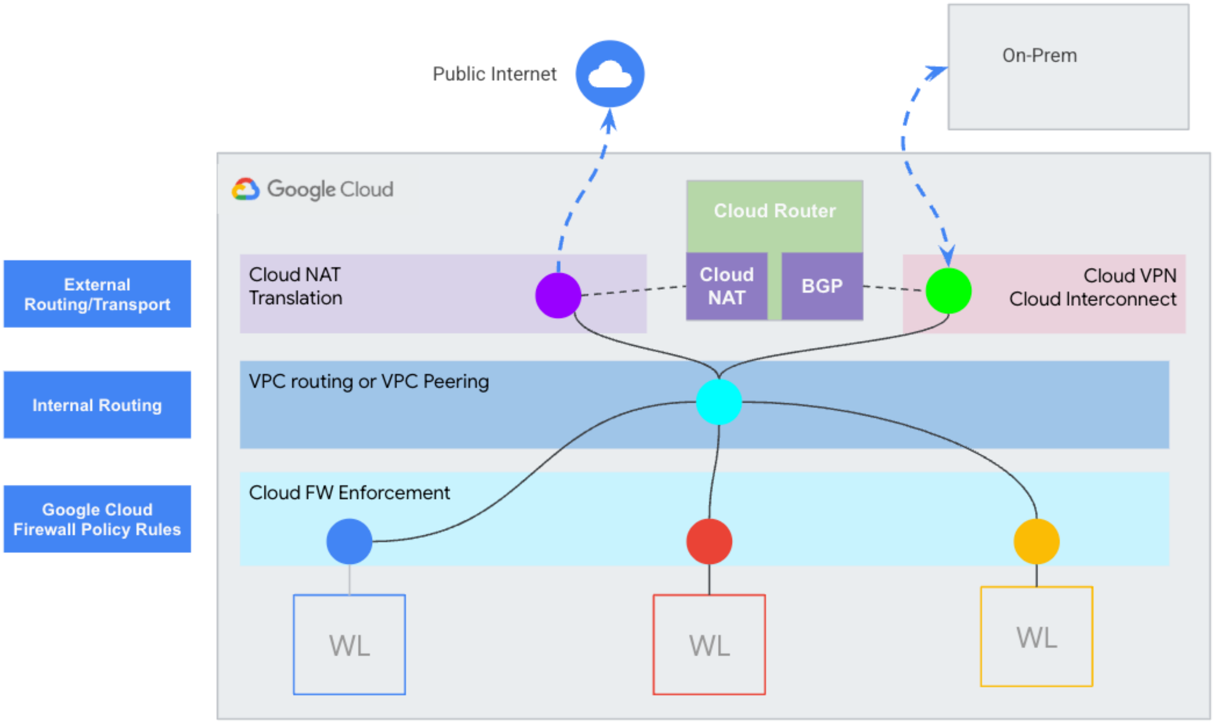
Tips for migrating on-prem firewall appliances to Google Cloud services and Cloud Firewall policies

In addition to acting as a network perimeter security device, traditional firewall (FW) appliances often serve additional functions like NATing, Layer 7 inspection and VPN termination. In Google Cloud, the responsibility of those functions are **not** placed solely on Cloud Firewall, but rather on a collection of independent services. The intent of this document is to help in a migration from on-prem firewall appliances to using Google Cloud services like Cloud Firewall policies, Cloud NAT and Cloud VPN.

Architecture

The diagram below shows a layered visualization of how multiple GCP products work in conjunction to match the functions of a traditional firewall.



The previous diagram shows how evaluation of Firewall Policy Rules (and VPC Cloud Firewall Rules) is done at the host level, rather than at a central point for all traffic.

Workflow

Egress (traffic initiated and leaving FROM Google Cloud resource)

- Egress Cloud Firewall policy rules are evaluated
  - If DENIED, traffic flow stops
- VPC routing decision completed based on Google Cloud route order
- Cloud NAT (or public IP) is responsible for translation of addresses to the Internet from a VPC
- Cloud VPN or Cloud Interconnect are responsible for Hybrid Connectivity

Ingress (traffic entering a Google Cloud resource)

- Cloud NAT is responsible for stateful translation of response traffic which was originally initiated from Cloud NAT.
  - Cloud NAT does not allow traffic initiated from the Internet into Google Cloud.
- Public IPs assigned to individual resources statically translate Internet traffic to VMs.
- Cloud VPN or Cloud Interconnect are responsible for Hybrid Connectivity
- VPC routing decision completed based on Google Cloud route order
- Cloud Firewall state is evaluated to allow response traffic
- Ingress Cloud Firewall policy rules evaluated
  - If DENIED, traffic flow stops

Migration

Most firewall (FW) appliances, either virtual or physical, are deployed in one of two modes: zone based or one where an Access Control List (ACL) is applied to an interface in a particular direction. In both cases, the firewall's primary security purpose is to **protect one perimeter or network segment** from another. **Cloud Firewall, however, is not meant to act as perimeter devices**; rather, it is a fully distributed set of rules to protect resources, such as virtual machines (VM).

This first order in a migration may be to replicate the *logic* of firewall rules into Google Cloud. The major difference between a firewall appliance rule (zone or ACL based) and a Cloud Firewall policy rule is the notion of **TARGET**. A Cloud Firewall policy consists of multiple rules. However, not every rule applies to every resource. That is defined by the rule's "TARGET". Let's take a look at each of these general attributes, including TARGET, in a firewall appliance and in a Google Cloud Cloud Firewall for a standard ***ingress*** rule:

- Priority:
  - Firewall appliance:
    - Firewall appliances rules typically have a line number or an index number that are evaluated with a **first match logic**, where the lower value is a higher priority.

    - Cloud Firewall:
        - Google Cloud Cloud Firewall policy rules all have a unique priority number from 0 - 2,147,483,643 (inclusive) also with a **first match logic**, where the lower value is a higher priority.

- Action:
    - Firewall appliance:
        - Firewall appliances usually have a permit/allow or a deny/block type action. At the end of most ACLs is an "implicit deny".
    - Cloud Firewall:
        - Cloud Firewall rules also have an allow and a deny action. There is an "implicit ingress deny" rule and an "implicit egress allow" rule. In addition, in some cases, Cloud Firewall rules have a third action: "go-to next". However, that action is outside the scope of this document.

- Protocol:
    - Firewall appliance:
        - A firewall appliance rule usually contains *at least* a layer 3 protocol. If no protocol is configured, it is typically ALL protocols.
    - Cloud Firewall:
        - In a Cloud Firewall rule, there is an option for ALL protocols or specific protocols: TCP, UDP or Other. The following protocol names can be used: *tcp*, *udp*, *icmp*, *esp*, *ah*, *sctp*, and *ipip*. All other protocols must be defined through their IANA protocol numbers.

- Source:
    - Firewall appliance:
        - Traditional L3-7 firewalls often use IP addresses or resolved DNS names as the source, even if the source is "any".
    - Cloud Firewall:
        - Cloud Firewalls have an additional option for an ingress rule. Sure, IPv4 or IPv6 addresses are common options and available in a Cloud Firewall rule. However, they can also be configured to use Tags as a source. For the remaining portion of this document, the term "tag" suggests an IAM-governed Tag, as opposed to a Google Cloud network tag.

- Destination:
    - Firewall appliance:
        - Traditional L3-7 firewalls often use IP addresses or resolved DNS names as destinations. For HTTP inspection rules, HTTP Host header can also act as the destination.

    - Cloud Firewall:
        - The destination on a Cloud Firewall ingress rule is *typically*[1] not required. Because the Cloud Firewall is fully distributed, the destination is most often the target. There is, however, a preview feature that allows a destination IP to be specified for an ingress rule.

- Destination Port
    - Firewall appliance:
        - A firewall appliances rule often contains a layer 4 port or the associated layer 7 application. If no port or application is configured, it is typically ALL ports or applications.
    - Cloud Firewall:
        - In a Cloud Firewall rule, if a protocol such as TCP or UDP is chosen, there is also an option to configure a port. There is no option for application.

- Direction:
    - Firewall appliance:
        - When an ACL is applied to a firewall's interface, it is applied in a direction: in or out. For zone based firewalls, the direction follows the direction of the initiated connection. For example: *From "untrust" To "trust"*. Stateful firewall appliances will construct a 5-tuple flow, and allow return traffic to pass.
    - Cloud Firewall:
        - The direction of a Cloud Firewall rule is either "ingress" or "egress". Ingress rules are used when the protected resource is the destination. Egress rules are used when the protected resource is the source of traffic. Cloud Firewall is also stateful, and thus return traffic will be allowed to pass.

- Target:
    - Firewall appliance:
        - For ACL based firewall appliances, the target is an "interface+direction". For zone based, the target is similar to the direction and follows a zone to zone flow.
    - Cloud Firewall:
        - The target will specify which resource or resources to apply a particular rule. Targets on an ingress Cloud Firewall Policy rule can be one of three options: all instances in the network, VMs with given Tags or service accounts.

[1] As of the writing of this document in late 2022, there is a preview feature that allows a destination IP to be configured within an ingress rule.

Below is a table for both ingress and egress Cloud Firewall Policy rule attributes and their respective firewall appliance rule attributes:

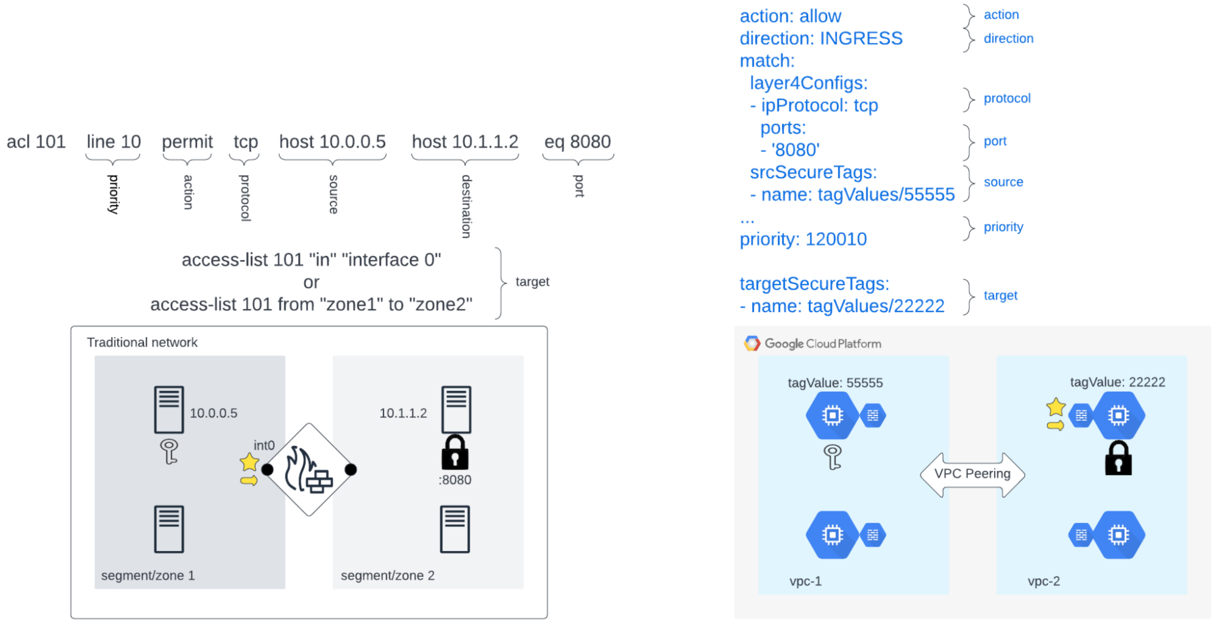
Firewall Appliance	Ingress Cloud Firewall Attribute	Egress Cloud Firewall Attribute
Line Number Index Number	Priority**	Priority**
Action	Action [Allow | Deny | Goto Next]	Action [Allow | Deny | Goto Next]
Protocol	Protocols [All | TCP | UDP | Other]	Protocols [All | TCP | UDP | Other]
Source	(Remote) Source [IP | Tag* | FQDN | Add. Grp | NTI]	TARGET [ALL | Tag* |Service Account]
Destination	TARGET [ALL | Tag* |Service Account] {IP address}	(Remote) Destination [IP | FQDN | NTI]
Port / Application	(Local Destination) Port {TCP ports | UDP ports}	(Remote Destination) Port {TCP ports | UDP ports}

*Tag = IAM Governed Tag for Cloud Firewalls

** Priority Numbers must be unique inside a Cloud Firewall Policy

Rule Comparison

For our first example, let’s take a look at a typical firewall rule that would allow “key” to access “lock” on TCP port 8080 in both a firewall appliance (pseudo code) and in a Cloud Firewall policy rule.



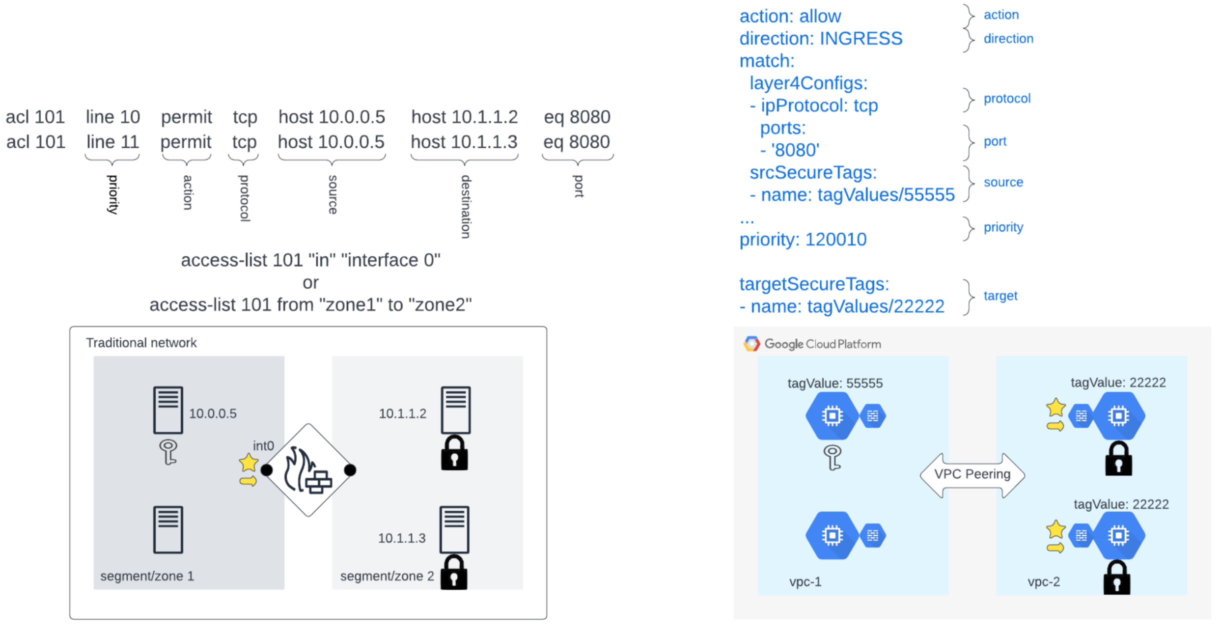
```
acl 101   line 10   permit   tcp   host 10.0.0.5   host 10.1.1.2   eq 8080
```

```
access-list 101 "in" "interface 0"
or
access-list 101 from "zone1" to "zone2"
```

```
action: allow
direction: INGRESS
match:
 layer4Configs:
 - ipProtocol: tcp
   ports:
   - '8080'
 srcSecureTags:
 - name: tagValues/55555
...
priority: 120010

targetSecureTags:
- name: tagValues/22222
```

Note that there is no destination IP configured on the Cloud Firewall rule, and the target specifies the resource to which the rule should be applied. If the source "key" required access to multiple resources you may see multiple rules in a traditional firewall configuration. However, with Cloud Firewall, it is best practice to use a single rule and add the target tag to the additional resource. In our example, if we add tag "22222" to the second VM in VPC-2, the respective Cloud Firewall rules are applied to that resource.



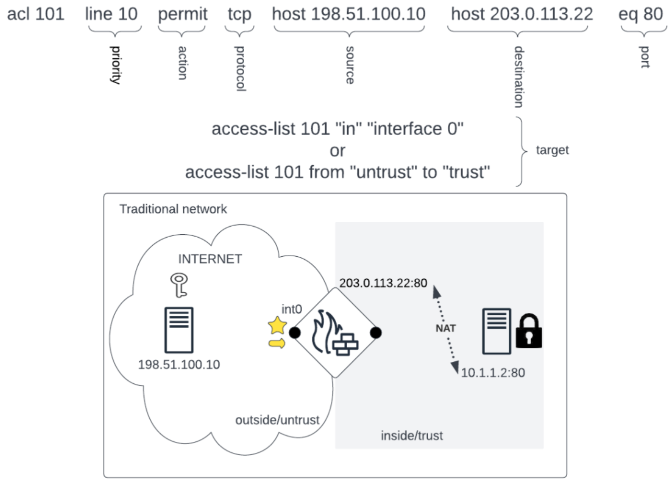
```
acl 101   line 10   permit   tcp   host 10.0.0.5   host 10.1.1.2   eq 8080
acl 101   line 11   permit   tcp   host 10.0.0.5   host 10.1.1.3   eq 8080
```

```
access-list 101 "in" "interface 0"
or
access-list 101 from "zone1" to "zone2"
```

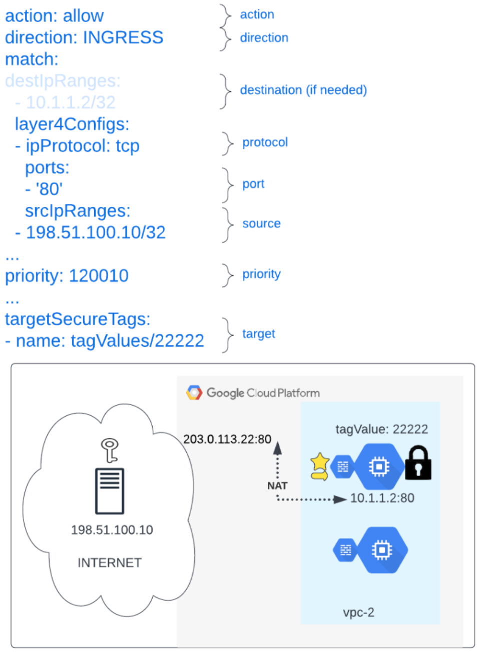
```
action: allow
direction: INGRESS
match:
 layer4Configs:
 - ipProtocol: tcp
   ports:
   - '8080'
 srcSecureTags:
 - name: tagValues/55555
...
priority: 120010

targetSecureTags:
- name: tagValues/22222
```

For our second example, we will compare ingress rules allowing an Internet source access to our server. In the following diagrams, the left side source network has been replaced with a cloud, representing the public Internet. Depending on the firewall appliances type, the ingress ACL may be written from the pre-NAT perspective or the post-NAT perspective. However, in Google Cloud, an ingress rule is written from a post-NAT perspective. The same logic holds for any Port Address Translation (PAT). **Since a Cloud Firewall rule is applied to a target, it should always be written from the perspective of the target (including egress rules). For many cases, because the ingress Cloud Firewall rule is distributed based on the target, there is no need to include a destination address. In a cloud environment, where IP addresses are often ephemeral and dynamic, not having a defined static destination address in a firewall rule is often best practice.** If the IP address of the intended target happens to change, or even use a Load Balancer address, the Cloud Firewall rule would still be effective. The rule would be applied to only those resources that match the target for all addresses on those targets. However, if for any reason a destination address was required, an ingress Cloud Firewall rule would be written from the vantage point of the target. This diagram shows a public Internet source attempting to reach the external address of a server on port 80.



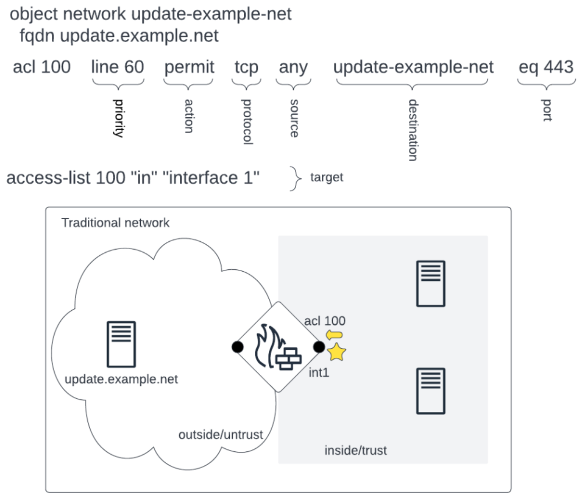
```
acl 101   line 10   permit   tcp   host 198.51.100.10   host 203.0.113.22   eq 80
```



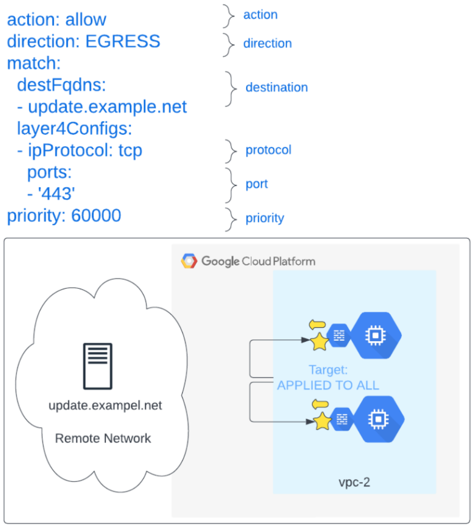
```
action: allow
direction: INGRESS
match:
destIpRanges:
  - 10.1.1.2/32
  layer4Configs:
  - ipProtocol: tcp
    ports:
    - '80'
    srcIpRanges:
  - 198.51.100.10/32
...
priority: 120010
...
targetSecureTags:
- name: tagValues/22222
```

For our third example, we will focus on an egress rule that will allow all resources in our internal network access to update servers for patching. In most firewall appliances there are different sets of rules to secure a perimeter's ingress and egress. For example, there may be one ACL applied at the farside interface, closest to the remote source, for ingress; and then another ACL applied at the nearside interface, closest to the local source, for egress. **This differs from**

Cloud Firewall policies where ingress and egress rules are combined in the same policy. In both examples below, the firewall rules use a Fully Qualified Domain Name (FQDN) to define the destination. One thing to note on this sample Cloud Firewall policy rule is the absence of an explicitly defined target. Because we haven't narrowed the target to a specific tag or service account, this egress rule will apply to all instances in the VPC. Those instances then act as the equivalent of the "source" in a traditional firewall rule.



```
object network update-example-net
 fqdn update.example.net
acl 100   line 60   permit   tcp   any   update-example-net   eq 443
access-list 100 "in" "interface 1"
```



```
action: allow
direction: EGRESS
match:
  destFqdns:
  - update.example.net
  layer4Configs:
  - ipProtocol: tcp
    ports:
    - '443'
priority: 60000
```

Design considerations

Because Google Cloud combines ingress and egress rules in the same policy, one common strategy is to use ranges of the priority numbers to group together egress and ingress rules. Remember, numbers from 0 to 2,147,483,643 can be used. Here is an example:

RESERVE for emergency use	0 - 19999
Egress deny specific rules	20,000-39,999
Egress allow local VPC	40,000-59,999
Egress allow remote private networks	60,000-79,999
Egress allow Internet	80,000-99,999
Egress explicit deny all (for logging)	100,000
RESERVE for emergency use	100,001-119,999

Ingress deny specific rules	120,000-139,999
Ingress allow trusted VPCs	140,000-159,999
Ingress allow remote private networks	160,000-179,999
Ingress allow Internet sources	180,000-199,999
Ingress explicit deny all (for logging)	2,000,000,000

This is just an example. Priority number ranges may vary depending on the number of rules expected for each type. It is recommended to sequence Cloud Firewall rules giving adequate room for rule additions that need to be placed in a very particular spot. Multiple Cloud Firewall rules in the same Cloud Firewall policy **cannot** have the same priority. A single firewall policy can have up to 10,000 rules. A given VPC can only have one firewall policy, but one firewall policy can be associated with many VPCs. There are cases where a single policy associated with multiple VPCs may be easier to manage; however, for enterprise level environments, it is more often recommended to have a 1:1 association between Cloud Firewall policies and VPCs. By attaching a policy to a single VPC, it helps to assume that one VPC does not impact the firewall quotas or limits of other VPCs.

Architecture Considerations

System Design

- Cloud Firewall policy rules should utilize the description field, along with a clear set of priority ranges.
- Periodically review configured rules, looking for multiple overlapping rules that can be condensed into single entities, overly permissive rules and/or rules that may not be needed, etc.
- Consider the multiple layered levels of Google Cloud Firewall: Hierarchical Firewall Policies, global Firewall Policies, VPC firewall rules, regional Firewall Policies. Keeping broad rules (or rules applicable to all instances) at a higher level can help reduce the total number of rules. For more information about Cloud Firewall Policy evaluation order, consult the following documentation here.
- Ensure VMs are tagged in the appropriate way upon creation.

Operational Excellence

- Enabling Firewall Logging to leverage Firewall Insights when appropriate.
  - Firewall Insights helps you identify shadowed rules, overly permissive rules, and rules with no hits.

    - Through machine learning based insights, Firewall Insights enables you to see changes in patterns over time.

Security, Privacy and Compliance

- Google Cloud follows a shared responsibility model, therefore it is imperative to place the proper firewall rules for your organization.
    - This can range from blocking incoming connections from the internet/on-prem/etc to VMs or from VMs to endpoints.
- As always, defense in depth (layered security) is the recommended approach. While Cloud Firewall provides a great layer of security, application level security is a great addition.
    - Other products that can be leveraged are:
        - Cloud NAT to prevent connections initiated from the Internet to reach your VMs.
        - Cloud Armor to protect your GCP resources through security policies.
        - Secure Web Proxy for outbound inspection and filtering.
    - Through IAM controls you can outline who has access to create/modify/delete firewall rules.
        - Ensure only the security, network or the appropriate teams, have permission to operate on the rules.
- Review changes to rules through VPC Audit Logs.

Google Cloud Centric Rules

There are certain Cloud Firewall policy rules that cloud deployments *may* require for correct operation beyond what you have configured on your on-prem firewall. To facilitate a migration to Cloud Firewall, the following list aims to outline supplementary and ancillary Cloud Firewall policy rules that you may require and a brief explanation of them.

Ingress Firewall Rules

- Load Balancer (LB) health check ranges
    - Objective: LB backends are constantly checked for availability through the use of health checks. These checks may not come from an IP range within a VPC subnet; they may arrive from a set of predetermined ranges. For backends to receive traffic they need to respond correctly to said health checks.
    - Target(s): Load Balancer backends
    - Ranges of Interest:
        - All Proxy LBs and Internal Internal LBs*
            - 35.191.0.0/16
            - 130.211.0.0/22

    - External Network Load Balancers
      - 35.191.0.0/16
      - 209.85.152.0/22
      - 209.85.204.0/22
    - Hybrid NEGs with Regional External HTTP(S) LB, Internal HTTP(S) LB, and Internal Regional TCP proxy LB
      - If the project is using `Distributed Envoy Health Checks` Health check traffic will be sourced from the Proxies in the user-defined `proxy-only subnet`.
      - Alternatively
        - 35.191.0.0/16
        - 130.211.0.0/22
- Load Balancer Proxy ranges (external)
  - Objective: Proxy based GCP LBs (HTTP(S) External LB and Regional External HTTP(S) LB) require firewall rules that allow requests from the proxies to backends.
  - Target(s): Proxy Load Balancer backends
  - Ranges of Interest:
    - HTTP(S) LB
      - Global Regular HTTP(S) LB
        - 35.191.0.0/16
        - 130.211.0.0/22
      - Global Classic HTTP(S) LB
        - If backends are instance groups or zonal NEGs
          - 35.191.0.0/16
          - 130.211.0.0/22
        - If backends are Internet NEGs
          - 34.96.0.0/20
          - 34.127.192.0/18
      - Regional External HTTP(S) LB
        - Requests to backends arrive from a user-defined Proxy-only subnet.
    - External TCP Proxy LB & External SSL Proxy LB
      - 35.191.0.0/16
      - 130.211.0.0/22
- Load Balancer Proxy ranges (internal)
  - Objective: Proxy LBs (HTTP(S) Internal LB and Internal Regional TCP Proxy LB), require separate source ranges to be allowed to their backends.
  - Target(s): Internal Proxy Load Balancer backends
  - Ranges of Interest:
    - Internal HTTP(S) LB

            - Requests to backends arrive from a user-defined Proxy-only subnet.
- Private Service Connect (PSC)
    - Objective: PSC allows service producers to publish services to a consumer VPC network. However, requests to the service producer will be sourced using a different range of IP addresses.
    - Target(s): Service Backends.
    - Ranges of Interest:
        - For PSC based on forwarding rules: All requests will come from a user defined PSC NAT subnet associated with the service.
- Identity-Aware Proxy (IAP)
    - Objective: IAP TCP forwarding allows you to establish an encrypted tunnel over which you can forward SSH, RDP, and other traffic to VM instances. IAP leverages a given IP range for these connections; therefore we need to allow ingress traffic from this range to the VMs.
    - Target(s): It is recommended to tag the VMs that will be accessed externally with a specific Tag.
    - Ranges of Interest:
        - 35.235.240.0/20

Egress Firewall Rules

Every VPC network has 2 implied firewall rules. One of these rules is an implied allow all (0.0.0.0/0) IPv4 egress rule with the lowest possible priority. The following egress rules would be recommended if a higher priority egress rule has been configured to deny outbound traffic.

- API addresses
    - Objective: Google offers multiple APIs to users. Traditionally, to access them requests are sent to public IPs. Due to this, access to the traditional Google APIs ranges may need to be explicitly allowed.
    - Target(s): Instances that will access Google APIs with Public IPs.
    - Ranges of Interest:
        - Frontend IPs for APIs may change. Due to this, it is recommended to allow egress to FQDNs for APIs that are being leveraged.
- Private Google API (PGA) ranges
    - Objective: PGA allows VMs (with or without public IPs) to reach a smaller set of IP ranges to access Google Cloud APIs.
    - Target(s): Instances that are expected to reach out to Google API services.
    - Ranges of Interest:
        - Private Google APIs
            - 199.36.153.8/30
        - Restricted Google APIs
            - 199.36.153.4/30

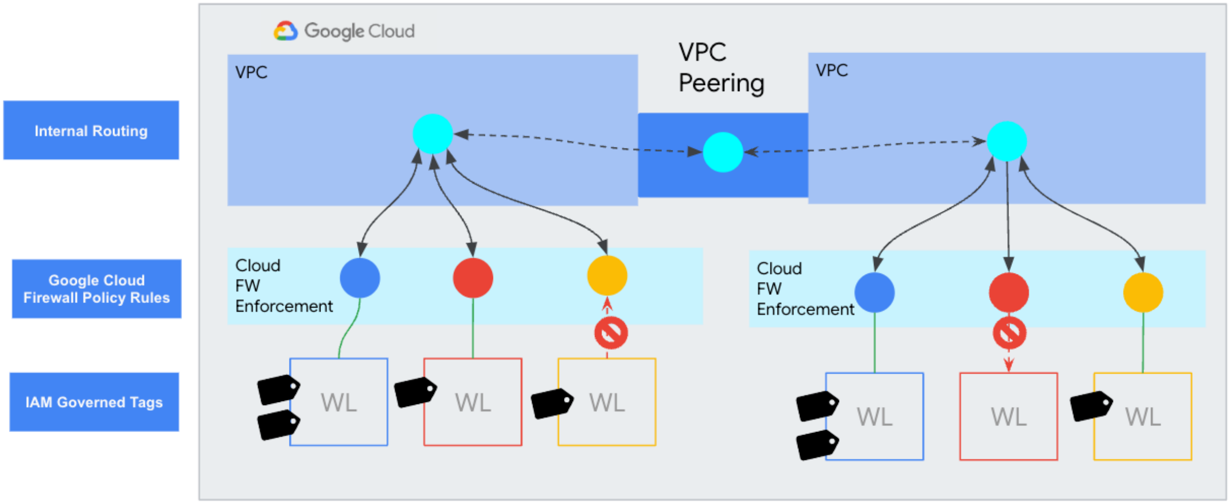
IAM Governed Tags

IAM-governed Tags are securable key-value pairs, defined within an organization, with IAM controls (allowing fine grained control). When used along with Cloud Firewall policies, they are associated with a VPC and attached to VMs on a per NIC basis.

Example:

- KEY: vm-type
    - VALUE: mysql
    - VALUE: appserver
    - VALUE: lbbackend
    - VALUE: bastion

When creating rules for a Cloud Firewall Policy you can leverage tags to identify the source or target in a given rule.



Deployment

The following section provides examples of how you may create a blank firewall policy and add some of the Google Cloud centric rules covered in the previous section.

Create a new global network firewall policy

```
gcloud compute network-firewall-policies create [FW_POLICY_NAME] \
--description [DESCRIPTION] \
--global
```

Create Google Cloud centric ingress firewall policy rules

1. Create ingress Rule for Load Balancer health checks

```
gcloud compute network-firewall-policies rules create 140010 \
--description "Allow Google Cloud Health Checks" \
--direction INGRESS \
--action allow \
--layer4-configs tcp:[HC_PORT_RANGE] \
--firewall-policy [FW_POLICY_NAME] \
--src-ip-ranges 35.191.0.0/16,130.211.0.0/22,209.85.152.0/22,209.85.204.0/22 \
--target-secure-tags tagValues/[TAG_VALUE] \
--global-firewall-policy
```

2. Create ingress rule for global proxy Load Balancer proxy ranges

   This sample rule references the currently used IP ranges for Google Cloud Global Load Balancer proxy ranges. The rule is written targeting ALL VMs, however, IAM governed tags can be used to narrow the targets.

```
gcloud compute network-firewall-policies rules create 140020 \
--description "Allow Global LB Proxy Ranges" \
--direction INGRESS \
--action allow \
--layer4-configs tcp:[APP_PORT_RANGE] \
--firewall-policy [FW_POLICY_NAME] \
--src-ip-ranges 35.191.0.0/16,130.211.0.0/22 \
--target-secure-tags tagValues/[TAG_VALUE] \
--global-firewall-policy
```

3. Create ingress rule for Internal Load Balancer proxy ranges, assuming TCP based application

```
gcloud compute network-firewall-policies rules create 140030 \
--description "Allow INTERNAL LB Proxy Ranges" \
--direction INGRESS \
--action allow \
--layer4-configs tcp:[APP_PORT_RANGE] \
--firewall-policy [FW_POLICY_NAME] \
--src-ip-ranges [INTERNAL_PROXY_RANGES] \
--target-secure-tags tagValues/[TAG_VALUE] \
--global-firewall-policy
```

4. Create ingress rule for PSC proxy ranges, assuming TCP based application

```
gcloud compute network-firewall-policies rules create 140040 \
--description "Allow PSC Proxy Ranges" \
--direction INGRESS \
--action allow \
--layer4-configs tcp:[APP_PORT_RANGE] \
--firewall-policy [FW_POLICY_NAME] \
--src-ip-ranges [PSC_PROXY_RANGES] \
--global-firewall-policy
```

5. Create ingress rule for IAP ranges

```
gcloud compute network-firewall-policies rules create 140050 \
--description "Allow GCP IAP Range" \
--direction INGRESS \
--action allow \
--layer4-configs tcp:22 \
--firewall-policy [FW_POLICY_NAME] \
--src-ip-ranges 35.235.240.0/20 \
--target-secure-tags tagValues/[TAG_VALUE] \
--global-firewall-policy
```

Create EGRESS firewall policy rules

1. Create egress rule for Private Google Access (PGA) API ranges

```
gcloud compute network-firewall-policies rules create 60010 \
--description "Allow PGA API Ranges" \
--direction EGRESS \
--action allow \
--layer4-configs tcp:80,tcp:443 \
--firewall-policy [FW_POLICY_NAME] \
--dest-ip-ranges 199.36.153.8/30,199.36.153.4/30 \
--global-firewall-policy
```

2. Create egress rule for Public Google API ranges

```
gcloud compute network-firewall-policies rules create 60020 \
--description "Allow GCP public API Ranges" \
--direction EGRESS \
--action allow \
--layer4-configs tcp:80,tcp:443 \
--firewall-policy [FW_POLICY_NAME] \
--dest-ip-ranges [API_RANGES] \
--global-firewall-policy
```

Evaluate and Apply the policy to your VPC

1. Review the Cloud Firewall policy and edit as necessary

   Take time and review the policy thoroughly with network and security teams.

```
gcloud compute network-firewall-policies describe [FW_POLICY_NAME] --global
```

2. Apply Cloud Firewall policy to a VPC

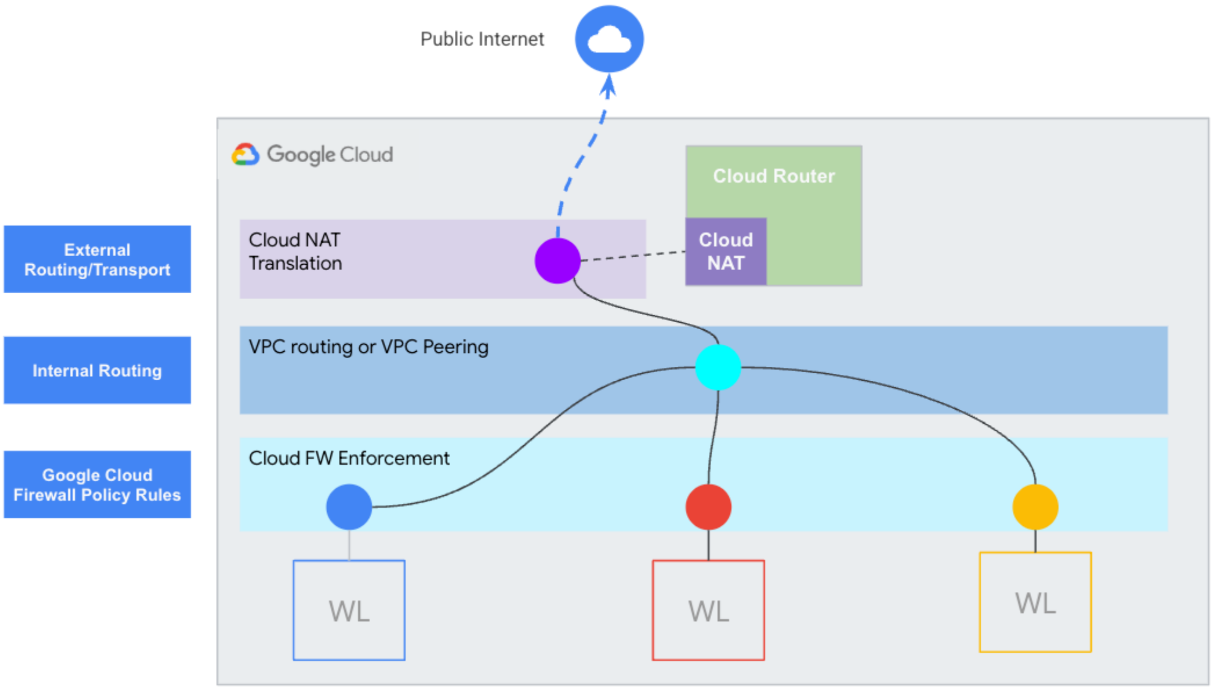
   When ready, apply the Cloud Firewall Policy to a VPC.

```
gcloud compute network-firewall-policies associations create \
--firewall-policy [FW_POLICY_NAME] \
--network [VPC] \
--global-firewall-policy
```

Beyond Cloud Firewall

On-prem firewall appliances often perform additional network and security services beyond advanced layer 3/4 filtering. For example, firewall appliances are often used as Internet egress proxies for a network or VPN concentrators. To migrate these functions into Google Cloud, look beyond Cloud Firewall and into additional Google Cloud services such as Cloud NAT and Cloud VPN.

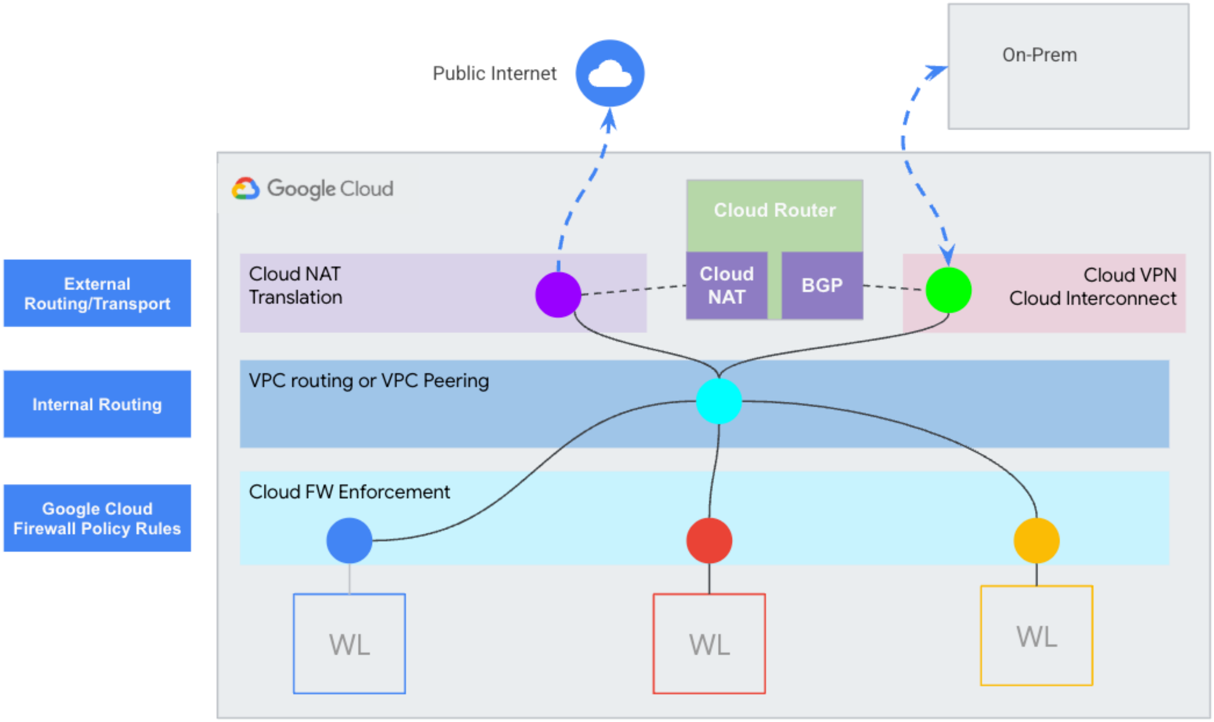
Cloud NAT and Cloud Firewall are very often used in conjunction to facilitate and secure Internet egress traffic. Egress Cloud Firewall Policy rules are first applied at the source resource level. If allowed, traffic may then be processed by Cloud NAT and routed to the Internet. Much like Cloud Firewall, Cloud NAT is a fully distributed Source NAT solution that has no network appliance as a bottleneck.



Cloud NAT offers two options for IP addresses: “automatic" IP allocation and “manual” IPs. The difference is simply that automatic allocation dynamically adds and removes public IP addresses as needed, whereas in manual IP allocation you allocate reserved IPs for Cloud NAT to use. If you require to access a public resource that requires allowlisting, using manual IP allocation with an appropriate number of public addresses is recommended. If, however, no allowlisting is needed, using automatic IP allocation is recommended. Migration from one to the other is also possible. So, if you are unsure of which you may need, start with “automatic” (you can create a custom dashboard to monitor the metric `nat_ip` to better determine IP usage), knowing that a migration to “manual” is possible.

A second recommendation for Cloud NAT is to enable “Dynamic Port Allocation” (DPA). With the traditional “Static” port allocation, each VM that uses the Cloud NAT is assigned the same number of ports, regardless of their usage. However, it is likely that not all VMs using Cloud NAT require the exact same number of ports. With DPA, Cloud NAT continually monitors port usage and can allocate (up to a maximum you can configure) or deallocate (down to a minimum you configure) VM NAT ports as needed. This generally results in more efficient use of Cloud NAT IP addresses and less chances for Cloud NAT port exhaustion when using a Manual IP address pool. To help ensure that the gateway has a sufficient number of NAT IP addresses, always consider the number of VMS and how many connections to a unique destination “*protocol:IP:port*” are required. Then configure the Cloud NAT port settings appropriately.

Cloud Firewall does not serve any inter-network routing, encryption or encapsulation functions that your on-prem firewall may perform. The onus for these responsibilities would fall on products like Cloud VPN, VPC Peering and Network Connectivity Center. Use Cloud Firewall policies jointly with all these offerings. For example, you may interconnect two VPCs in your organization with VPC peering, which would exchange all subnet routes between the peered VPCs. Then implement Cloud Firewall policies to narrow the allowed scope of connectivity.



What’s Next:

Learn more about the Google Cloud products and how they may align with your firewall appliance features:

FW appliances Feature	GCP Product Equivalent
Standard L3/4 filtering	Cloud Firewall Essentials
Layer 7 HTTP based filtering and security	Cloud Armor Secure Web Proxy
FQDN sources/destinations	Cloud Firewall Standard FQDN
Geo Based sources/destinations	Cloud Firewall Standard geo-location
Object Groups	Cloud Firewall Essentials Address Groups
Packet simulation tests	Network Intelligence Center (NIC) Connectivity Tests
Packet Captures	Packet Mirroring
VPN	Cloud VPN
Address Translation (NAT)	Cloud NAT
Logging	Cloud Logging
Outbound Layer 7 Web Inspection	Secure Web Proxy

FAQ:

Are Google Cloud Cloud Firewall rules stateful or stateless?
Google Cloud Cloud Firewall Rules are stateful. There is no need to create rules to allow return or response traffic.

Do I need to configure my Cloud Firewall Rules to allow Google Cloud DNS or DHCP services?
No. This traffic is allowed via a metadata service. There is no need to account for access to Google Cloud DNS or DHCP services. However, additional rules may be required if you are trying to allow a VM to reach a non-Google Cloud Native DNS/DHCP service.

Do Cloud Firewalls support IPv6?
Yes. IPv6 is supported for VPCs with IPv6 enabled.

Can Cloud Firewall filter egress traffic based on HTTP Host Header or URL?
No. That is not something that Cloud Firewall can do. Cloud Firewall Standard does offer FQDN based rules, but that is DNS resolution based, not HTTP based.

Do I need to create Cloud Firewall rules to allow communication between VMs in the same subnet?
Yes. Unlike traditional firewalls which protect one network segment from another, Cloud Firewall policy rules include an implicit deny ingress rule even for resources in the same VPC subnet.

Can Cloud Firewall be used to filter traffic between resources in the same layer 3 network or VPC like a transparent firewall can?
Cloud Firewall is fully distributed. Thus, it can be used to filter traffic between resources in the same layer 3 network or VPC much like a transparent firewall can.

Can an address group be used in multiple rules?
Yes, a single address group can be used in multiple rules in the same policy and even in multiple policies. They can be used as sources for ingress rules and as destinations in egress rules.

Are Cloud Firewall Rules logged?
Google Cloud Firewall rules logging can be enabled at a per-rule basis. The default implicit rules are not logged. Firewall logging is encouraged to enable forensics into your firewall set-up, additionally logging allows you to leverage Firewall Insights from Network Connectivity Center, which will help you identify shadowed/overly permissive rules. Please note that firewall

logging is billed based on log generation, so enabling logging on rules may result in increased cost.