Google Cloud firewall rules let you allow or deny traffic to and from your virtual machine (VM) instances based on a configuration that you specify. Enabled Google Cloud firewall rules are always enforced, protecting your instances regardless of their configuration and operating system, even if they have not started up.

Every Virtual Private Cloud (VPC) network functions as a distributed firewall. While firewall rules are defined at the network level, connections are allowed or denied on a per-instance basis. You can think of the Google Cloud firewall rules as existing not only between your instances and other networks, but also between individual instances within the same network.

This page is an overview of firewall rules. If you're looking for information about how to create and work with firewall
ing firewall rules (/vpc/docs/using-firewalls).

For more information about firewalls, see Firewall (computing) (https://wikipedia.org/wiki/Firewall_(computing)).

When you create a Google Cloud firewall rule, you specify a VPC network and a set of components that define what the rule does. The components enable you to target certain types of traffic, based on the traffic's protocol, ports, sources, and destinations. For more information, see firewall rule components (#firewall_rule_components).

You create or modify Google Cloud firewall rules by using the Google Cloud Console (https://console.cloud.google.com/), `gcloud` command-line tool (/sdk/gcloud/reference/compute/firewall-rules/), and REST API (/compute/docs/reference/v1/firewalls). When you create or modify a firewall rule, you can specify the instances to which it is intended to apply by using the target component (#rule_assignment) of the rule.

In addition to firewall rules that you create, Google Cloud has other rules that can affect incoming (ingress) or outgoing (egress) traffic:

- Google Cloud doesn't allow certain IP protocols, such as GRE (https://wikipedia.org/wiki/Generic_Routing_Encapsulation), within a VPC network. For more information, see always blocked traffic (#blockedtraffic).

- Google Cloud always allows communication between a VM instance and its corresponding metadata server at `169.254.169.254`. For more information, see always allowed traffic (#alwaysallowed).
- Every network has two implied firewall rules (#default_firewall_rules) that permit outgoing connections and block incoming connections. Firewall rules that you create can override these implied rules.
- The default network is pre-populated with firewall rules (#more_rules_default_vpc) that you can delete or modify.

Google Cloud firewall rules have the following characteristics:

- Each firewall rule applies to incoming (ingress) or outgoing (egress) traffic, not both. For more information, see direction of traffic (#direction_of_the_rule).
- Firewall rules only support IPv4 traffic. When specifying a source for an ingress rule or a destination for an egress rule by address, you can only use an IPv4 address or IPv4 block in CIDR notation.
- Each firewall rule's action is either `allow` or `deny` (#action_of_the_rule). The rule applies to traffic as long as it is enforced (#enforcement). For example, you can disable a rule for troubleshooting purposes.
- When you create a firewall rule, you must select a VPC network. While the rule is enforced at the instance level, its configuration is associated with a VPC network. This means that you cannot share firewall rules among VPC networks, including networks connected by VPC Network Peering (/vpc/docs/vpc-peering) or by using Cloud VPN tunnels (/vpn/docs/concepts/choosing-networks-routing#dynamic-routing).
- Google Cloud firewall rules are stateful (https://wikipedia.org/wiki/Stateful_firewall). After a session has been established, firewall rules allow bidirectional communication. You cannot configure a firewall rule to deny associated response traffic. Google Cloud associates incoming packets with corresponding outbound packets by using a connection tracking table. Google Cloud implements connection tracking regardless if the protocol supports connections. If a connection is allowed between a source and a target (for an ingress rule) or between a target and a destination (for an egress rule), all response traffic is allowed as long as the firewall's connection tracking state is active. A firewall rule's tracking state is considered active if at least one packet is sent every 10 minutes.

- Google Cloud firewall rules do not reassemble fragmented TCP packets. Therefore, a firewall rule applicable to the TCP protocol can only apply to the first fragment because it contains the TCP header. Firewall rules applicable to the TCP protocol do not apply to the subsequent TCP fragments.

- The maximum number of tracked connections in the firewall rule table depends on the number of stateful connections supported by the machine type of the instance.

Instance machine type	Maximum number of stateful connections
Shared-core machine types (/compute/docs/machine-types#sharedcore)	130,000
Instances with 1–8 vCPUs	130,000 connections per vCPU
Instances with more than 8 vCPUs	1,040,000 (130,000×8) connections total

Every VPC network has two implied firewall rules. These rules exist, but are not shown in the Cloud Console:

- **Implied allow egress rule.** An egress rule whose action is `allow`, destination is `0.0.0.0/0`, and priority is the lowest possible (`65535`) lets any instance send traffic to any destination, except for traffic blocked (#blockedtraffic) by Google Cloud. A higher priority firewall rule may restrict outbound access. Internet access is allowed if no other firewall rules deny outbound traffic and if the instance has an external IP address or uses a Cloud NAT instance. For more information, see Internet access requirements (/vpc/docs/vpc#internet_access_reqs).

- **Implied deny ingress rule.** An ingress rule whose action is `deny`, source is `0.0.0.0/0`, and priority is the lowest possible (`65535`) protects all instances by blocking incoming traffic to them. A higher priority rule might allow incoming access. The default network includes some additional rules (#more_rules_default_vpc) that override this one, allowing certain types of incoming traffic.

The implied rules *cannot* be removed, but they have the lowest possible priorities. You can create rules that override them as long as your rules have higher priorities (priority numbers *less than* `65535`). Because `deny` rules take precedence over `allow` rules of the same priority, an ingress `allow` rule with a priority of `65535` never takes effect.

Implied firewall rules are present in all VPC networks, regardless of how the networks are created, and whether they a mode or custom mode VPC networks (/vpc/docs/vpc#subnet-ranges). The default network has the same implied rule

The default network is pre-populated with firewall rules that allow incoming traffic to instances. These rules can be deleted or modified as necessary:

- `default-allow-internal`
  Allows ingress connections for all protocols and ports among instances in the network. This rule has the second-to-lowest priority of `65534`, and it effectively permits incoming connections to VM instances from others in the same network.
- `default-allow-ssh`
  Allows ingress connections on TCP port 22 from any source to any instance in the network. This rule has a priority of `65534`.
- `default-allow-rdp`
  Allows ingress connections on TCP port 3389 from any source to any instance in the network. This rule has a priority of `65534`, and it enables connections to instances running the Microsoft Remote Desktop Protocol (RDP).
- `default-allow-icmp`
  Allows ingress ICMP traffic from any source to any instance in the network. This rule has a priority of `65534`, and it enables tools such as `ping`.

These rules are included in the default network, but you can create your own rules that allow these types of traffic in y networks.

Google Cloud always blocks the traffic that is described in the following table. Your firewall rules *cannot* be used to allow any of this traffic.

Always blocked traffic	Applies to
GRE traffic	All sources, all destinations, including among instances that use internal IP addresses

Protocols other than TCP, UDP, ICMP, and IPIP	Traffic between: • instances and the internet • instances if they are addressed by external IP addresses • instances if a load balancer with an external IP address is involved
Egress traffic on TCP port 25 (SMTP)	Traffic from: • instances to the internet • instances to other instances addressed by external IP addresses

Google Cloud runs a local metadata server alongside each instance at `169.254.169.254`. This server is essential to the operation of the instance, so the instance can access it regardless of any firewall rules that you configure. The metadata server provides the following basic services to the instance:

- DHCP (https://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol)
- DNS resolution, following the name resolution order (/dns/docs/overview#vpc-name-resolution-order) for the VPC network. Unless you have configured an alternative name server, DNS resolution includes looking up Compute Engine internal DNS (/compute/docs/internal-dns) and querying Cloud DNS zones (/dns/docs/overview#concepts) and public DNS names.
- Instance metadata (/compute/docs/storing-retrieving-metadata)
- Network Time Protocol (NTP) (https://en.wikipedia.org/wiki/Network_Time_Protocol)

Firewall rules cannot block traffic that an instance sends to one of its own IP addresses because that traffic never lea
M itself. These addresses include its primary internal IP address, any alias IP address, and loopback addresses. Also, i
ce participates as a backend for an internal load balancer, the load balancer's IP address is also assigned to it.

Each firewall rule consists of the following configuration components:

- The direction of traffic (#direction_of_the_rule): ingress rules apply to incoming connections from specified *sources* to Google Cloud *targets*, and egress rules apply to traffic going to specified *destinations* from *targets*.
- A numerical priority (#priority_order_for_firewall_rules), which determines whether the rule is applied. Only the highest priority (lowest priority number) rule whose other components match

traffic is applied; conflicting rules with lower priorities are ignored.

- An action on match (#action_of_the_rule), either **allow** or **deny**, which determines whether the rule permits or blocks traffic.
- The enforcement status (#enforcement) of the firewall rule: You can enable and disable firewall rules without deleting them.
- A target (#rule_assignment), which defines the instances (including GKE clusters and App Engine flexible environment instances) to which the rule applies.
- A source (#sources_or_destinations_for_the_rule) for ingress rules or a destination for egress rules.
- The protocol (#protocols_and_ports) (such as TCP, UDP, or ICMP) and port.

Ingress (inbound) rule

Priority	Action	Enforcement	Target (defines the destination)	Source	Protocols and ports
Integer from **0** to **65535**, inclusive; default **1000**	**allow** or **deny**	**enabled** (default) or **disabled**	The *target* parameter specifies the destination; it can be one of the following: • All instances in the VPC network • Instances by network tag (/vpc/docs/add-remove-network-tags) • Instances by service account (#serviceaccounts)	One of the following: • Range of IPv4 addresses; default is any (**0.0.0.0/0**) • Instances by network tag • Instances by service account	Specify a protocol or a protocol and a port. If not set, the rule applies to all protocols.

Egress (outbound) rule

Priority	Action	Enforcement	Target (defines the source)	Destination	Protocols and ports
Integer from **0** to **65535**, inclusive; default **1000**	**allow** or **deny**	**enabled** (default) or **disabled**	The *target* parameter specifies the source; it can be one of the following: • All instances in the VPC network • Instances by network tag	Any network or a specific range of IPv4 addresses; default is any (**0.0.0.0/0**)	Specify a protocol or a protocol and a port. If not set, the rule applies to all protocols.

- Instances by service account

The direction of a firewall rule can be either ingress or egress. The direction is always defined from the perspective of the target.

- The ingress direction describes traffic sent from a source to a target. Ingress rules apply to packets for new sessions where the destination of the packet is the target.
- The egress direction describes traffic sent from a target to a destination. Egress rules apply to packets for new sessions where the source of the packet is the target.
- If you don't specify a direction, Google Cloud uses ingress.

Consider an example connection between two VMs in the same network. Traffic from VM1 to VM2 can be controlled by using either of these firewall rules:

- An ingress rule with a target of VM2 and a source of VM1.
- An egress rule with a target of VM1 and a destination of VM2.

The firewall rule priority is an integer from `0` to `65535`, inclusive. *Lower integers indicate higher priorities*. If you do not specify a priority when creating a rule, it is assigned a priority of `1000`.

The relative priority of a firewall rule determines whether it is applicable when evaluated against others. The evaluation logic works as follows:

- The highest priority rule applicable to a target for a given type of traffic takes precedence. Target specificity does not matter. For example, a higher priority ingress rule for certain ports and protocols intended for all targets overrides a similarly defined rule for the same ports and protocols intended for specific targets.
- The highest priority rule applicable for a given protocol and port definition takes precedence, even when the protocol and port definition is more general. For example, a higher priority ingress rule allowing traffic for all protocols and ports intended for given targets overrides a lower priority ingress rule denying TCP 22 for the same targets.
- A rule with a `deny` action overrides another with an `allow` action *only if the two rules have the same priority*. Using relative priorities, it is possible to build `allow` rules that override `deny` rules,

and `deny` rules that override `allow` rules.

- Rules with the same priority and the same action have the same result. However, the rule that is used during the evaluation is indeterminate. Normally, it doesn't matter which rule is used except when you enable Firewall Rules Logging (/vpc/docs/firewall-rules-logging). If you want your logs to show firewall rules being evaluated in a consistent and well- defined order, assign them unique priorities.

Consider the following example where two firewall rules exist:

- An ingress rule from sources `0.0.0.0/0` (anywhere) applicable to all targets, all protocols, and all ports, having a `deny` action and a priority of `1000`.
- An ingress rule from sources `0.0.0.0/0` (anywhere) applicable to specific targets with the tag `webserver`, for traffic on TCP 80, with an `allow` action.

The priority of the second rule determines whether TCP traffic on port 80 is allowed for the `webserver` targets:

- If the priority of the second rule is set to a number *greater than* `1000`, it has a *lower* priority, so the first rule denying all traffic applies.
- If the priority of the second rule is set to `1000`, the two rules have identical priorities, so the first rule denying all traffic applies.
- If the priority of the second rule is set to a number *less than* `1000`, it has a *higher* priority, thus allowing traffic on TCP 80 for the `webserver` targets. Absent other rules, the first rule would still deny other types of traffic to the `webserver` targets, and it would also deny all traffic, including TCP 80, to instances *without* the `webserver` tag.

The previous example demonstrates how you can use priorities to create selective `allow` rules and global `deny` rules to implement a security best practice of least privilege.

Every network has two non-removable, low-priority, implied firewall rules, and the default network comes with additio able firewall rules. For more information, see default and implied rules (#default_firewall_rules).

The action component of a firewall rule determines whether it permits or blocks traffic, subject to the other components of the rule:

- An `allow` action permits connections that match the other specified components.

- A `deny` action blocks connections that match the other specified components.

A firewall rule can only have one action component. Both `allow` and `deny` *cannot* be specified in the same rule. To de
der in which the rules should be applied, create separate firewall rules with different priorities.

You can change whether a firewall rule is *enforced* by setting its state to `enabled` or `disabled`. Disabling a rule is useful for troubleshooting or to grant temporary access to instances. It's much easier to disable a rule, test, and then re-enable it, than it is to delete and re-create the rule.

Unless you specify otherwise, all firewall rules are `enabled` when they are created. You can also choose to create a rule (/vpc/docs/using-firewalls#creating_firewall_rules) in a `disabled` state.

The enforcement state for firewall rules can be changed from `enabled` to `disabled` and back by updating the rule (/vpc/docs/using-firewalls#updating_firewall_rules).

Consider disabling a firewall rule for situations like these:

- For troubleshooting: If you're not sure whether a firewall rule is blocking or allowing traffic, disable it temporarily to determine if traffic is allowed or blocked. This is useful to troubleshoot the effect of one rule in conjunction with others.
- For maintenance: Disabling firewall rules can make periodic maintenance simpler. Suppose you have a firewall rule that blocks incoming SSH to targets (for example, instances by target tag), and that rule is `enabled`. When you need to perform maintenance, you can disable the rule. After you finish, enable the rule again.

For an ingress (inbound) rule, the *target* parameter designates the destination VM instances, including GKE clusters and App Engine flexible environment instances. For an egress (outbound) rule, the target designates the source instances. Thus, always use the *target* parameter to designate Google Cloud instances, but whether a target is a destination of traffic or a source for traffic depends on the direction of the rule.

You specify a target by using one of the following options:

- *All instances in the network*. The firewall rule applies to all instances in the network.

- *Instances by target tags*. The firewall rule applies only to instances with a matching network tag (/vpc/docs/add-remove-network-tags).
- *Instances by target service accounts*. The firewall rule applies only to instances that use a specific service account (#serviceaccounts). For the maximum number of target service accounts that you can apply per firewall rule, see VPC resource quotas (/vpc/docs/quota#per_network).

For information about the benefits and limitations of target tags and target service accounts, see filtering by service account versus network tag (#service-accounts-vs-tags).

The target of an ingress firewall rule applies to all traffic *arriving* on an instance's network interface in the VPC network, regardless of how the target is specified. An ingress firewall rule takes effect on packets whose destination matches one of the following IP addresses:

- The primary internal IP address assigned to the instance's network interface in the VPC network.
- Any configured alias IP ranges (/vpc/docs/alias-ip) on the instance's network interface in the VPC network.
- The external IP address that's associated with the instance's network interface in the VPC network.
- A Google Cloud load balancer if the instance is a backend of the load balancer.

The target of an egress firewall rule applies to all traffic *leaving* an instance's network interface in the VPC network, regardless of how the target is specified. An egress firewall rule takes effect on packets whose source matches either the primary internal IP address or any configured alias IP ranges on the instance's network interface in the VPC network.

You specify *either* a source or a destination, but not both, depending on the direction of the firewall that you create:

- For ingress (inbound) rules, the *target* parameter specifies the destination instances for traffic; you cannot use the *destination* parameter. You specify the source by using the *source* parameter.
- For egress (outbound) rules, the *target* parameter specifies the source instances for traffic; you cannot use the *source* parameter. You specify the destination by using the *destination*

parameter.

The *source* parameter is only applicable to ingress rules. It must be one of the following:

- *Source IP ranges*: You can specify ranges of IP addresses as sources for packets. The ranges can include addresses inside your VPC network and addresses outside it. Source IP ranges can be used to define sources both inside and outside Google Cloud.
- *Source tags*: You can define the source for packets as the primary internal IP address of the network interface of VM instances in the same VPC network, identifying those source instances by a matching network tag (/vpc/docs/add-remove-network-tags). Source tags *only* apply to traffic sent from the network interface of another applicable instance in your VPC network. A source tag *cannot* control packets whose sources are external IP addresses, even if the external IP addresses belong to instances. For the maximum number of source tags that you can apply per firewall rule, see VPC resource quotas (/vpc/docs/quota#per_network).
- *Source service accounts*: You can define the source for packets as the primary internal IP address of the network interface of instances in the same VPC network, identifying those source instances by the service accounts (#serviceaccounts) they use. Source service accounts *only* apply to traffic sent from the network interface of another applicable instance in your VPC network. A source service account *cannot* control packets whose sources are external IP addresses, even if the external IP addresses belong to instances. For the maximum number of source service accounts that you can apply per firewall rule, see VPC resource quotas (/vpc/docs/quota#per_network).
- A combination of *source IP ranges* and *source tags* can be used.
- A combination of *source IP ranges* and *source service accounts* can be used.
- If all *source IP ranges*, *source tags*, and *source service accounts* are omitted, Google Cloud defines the source as any IP address (`0.0.0.0/0`).

tant: Network tags and service accounts cannot be used in the same firewall rule. For more information, see filtering e account versus network tag (#service-accounts-vs-tags).

The *destination* parameter is only applicable to egress rules. The *destination* parameter only accepts IP address ranges. The ranges can include addresses inside your VPC network and addresses outside it.

If you do not specify a destination range, Google Cloud defines the destination to be all IP addresses (`0.0.0.0/0`).

You can narrow the scope of a firewall rule by specifying protocols or protocols and ports. You can specify a protocol or a combination of protocols and their ports. If you omit both protocols and ports, the firewall rule is applicable for all traffic on any protocol and any port.

To make a firewall rule specific, you must first specify a protocol. If the protocol supports ports, you can optionally specify a port number or port range. Not all protocols support ports, though. For example, ports exist for TCP and UDP, but not for ICMP. (ICMP does have different *ICMP types*, but they are not ports.)

You can specify a protocol by using its name (`tcp`, `udp`, `icmp`, `esp`, `ah`, `sctp`, `ipip`) or its decimal IP protocol number (https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml).

Google Cloud firewall rules use port information to reference the *destination port of a packet*, not its source port:

- For ingress (inbound) firewall rules, destination ports are ports on systems identified by the rule's *target* parameter. (For ingress rules, the *target* parameter specifies the destination VMs for traffic (#sources_or_destinations_for_the_rule).)
- For egress (outbound) firewall rules, destination ports represent ports on the systems identified by the rule's *destination* parameter.

The following table summarizes valid protocol and port specification combinations for Google Cloud firewall rules.

Specification	Example	Explanation
No protocol and port	—	If you do not specify a protocol, the firewall rule applies to all protocols and their applicable ports.
Protocol	`tcp`	If you specify a protocol without any port information, the firewall rule applies to that protocol and all its applicable ports.
Protocol and single port	`tcp:80`	If you specify a protocol and a single port, the firewall rule applies to that port of the protocol.
Protocol and port range	`tcp:20-22`	If you specify a protocol and a port range, the firewall rule applies to that port range for the protocol.

Combinations	`icmp,`
	`tcp:80`
	`tcp:443`
	`udp:67-69`
		You can specify various combinations of protocols and ports to which the firewall rule applies. For more information, see creating firewall rules (/vpc/docs/using-firewalls#creating_firewall_rules).

tant: A port cannot be specified by itself. If you only specify a number, Google Cloud interprets that as a decimal IP *col*. For example, if you specify **`80`** by itself, Google Cloud interprets that as IP protocol **`80`** (ISO-IP), which is not the sa *ort* 80 (**`tcp:80`**).

You can use service accounts (/iam/docs/service-accounts) to create firewall rules that are more specific in nature:

- For both ingress and egress rules, you can use service accounts to specify targets.
- For ingress rules, you can specify the source for incoming packets as the primary internal IP address of any VM in the network where the VM uses a particular service account.

The service account must be created (/iam/docs/creating-managing-service-accounts) *before* you create a firewall rule that relies on it.

Firewall rules that use service accounts to identify instances apply to both new instances created and associated with the service account (/compute/docs/access/create-enable-service-accounts-for-instances#createanewserviceaccount) and existing instances if you change their service accounts (/compute/docs/access/create-enable-service-accounts-for-instances#changeserviceaccountandscopes). Changing the service account associated with an instance requires that you stop and restart it. You can associate service accounts with individual instances and with instance templates used by managed instance groups (/compute/docs/instance-groups).

You can use service accounts from the same project where the firewall rule is defined or from other projects, such as d VPC (/vpc/docs/shared-vpc) host projects. However, you cannot use service accounts that come from other projec e VPC Network Peering to connect to networks in those other projects.

This section highlights key points to consider when deciding if you should use service accounts or network tags to define targets and sources (for ingress rules).

If you need strict control over how firewall rules are applied to VMs, use target service accounts and source service accounts instead of target tags and source tags:

- *A network tag is an arbitrary attribute.* One or more network tags can be associated (/vpc/docs/add-remove-network-tags) with an instance by any Cloud Identity and Access Management (Cloud IAM) member who has permission to edit it. Cloud IAM members with the Compute Engine Instance Admin (/iam/docs/understanding-roles#compute-engine-roles) role to a project have this permission. Cloud IAM members who can edit an instance can change its network tags, which could change the set of applicable firewall rules for that instance.
- *A service account represents an identity associated with an instance.* Only one service account can be associated with an instance. You control access to the service account by controlling the grant of the Service Account User (/iam/docs/understanding-roles#service-accounts-roles) role for other Cloud IAM members. For a Cloud IAM member to start an instance by using a service account, that member must have the Service Account User role to at least that service account and appropriate permissions to create instances (for example, having the Compute Engine Instance Admin role to the project).

You cannot mix and match service accounts and network tags in any firewall rule:

- You cannot use target service accounts and target tags together in any firewall rule (ingress or egress).
- If you specify targets by target tag or target service account, the following are invalid sources for ingress firewall rules.

Targets	Invalid sources
Target tags	Source service accounts Combination of source IP ranges and source service accounts
Target service account	Source tags Combination of source IP ranges and source tags

Following are operational considerations for service accounts and network tags:

- Changing a service account for an instance requires stopping and restarting it. Adding or removing tags can be done while the instance is running.

- There are a maximum number of target service accounts, source service accounts, target network tags, and source network tags that can be specified for firewall rules. For more information, see VPC resource quotas (/vpc/docs/quota#per_network).
- If you identify instances by network tag, the firewall rule applies to the primary internal IP address of the instance.

The following use cases demonstrate how firewall rules work. In these examples, all the firewall rules are enabled.

Ingress firewall rules control incoming connections from a source to target instances in your VPC network. The source for an ingress rule can be defined as one of the following:

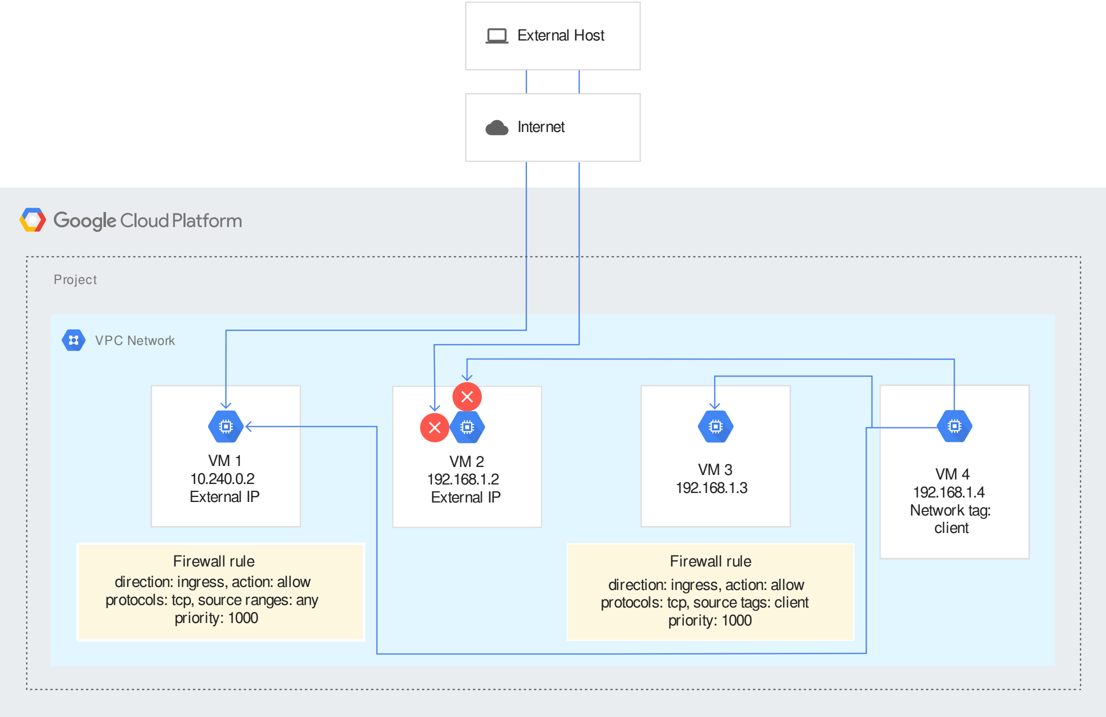
- A range of IPv4 addresses; the default is any (`0.0.0.0/0`)
- Other instances in your VPC network identified by service account
- Other instances in your VPC network identified by network tags

The default source is any IP address (`0.0.0.0/0`). If you want to control incoming connections for sources outside your VPC network, including other sources on the internet, use a range of IPv4 addresses in CIDR format.

Ingress rules with an `allow` action permit incoming traffic based on the other components of the rule (#firewall_rule_components). In addition to specifying the source and target for the rule, you can limit the rule to apply to specific protocols and ports. Similarly, ingress rules with a `deny` action can be used to protect instances by blocking incoming traffic based on the firewall rule components.

n: You can also use target service accounts or target tags to specify the ingress destinations. If you do that, you lim n specify the source for the rule. For more information, see filtering by service account versus network tag vice-accounts-vs-tags).

The following diagram illustrates some examples where firewall rules can control ingress connections. The examples use the *target* parameter in rule assignments to apply rules to specific instances.



(/vpc/images/firewalls/firewall_overview_ingress_examples.svg)
Ingress firewall rules example (click to enlarge)

- An ingress rule with priority `1000` is applicable to **VM 1**. This rule allows incoming TCP traffic from any source (`0.0.0.0/0`). TCP traffic from other instances in the VPC network is allowed, subject to applicable egress rules for those other instances. **VM 4** is able to communicate with **VM 1** over TCP because **VM 4** has no egress rule blocking such communication (only the implied allow egress rule is applicable). Because **VM 1** has an external IP, this rule also permits incoming TCP traffic from external hosts on the internet.

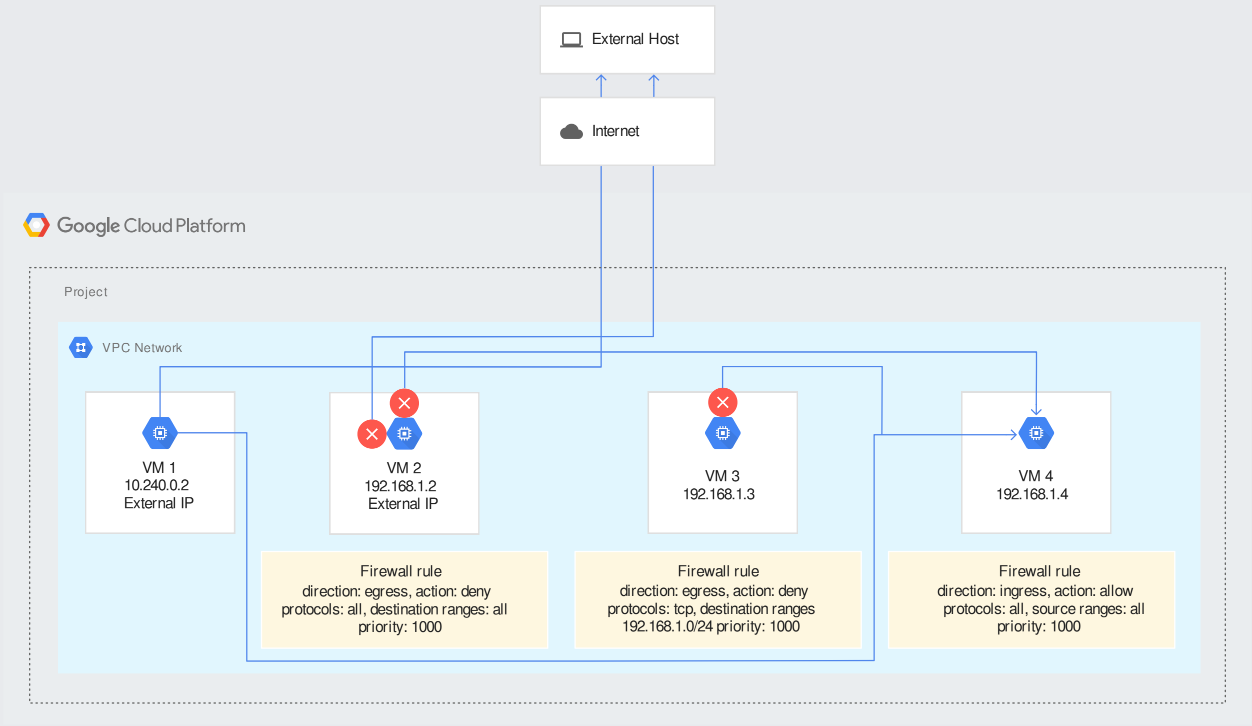
- **VM 2** has no specified ingress firewall rule, so the implied deny ingress rule blocks all incoming traffic. Connections from other instances in the network are blocked, regardless of egress rules for the other instances. Because **VM 2** has an external IP, there is a *path* to it from external hosts on the internet, but the implied deny ingress rule blocks external incoming traffic as well.

- An ingress rule with priority `1000` is applicable to **VM 3**. This rule allows TCP traffic from instances in the network with the network tag `client`, such as **VM 4**. TCP traffic from **VM 4** to **VM 3** is allowed because **VM 4** has no egress rule blocking such communication (only the implied allow egress rule is applicable). Because **VM 3** does not have an external IP, there is no path to it from external hosts on the internet.

Egress firewall rules control outgoing connections from target instances in your VPC network. Egress rules with an `allow` action permit traffic from instances based on the other components of the rule (#firewall_rule_components). For example, you can permit outbound traffic to specific destinations, such as a range of IPv4 addresses, on protocols and ports that you specify. Similarly, egress rules with a `deny` action block traffic based on the other components of the rule.

Every egress rule needs a *destination*. The default destination is any IP address (`0.0.0.0/0`), but you can create a more specific destination by using a range of IPv4 addresses in CIDR format. When specifying a range of IPv4 addresses, you can control traffic to instances in your network and to destinations outside your network, including destinations on the internet.

The following diagram illustrates some examples where firewall rules can control egress connections. The examples use the *target* parameter in rule assignments to apply rules to specific instances.


External Host
Internet
Google Cloud Platform
Project
VPC Network
VM 1
10.240.0.2
External IP
VM 2
192.168.1.2
External IP
VM 3
192.168.1.3
VM 4
192.168.1.4
Firewall rule
direction: egress, action: deny
protocols: all, destination ranges: all
priority: 1000
Firewall rule
direction: egress, action: deny
protocols: tcp, destination ranges
192.168.1.0/24 priority: 1000
Firewall rule
direction: ingress, action: allow
protocols: all, source ranges: all
priority: 1000

(/vpc/images/firewalls/firewall_overview_egress_examples.svg)
Egress firewall rules example (click to enlarge)

- **VM 1** has no specified egress firewall rule, so the implied allow egress rule lets it send traffic to any destination. Connections to other instances in the VPC network are allowed, subject to applicable ingress rules for those other instances. **VM 1** is able to send traffic to **VM 4** because **VM 4** has an ingress rule allowing incoming traffic from any IP address range. Because **VM 1** has an external IP address, it is able to send traffic to external hosts on the internet. Incoming responses to traffic sent by **VM 1** are allowed because firewall rules are stateful.

- An egress rule with priority `1000` is applicable to **VM 2**. This rule denies all outgoing traffic to all destinations (`0.0.0.0/0`). Outgoing traffic to other instances in the VPC network is blocked, regardless of the ingress rules applied to the other instances. Even though **VM 2** has an external IP address, this firewall rule blocks its outgoing traffic to external hosts on the internet.
- An egress rule with priority `1000` is applicable to **VM 3**. This rule blocks its outgoing TCP traffic to any destination in the `192.168.1.0/24` IP range. Even though ingress rules for **VM 4** permit all incoming traffic, **VM 3** cannot send TCP traffic to **VM 4**. However, **VM 3** is free to send UDP traffic to **VM 4** because the egress rule only applies to the TCP protocol.

  Also, **VM 3** can send any traffic to other instances in the VPC network outside the `192.168.1.0/24` IP range, as long as those other instances have ingress rules to permit such traffic. Because it does not have an external IP address, it has no path to send traffic outside the VPC network.

- To create and work with firewall rules, see Using firewall rules (/vpc/docs/using-firewalls).