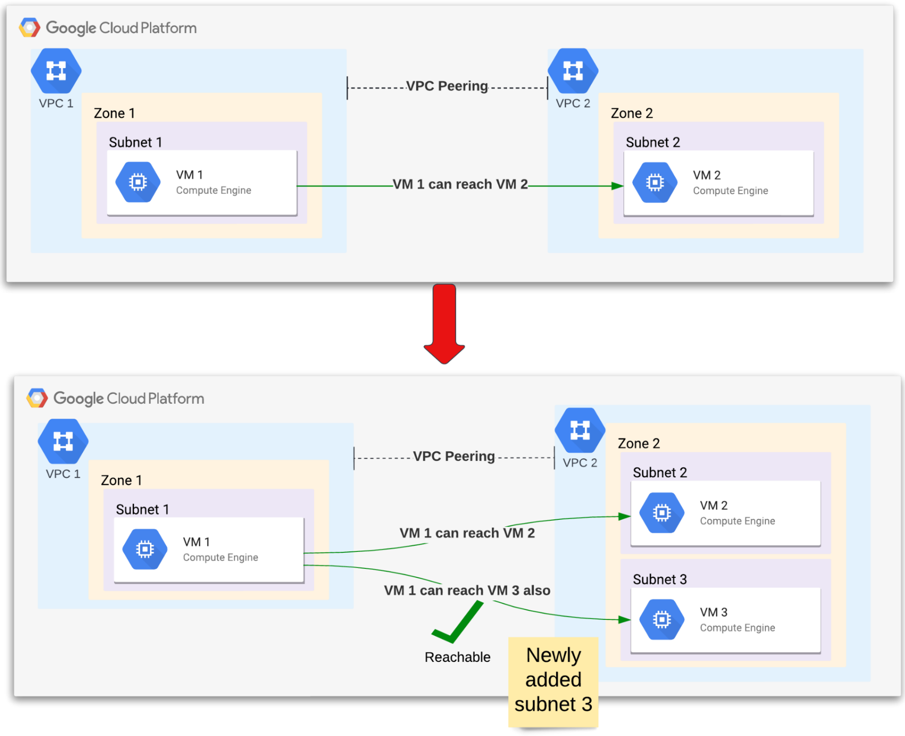


Management

A rule of thumb is to automate as many infrastructure components as possible to reduce overall management overhead. Ideally, when a change is applied to one system, another dependent system should absorb the cascading changes transparently. Building a static system that does not adapt automatically leads to low velocity and high maintenance cost.

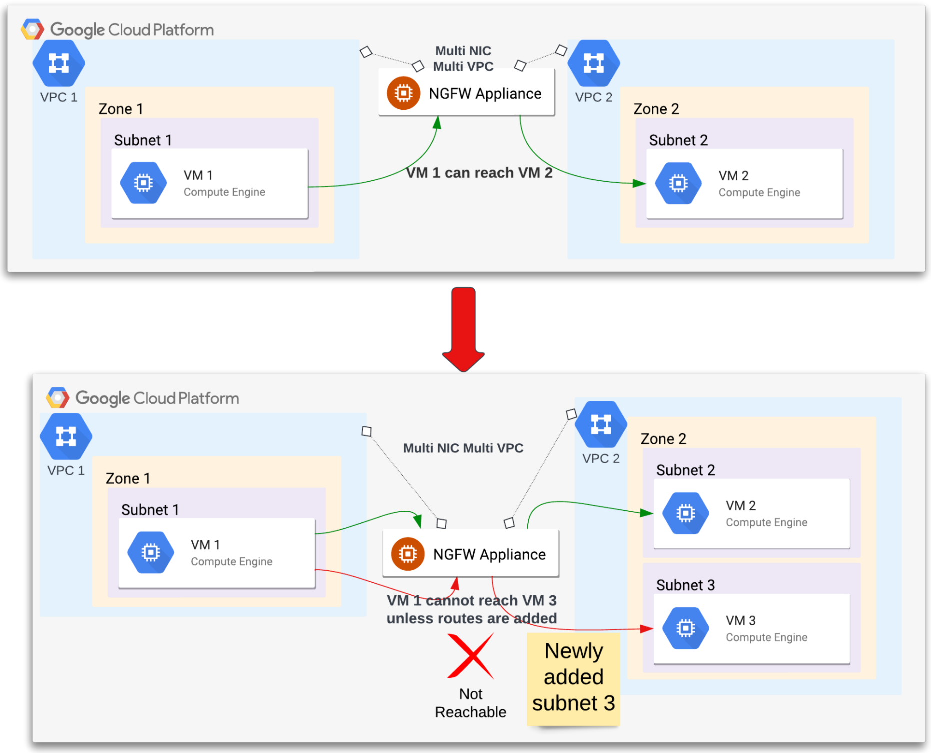
Routing

Often routing and other networking constructs will have to be changed as the footprint of the infrastructure grows on GCP. Typically on GCP, when new subnets are added, updated or removed all the dependent systems seamlessly work. If multiple VPCs are involved and peered natively, any subnet changes are reflected automatically across the fleet. Similarly when connectivity is established with on-prem and cloud routers are used, any changes to routing on GCP are reflected to on-prem automatically. In the picture below as VPC1 and VPC2 are peered VM1 in VPC1 can reach VM2 in VPC2. When a new subnet (subnet3) and VM3 is added in VPC2, VM1 in VPC1 can automatically reach VM3 in VPC2 also.



Some NGFW can also integrate directly with GCP using Network Connectivity Center (NCC). The advantage of using NCC is to eliminate the need for manual routing especially if the NFGW

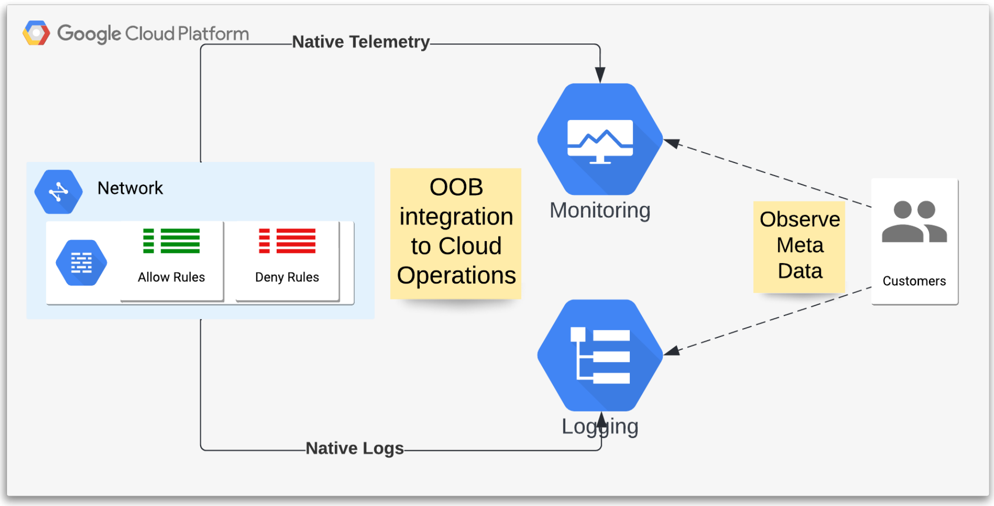
also supports routing capabilities. Without using NCC, operators of appliances have to manually keep routes up to date whenever some networking changes are made inside VPCs. Some appliances vendors like Fortigate can directly push route changes via GCP APIs.
When NGFW appliances do not leverage NCC, they are typically sandwiched between the VPCs. The routes are not exchanged automatically using this design. Operators have to force the traffic from one subnet to another only by passing through the NGFW using static routes . This implies that the configuration of NGFW may have to be changed along with the routing table when a new subnet is added and the connectivity were to be established with this new subnet.



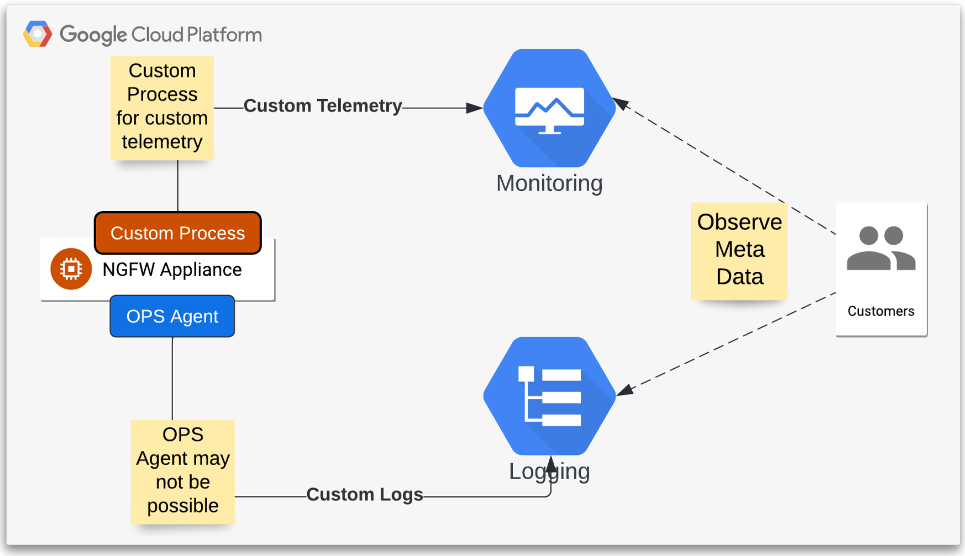
Monitoring

All GCP services are tightly and natively integrated with Cloud Operations. Google manages the integration to Cloud Monitoring and Cloud Logging out of the box . Typically for many services it is done without any involvement from the customer. For GCP firewall, an option is provided to the customer to enable or disable firewall logging. On enabling the logging, metadata information about the firewall rules like hits, source and destination address, etc are automatically logged into the Cloud Logging. This provides useful information about the firewall rules when needed during troubleshooting. Furthermore, as native firewalls are managed by

Google, customers need not have to monitor these services as the onus is on Google SRE teams.

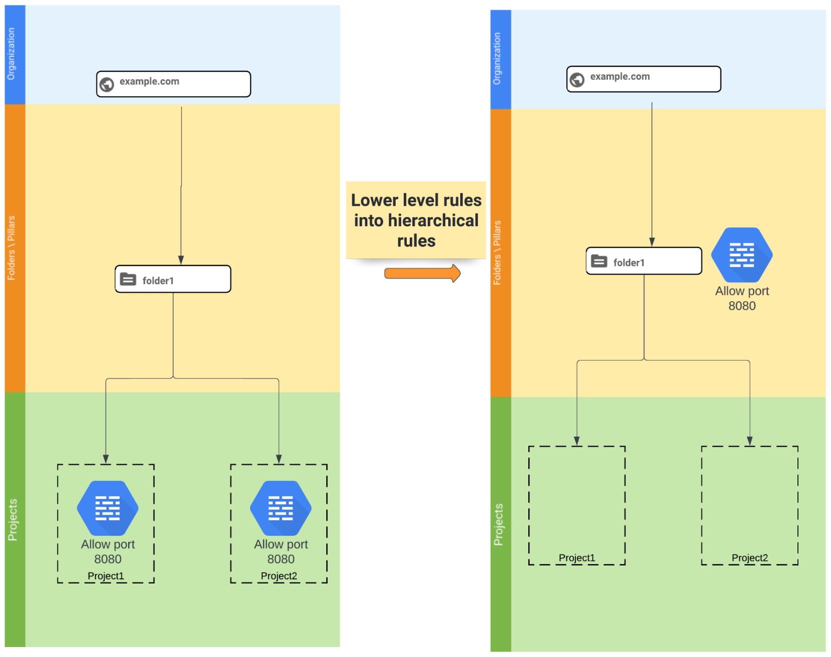


In contrast, when NGFW is provisioned, additional effort is required to provide deep integration between these appliances and Cloud Operation suite. Typically these NGFW are provisioned on VMs that leverage Google Compute Engine (GCE). When GCE is used for these appliances, most of the infrastructure related telemetry (CPU utilization, Storage Utilization, Network rate, etc) are natively integrated to Cloud Monitoring. However, additional tools/processes have to be developed if there are any custom NGFW specific telemetry, required for monitoring the appliances.
GCP offers open-source Ops Agents which can be installed on GCE NGFW appliance VMs. These agents provide deep integration between these 3rd party appliances and Cloud Operation for both monitoring and logging use cases. NGFW may persist application logs in non-standard locations but the agents also provide capability to configure and stream logs from these locations . However, some appliances may not provide any ability to alter the VM image or install these agents on top. In this case, proprietary solutions have to be adopted as dictated by the NGFW vendors.



Hierarchical firewall rules

Quite often, as the enterprise cloud footprint grows, so does the number of firewall rules and other networking constructs. GCP provides hierarchical firewall rules to reduce the toil of managing a large number of firewall rules. With hierarchical firewall rules, many lower level rules can be eliminated and/or converted into smaller number of higher level rules if the organization resource hierarchy is set up correctly. If the number of firewall rules are kept smaller, it reduces the overall manageability overhead for security teams.



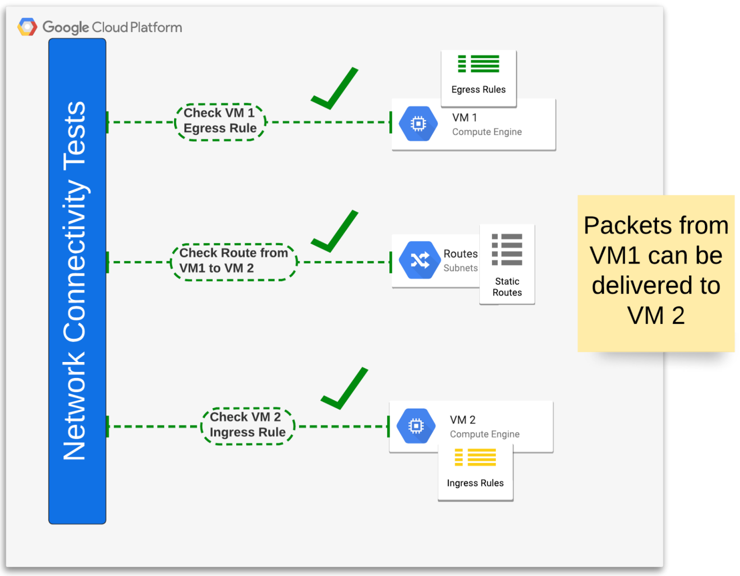
When NGFW are set up , they are typically sandwiched between VPCs and are provisioned inside a GCP project. Currently there isn't a way to set up NGFW at a higher level in the resource hierarchy. When all firewall based decisions are delegated to NGFW appliances, then hierarchical firewall rules cannot be used to reduce the number of rules. Each NGFW vendor may have a different set of features to reduce the excessive set of rules.

Troubleshooting

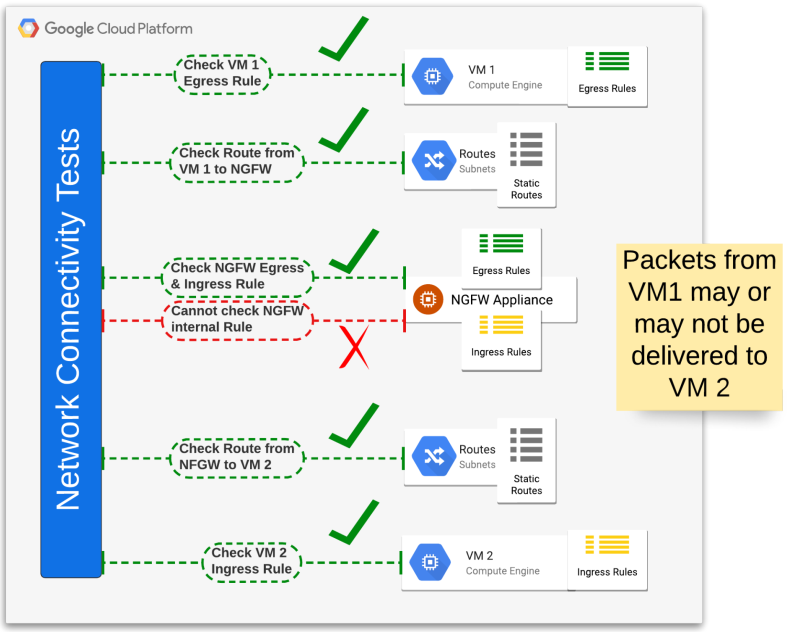
GCP offers a few services that help with debugging network related issues. Especially for a large organization with a large GCP footprint, these tools could serve an important role for security and network operators.

Network Connectivity Tests

Network connectivity tests help with analyzing connectivity among different endpoints. Ideally network operators should create a suite of test cases for all possible endpoints in their organization. They should be used while making network changes just like unit tests are used while developing software. Connectivity test provides insights for determining misconfigured firewall rules, misconfigured network routing, etc. When native firewall rules are used, the network connectivity test returns almost a true representation of connectivity. If the packet cannot be delivered, the connectivity test clearly identifies the root cause of that behavior. For instance, it can show which firewall rule is disallowing the packet or which route is misconfigured. Retrieving this type of information can assist while making network firewall changes. Running all the tests after making a network change will provide confidence to the operators that overall connectivity to major endpoints are still intact or broken.



When NGFW appliances are used then network connectivity tests cannot portray the true representation of connectivity. When connectivity is being tested between 2 endpoints via NGFW, the test will verify that the packets can be delivered to NGFW, but it may not fully determine what NGFW could do. If NGFW drops the packets because of some internal firewall rule, the connectivity test may still give an incorrect result that the connectivity can be established.



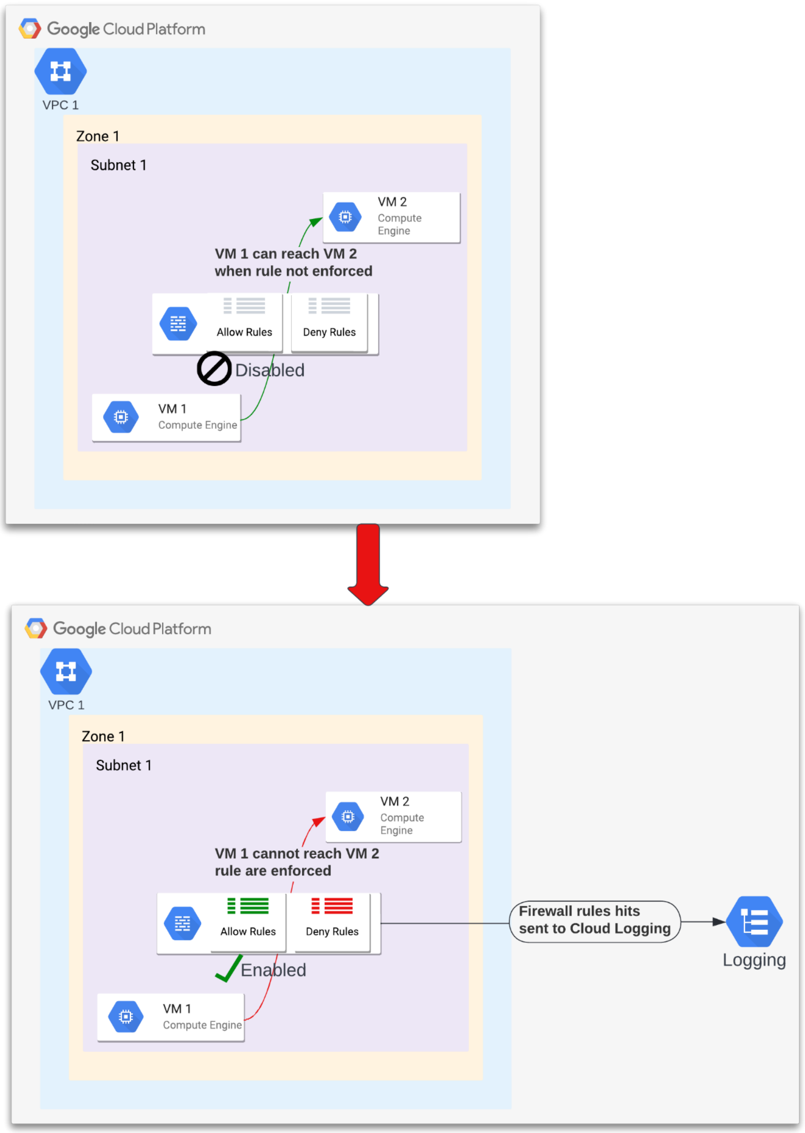
Firewall insights

With multiple firewall rules, it is imperative that they don't step over each other. As the number of rules grows, so does the potential of these rules shadowing other rules. Google automatically generates firewall insights that detects shadowed rules directly inferencing from the rules configuration. Furthermore, it is valuable to know if certain firewall rules are being overly permissive or how many times a rule is firing. When logging for these rules is enabled, firewall insights also help in detecting such scenarios. This type of information can assist operators to further refine existing firewall rules so that they are kept in pristine state and follow the principle of least privilege. It's recommended that operators maintain a cadence to review these insights and act upon them.

This GCP feature loses some of its value when NGFW appliances are used. Typically with NGFW appliances, an overly permissive firewall rules are authored on GCP native firewall so that all types of traffic can be routed to NGFW itself. NGFW appliances in turn make the decision of what traffic to allow or deny. This implies that the insights of the firewall rules would be more useful if they are generated for the rules native to NGFW, rather than native to GCP. Similar features for NGFW appliances will be vendor specific.

Enforcement

When working with firewall rules it is important to have capabilities which enable testing changes easily. GCP firewall provides enforcement capabilities to troubleshoot the rules first without deleting them altogether. Customers can also log firewall rules so that they can debug them on real traffic. As this provides visibility, it reduces the overall risk of making changes to firewall rules.



In contrast when NGFW appliances are used, similar functionality may or may not be provided by the vendor. The risk of breaking existing connectivity is high, without this capability to test.

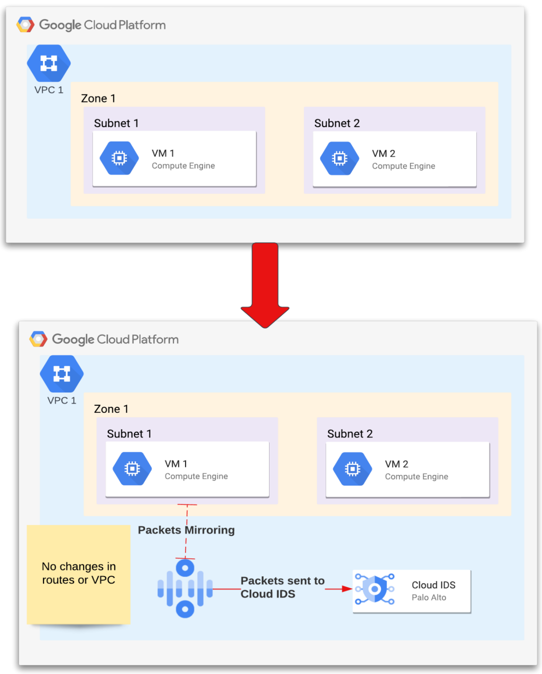
Intrusion Handling

Many customers require intrusion handling and finer grained control over native firewalls.

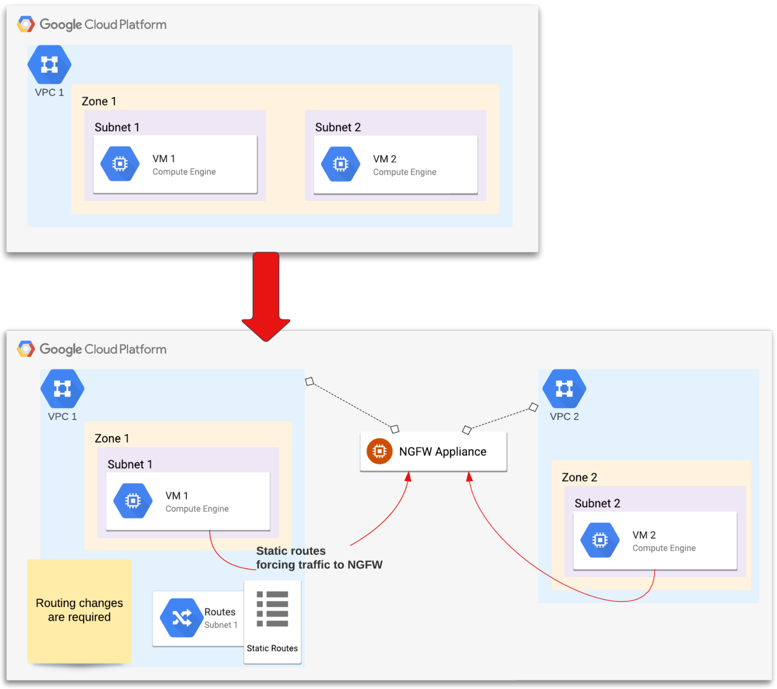
IDS

Google has partnered with Palo Alto Networks and built a fully managed Intrusion Detection Service based on Palo Alto Technology. Cloud IDS provides intrusion detection capability that one would expect from Palo Alto appliances. A few advantages of using Cloud IDS over another appliance with similar functionality are

1) VPC design does not need to change. Existing setup of the network can stay intact with Cloud IDS. This non-invasive integration of the network with Cloud IDS is made possible by packet mirroring capability on GCP. Existing GCP customers can stream packets to a fully managed Cloud IDS system simply by mirroring subnet traffic to it.



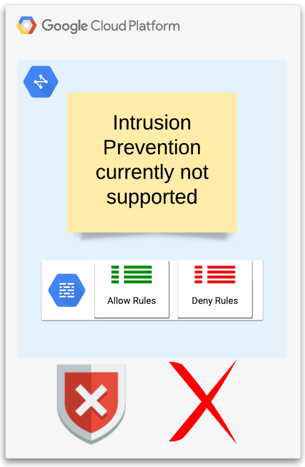
In contrast when NGFW appliances are used for IDS capability, they require changes into the existing VPC, as those appliances are typically sandwiched among them. This almost always has an impact on existing routes.

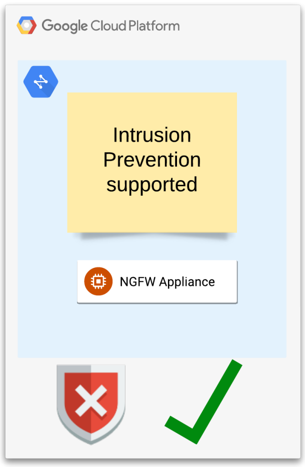


2) Customers do not need to manage another system. As Cloud IDS is fully managed by Google, customers can rely on Google SRE teams to ensure SLAs are met. When an appliance IDS is used, the customer SRE teams have to fully operate and manage it in their HA/DR setup.

IPS

Customers often need automated capability to not only detect the malicious activity and alert the operators, but also to block the traffic from source and let legitimate traffic flow though. IPS often helps with speedy recovery from a malicious activity as it automatically handles the process of stopping the attack.
Currently* intrusion prevention capabilities are not provided natively by GCP firewall. If customers have a strong requirement for intrusion prevention, they should choose a vendor that provides that capability.


Google Cloud Platform
Intrusion Prevention currently not supported
Allow Rules
Deny Rules


Google Cloud Platform
Intrusion Prevention supported
NGFW Appliance

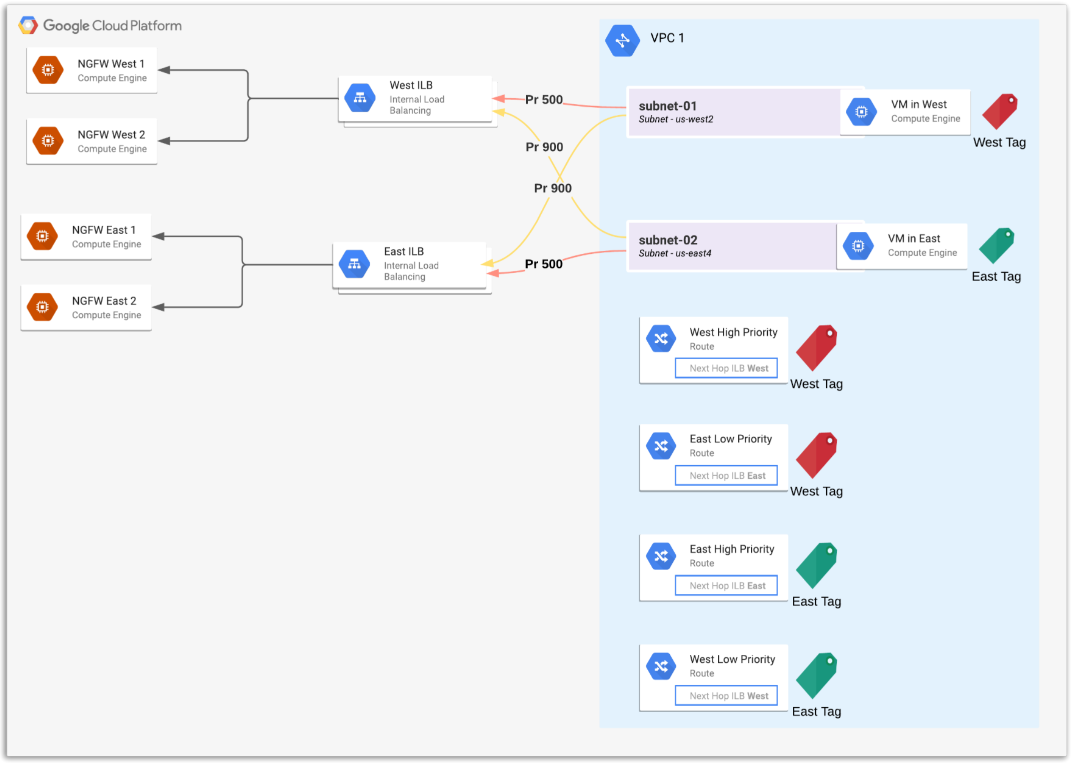
Continuous availability

To build a resilient infrastructure architecture every component in a system should be able to handle service, zone and regional failures. Automatic failovers with least manual intervention is a good practice to follow. To provide continuous availability, a system should be able to automatically handle multiple types of failures. One such design to allow for continuous availability is to run multiple copies of the system on multiple locations all the time. Operating the system with multi zone and multi region in an active-active setup is a good baseline to strive for.
With native GCP firewalls, customers do not have to be responsible for continuous availability of these systems as many of these services are distributed and global in nature. In contrast to provide continuous availability with NGFW appliances, special design and consideration have to be taken into account.
Some NGFW appliances can leverage network connectivity center and also work as a router between VPCs subnets . Those appliances should support TCP session failover and region failover. This is possible when NGFW can also exchange routes with GCP Cloud Router as shown here for Palo Alto NGFWs.

If NGFW doesn't support all that capability, then the following GCP native design can be leveraged to achieve some degree of High Availability.

- To provide HA within the same region Internal Load Balancers can be used. ILB are used in front of NGFW to ensure health checks of these appliances are measured and if unhealthy, taken out of rotation. Multiple instances of appliances in different zones can handle zonal failure in this design. For example in the diagram there are 2 instances of NGFW appliances both, in the West region namely NGFW West1 and NGFW West2. If NGFW West1 becomes unhealthy, all the traffic will automatically be routed to NGFW West2 in the same West region.
- To provide HA within the different regions Policy Based Routing (PBR)** can be used. PBR is used to ensure multiple routes with different priorities are set up for different clients (VM in West and VM in East) accessing the NGFWs. In normal operation, route priority will dictate that clients in one region will send traffic to ILB in that region only. Currently* PBR does not support automatic detection of failure and route updates with Internal Load Balancer as the next hop (ILBasNH). Ideally when all NGFW instances become unhealthy in one region, then the routes should be automatically updated so that traffic can be sent to ILBs in another region. Notice that using this design,

customers still have to manually update the routes.

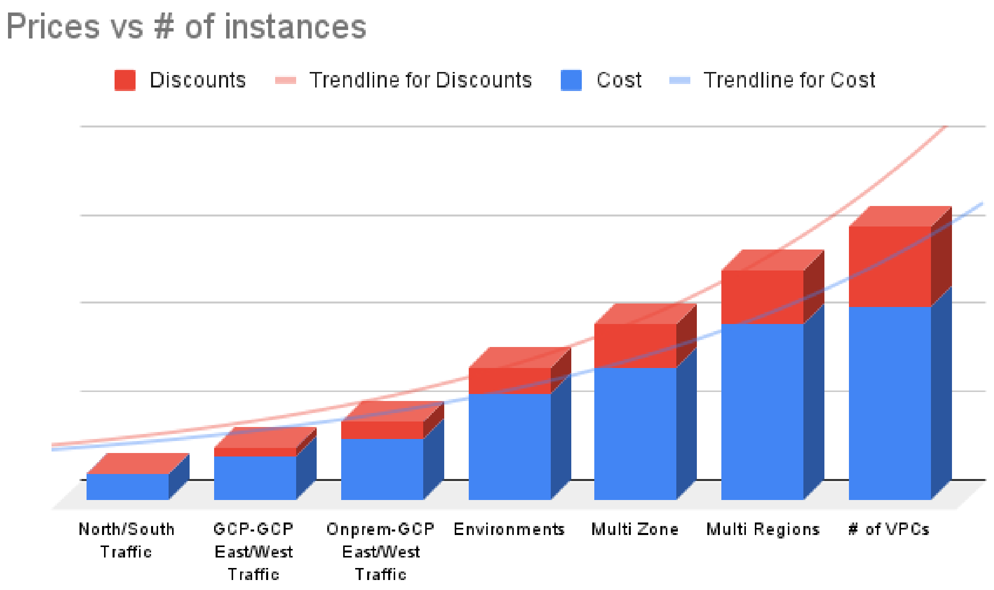


The above design is for providing continuous availability using only 2 regions and multiple zones. It requires many static routes with tags to be authored. When customers have their GCP footprint in more than 2 regions, then adding the 3rd region into the design requires authoring exponentially more routes.

Price

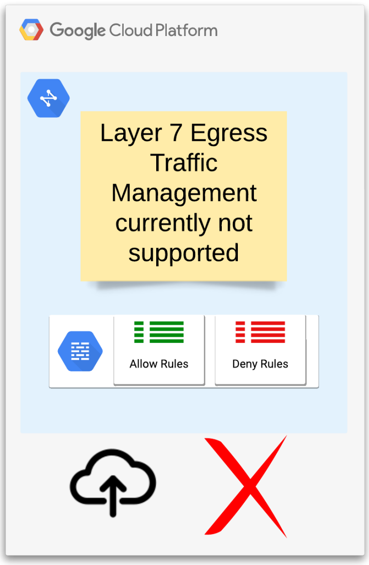
A lot of factors are collectively used to determine the cost of running native services or NGFW appliances on GCP. Typically the appliances may have license cost based on how many instances are used. From a cost perspective, special design considerations could be taken when working with NGFW appliances . In case the price is based on number of instances, the following factors should be considered to get to the optimal number

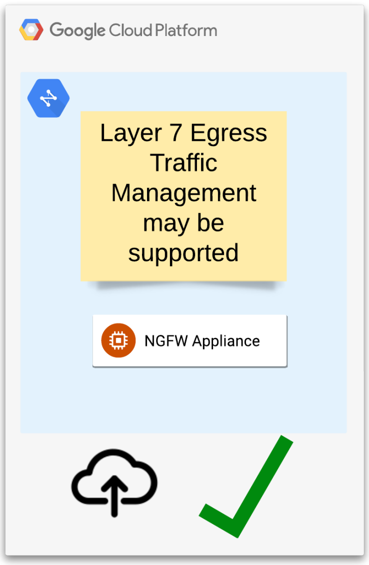
1) Traffic Type
    a) Separate North/South traffic instances . It is a good practice to have a different set of appliances for handling public ingress/egress traffic from private traffic. This can increase the overall number of appliances in the enterprise.
    b) Separate Internal East/West traffic instances . Another good practice is to segregate the set of appliances for traffic originating from private networks within GCP (from different subnets) from the set of appliances used for public traffic. This design separates backend traffic from public traffic, which helps in defining different scalability and resource requirements for these appliances.
    c) Separate On-prem East/West traffic instances . To ensure big data transfers between on-prem and GCP do not impact other types of traffic, it is advisable to separate on-prem traffic also with different sets of appliances.
2) Environment segregation
    a) Different set of appliances for dev, qa and production environment. To achieve isolation among environments, it is recommended to dedicate different set NGFW appliances for different environments.
3) HA Setup
    a) More than one instance per zone. To ensure the system is resilient, it is recommended to have NGFW appliances operate in multiple zones per region. A desired number of minimum zones can impact the count of appliances proportionally.
4) Multi Regions
    a) Separate instances per region . To provide continuous availability, a recommended approach is to provision NGFW appliances on more than one region at all times.
    b) Number of regions . Total number of appliances grows as a factor of the total number of regions the enterprise wants to operate in.
5) VPC Setup
    a) Number of VPC in the organization. Currently on GCP an appliance can only support 8 nics. If the organization has more VPCs among which appliances have to be sandwiched, then the only options would be to add more appliances . This can impact the number of appliances significantly also, therefore it is important to design the network carefully from start.


Prices vs # of instances
Discounts
Trendline for Discounts
Cost
Trendline for Cost
North/South Traffic
GCP-GCP East/West Traffic
Onprem-GCP East/West Traffic
Environments
Multi Zone
Multi Regions
# of VPCs

Egress

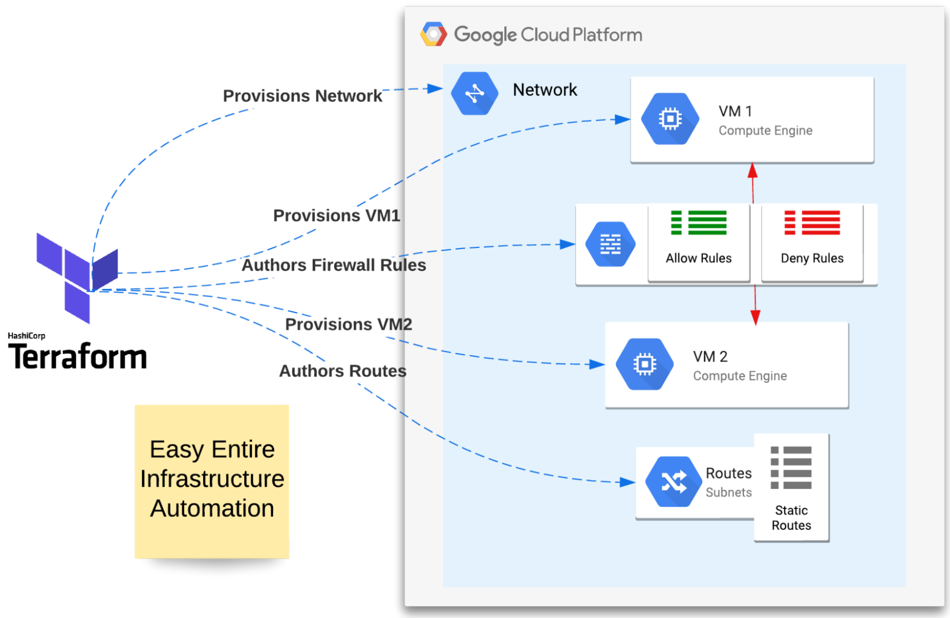
A common requirement from customers is to centrally and effectively control all egress traffic originating from any workloads across the organization. Currently* GCP supports only L3/L4 management of egress traffic using native firewall rules. Just using L3/L4 controls often makes it quite difficult to manage and control egress traffic to the internet. There are many use cases where this traffic management is best suited to be done at Layer 7 (Http/Https). For example managing egress traffic based on URLs.
As GCP currently* does not support egress traffic management at Layer 7, using an NGFW appliance with this capability is a better choice.



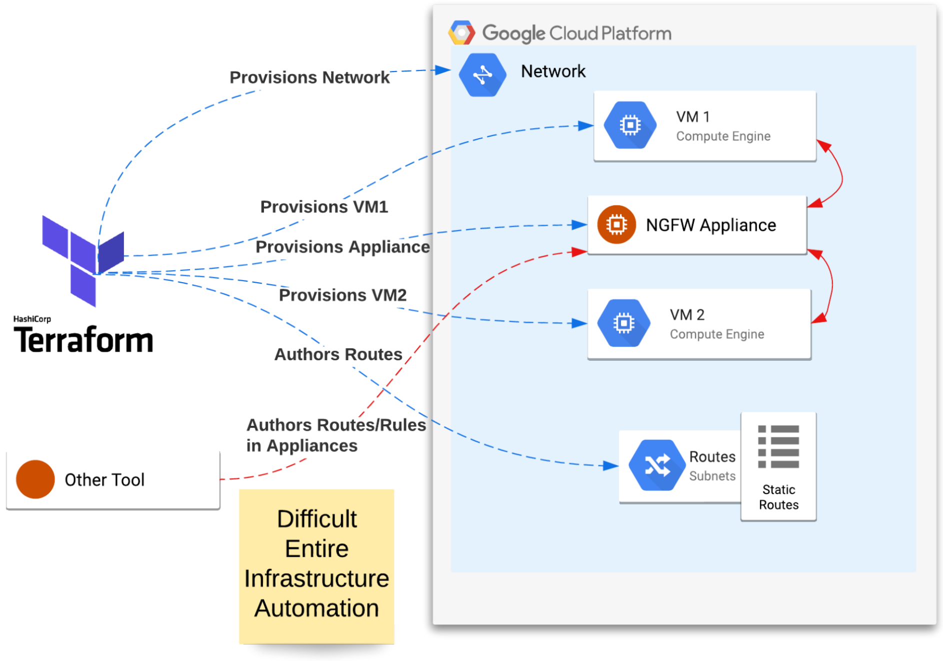


Automation

The best practice to work with cloud providers is to use infrastructure as code (IaC). On a regular basis customers should try to repave entire infrastructure to ensure that in case of urgency they can recreate their setup with ease. Tools like terraform make this process declarative and easy. Most of the GCP services have integration with terraform . As and when new services and features are released, the new terraform implementations are also developed. These terraform modules can be leveraged to build GCP native services like VPC network , routes, firewalls, etc. Often out of box modules can be used with ease to build the entire GCP infrastructure.

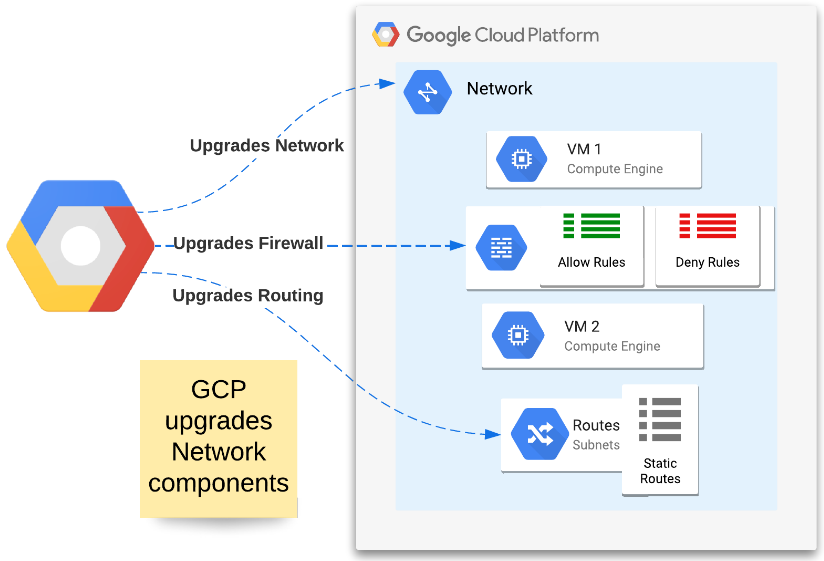


When appliances are used for handling firewall rules, routing, etc, then to build complete automation with terraform becomes more difficult. From a GCP perspective , using IaC, the NGFW appliances can be provisioned as virtual machines and native routes and native firewall rules can be authored. As these appliances are mainly responsible for handling their own internal routing and their own internal firewall rules, some other processes have to be built to automate this configuration also.


Google Cloud Platform
Network
VM 1
Compute Engine
NGFW Appliance
VM 2
Compute Engine
Routes
Subnets
Static Routes
Provisions Network
Provisions VM1
Provisions Appliance
Provisions VM2
Authors Routes
Authors Routes/Rules in Appliances
HashiCorp Terraform
Other Tool
Difficult Entire Infrastructure Automation

Upgrades

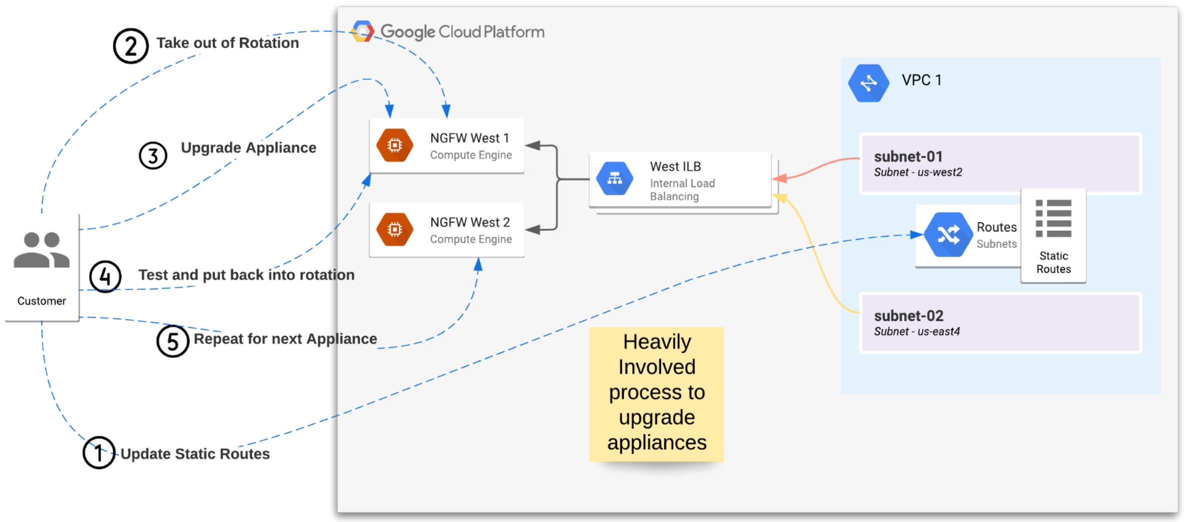
As GCP native services are managed by Google, customers are not responsible for managing upgrades of those services. The services like VPC, firewalls, routing, etc are released without any involvement from the customers or downtime for any applications leveraging them.



Appliances upgrades and patches are completely customer responsibility and have to be done carefully to avoid any downtime. As these appliances are handling the majority of the traffic, removing any instance could have a direct impact on performance. Typically NGFW appliances are taken out of rotation and upgraded in a rolling fashion. Based on the architecture of how these appliances are setup, to reduce the risk of downtime, it could involve

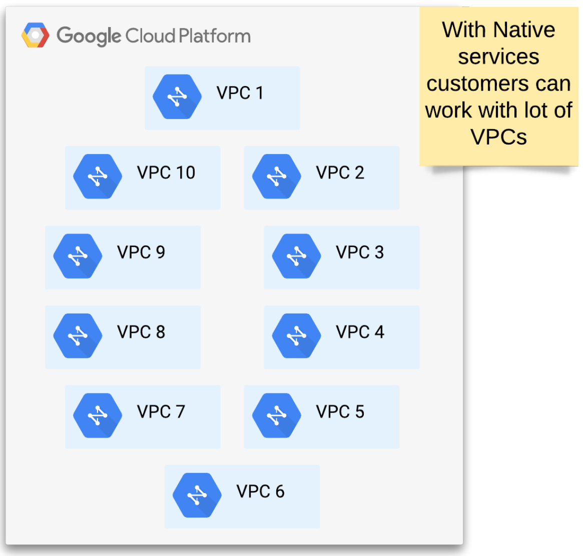
- Changing static routes within a region
- Switching traffic from one region to another

In a multi region setup sometimes rolling the upgrade could take weeks to complete.



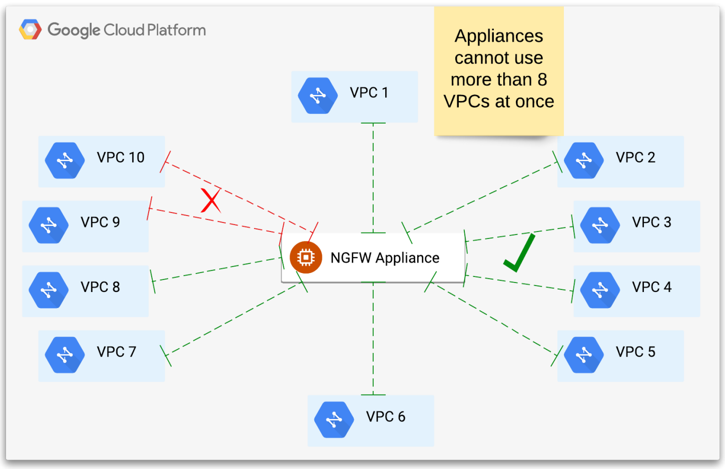
VPC Limits

Based on the design requirements of an organization, the numbers of VPCs may vary. When using native firewalls, the number of VPCs would have its own quota and limits. To reduce management overhead, it is recommended to have less number of VPCs, however Customers can choose to have a large number of VPCs in their organization.



Typically appliances are sandwiched among VPCs by using multiple network interfaces.

Currently a virtual instance with multi nic can use at most 8 interfaces simultaneously. This implies that a NGFW appliance can support at most 8 VPCs at a time. If the number of VPC for an organization is more than 8, then they have to add more appliances that can impact the cost and overall management overhead.



Conclusion

There are pros and cons for using both GCP native firewall and NGFW appliances. Depending on the customer's requirements, NGFW could be a better choice if the above considerations are also taken into account. In other cases if customers get the feature parity they are looking for with GCP native firewall, then later may be a better choice. This document mainly provides general guidelines and leaves the final decision to customers and as always, it will depend on the individual use cases.

Additional Resources

- Deploying Multi Cluster Gateway
- Symmetric Hashing
- Setting up Internal TCP/UDP Load Balancing for third-party appliances | Google Cloud
- Container Series by Palo Alto
- Network Connectivity Center
- Cloud IDS

* Google is working to add more features and capabilities in this and related area.
** PBR is scheduled to be released for private preview in september 2022