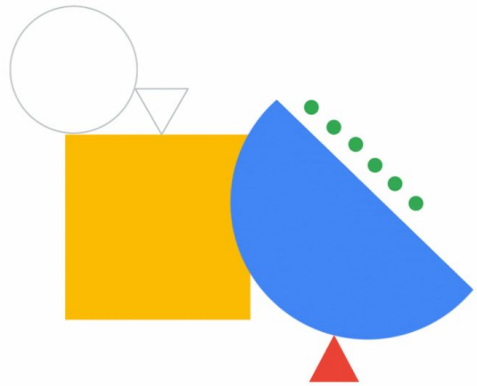


Monitoring Google Cloud Network



In this module, let's spend some time analyzing Google's Virtual Private Cloud.

Objectives

01

Collect and analyze VPC Flow Logs, Firewall Rules Logging, load balancer logs, and Cloud NAT logs.

02

Enable and monitor Packet Mirroring.

03

Explain the capabilities of the Network Intelligence Center.



Specifically, you learn to:

- Collect and analyze VPC Flow Logs, Firewall Rules Logging, load balancer logs, and Cloud NAT logs so you can see what's happening to the traffic across your network.
- Enable Packet Mirroring so you can replicate packets at the virtual machine network interface, and forward it for further analysis.
- We will also cover the capabilities of the Network Intelligence Center.

In this section, you explore



- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ Load Balancer logs
- ✓ Cloud NAT logs
- ✓ Packet Mirroring
- ✓ Network Intelligence Center

Let's start with monitoring the network.

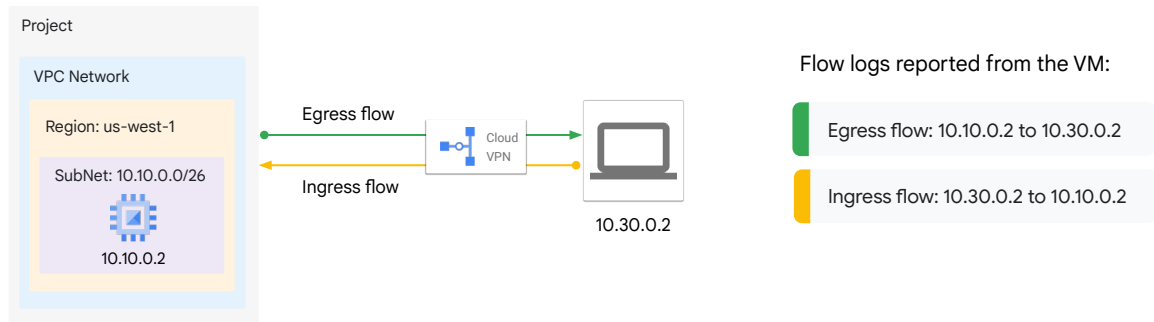
VPC Flow Logs is used to monitor network
by **recording a portion** of network flows
sent and received by VM instances
(including GKE nodes).

VPC Flow Logs records a sample (about one out of ten packets) of network flows sent from and received by VM instances, including Google Kubernetes Engine nodes. These logs can be used for network monitoring, traffic analysis, forensics, real-time security analysis, and expense optimization.

VPC Flow Logs is part of Andromeda, the software that powers VPC networks. VPC Flow Logs introduces no delay or performance penalty when enabled.

VPC Flow Logs example

VM to external traffic flow



This slide shows an example of a VM to external traffic flow pattern. VPC Flow Logs provides visibility into traffic, which helps monitor the flow between zones and IP addresses.

In this example, the traffic flows between a VM and an external network connected either through a Cloud VPN or Cloud Interconnect.

Traffic flows are reported from the VM only and include:

- Ingress traffic: The logs reported with VM as its destination. In this example, the ingress traffic is reported from the source VM 10.30.0.2 to 10.10.0.2
- Egress traffic: The logs reported with VM as its source. In this example, the egress traffic is reported from the source VM 10.10.0.2 to 10.30.0.2

VPC Flow Logs properties



Samples are from the VM's perspective.



Samples are logged for each VMs.



VMs support multiple network interface

These are some of the main properties you must remember when working with VPC Flow Logs:

- VPC Flow Log samples are from a VM's perspective. For this reason, if an egress firewall is denied, those packets are sampled by VPC Flow Logs. Similarly, the ingress blocked packets are not logged because they are sampled after the ingress firewall rules.
- VPC Flow Logs samples TCP, UDP, ICMP, ESP and GRE flows from each VM. It records inbound and outbound flows for each VM, thus capturing traffic between VM's, VM to on-premises, VM to another host on the internet.
- VMs support multiple network interface and can be enabled at the subnet level.

Enabling VPC Flow Logs

- ✓ VPC Flow Logs is activated or deactivated at a subnet level.
 - All VMs within that subnet have VPC Flow Logs automatically enabled.
- ✓ You can enable VPC Flow Logs during subnet creation.
 - You can optionally adjust log sampling and aggregation to adjust the metadata and sample rate written to logs.

You can activate or deactivate VPC Flow Logs per VPC subnet. When enabled for a subnet, VPC Flow Logs collects data from all VM instances in that subnet.

To enable VPC Flow Logs, during subnet creation, select **On** next to Flow Logs. You can optionally adjust log sampling and aggregation to adjust the metadata and sample rate that is written to logs.

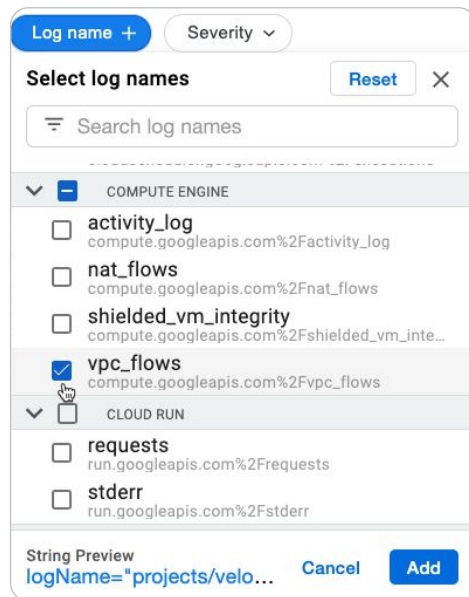
Log entries contain many useful fields

Field	Type	Description
src_ip	string	Source IP address
src_port	int32	Source port
dest_ip	string	Destination IP address
dest_port	int32	Destination port
protocol	int32	IANA protocol number

Each log entry contains a record of different fields. For example, this table illustrates the IP connection information that is recorded. Information consists of the source IP address and port, the destination IP address and port, and the protocol number. This set is commonly referred to as 5-tuple.

Other fields include the start and end time of the first and last observed packet, the bytes and packets sent, instance details including network tags, VPC details, and geographic details. For more information on all data recorded by VPC Flow Logs, see the [documentation](#).

Use Logs Explorer to access your VPC Flow Logs



Logs Explorer can be used to access the VPC Flow Logs. The entries will be `vpc_flows` below the Compute Engine section. Searching the log names for `vpc_flows` works well.

Analyze logs with Log Analytics

The screenshot displays the Google Cloud Log Analytics interface. At the top, there's a 'Log Analytics' header with a 'SHARE LINK' button. Below it, a 'Query' section shows a SQL query:

```
1 SELECT timestamp, resource.type, severity, json_payload
2 FROM 'logs_next22_US._AllLogs'
3 WHERE timestamp > TIMESTAMP_SUB(CURRENT_TIMESTAMP(), INTERVAL 1 HOUR)
4 AND json_payload IS NOT NULL
5 AND JSON_VALUE(json_payload.message) = 'request complete'
6 AND JSON_VALUE(resource.labels.pod_name) LIKE 'frontend%'
7 LIMIT 50
```

 Above the query are buttons for 'Format', 'Clear', and 'SQL reference'. To the right are 'Run in BigQuery' and 'Run query' buttons, with a 'Ready to run' status. Below the query, the 'Log views' section shows 'Results (50)' with a 'Download' button. The results table has columns for log ID, timestamp, resource type, severity, and log message. The first three rows are visible, showing timestamps from 2022-09-22 and resource type 'k8s_container'. The third row's log message is expanded, showing a JSON object with details like 'http.req.id', 'http.req.method', 'http.req.path', 'http.res.bytes', 'http.res.status', 'http.res.took.ms', 'message', 'session', and 'timestamp'.

Upgrade bucket to use Log Analytics and create a linked dataset to make log data visible to BigQuery.

Log Analytics powered by BigQuery provides new capabilities to analyze flow log data and generate useful insights. With Log Analytics:

- You can analyze ad-hoc query-time without complex pre-processing as before.
- You can use BigQuery to query data and upgrade buckets to use Log Analytics and then create a linked dataset.

Refer to the documentation for curated [sample queries](#) to get started with Flow Log Analysis.

In this section, you explore



- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ Load Balancer Logs
- ✓ Cloud NAT Logs
- ✓ Packet Mirroring
- ✓ Network Intelligence Center

Another essential part of knowing what's happening at the VPC network level is knowing what the firewall rules are doing.

VPC firewall rules



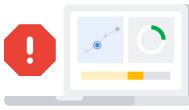
Allow or deny connections.

Protect instances.

VPC firewall rules let you allow or deny connections to or from your virtual machine (VM) instances based on a configuration that you specify.

Enabled VPC firewall rules are always enforced, and protect your instances regardless of their configuration and operating system, even if they didn't start.

Firewall Rules Logging



Did my firewall rules cause that application outage?



How many connections match the rule I just created?



Are my firewall rules stopping (or allowing) the correct traffic?

Firewall Rules Logging lets you audit, verify, and analyze the effects of your firewall rules.

It can help answer questions like:

- Did my firewall rules cause that application outage?
- How many connections match the rule I just created?
- Are my firewall rules stopping (or allowing) the correct traffic?

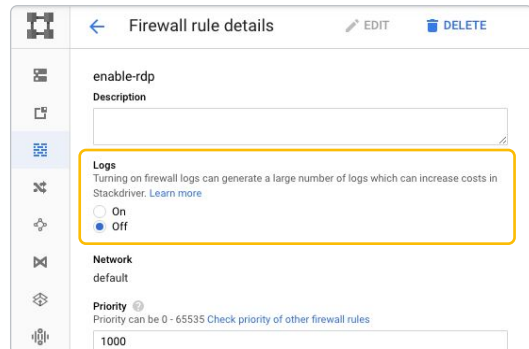
See the Firewall Rule Logging [documentation](#) for details.

Enabling Firewall Rules Logging in the console

Firewall Rules Logging is disabled by default.

You enable it on a per-rule basis.

Firewall Rules Logging can only record TCP and UDP connections.



The screenshot shows the 'Firewall rule details' page for a rule named 'enable-rdp'. The 'Logs' section is highlighted with a yellow box. It contains the text: 'Turning on firewall logs can generate a large number of logs which can increase costs in Stackdriver. [Learn more](#)'. Below this text are two radio buttons: 'On' (which is selected) and 'Off'. Other visible fields include 'Description' (empty), 'Network' (set to 'default'), and 'Priority' (set to '1000').

By default, Firewall Rules Logging is disabled.

You can enable it on a per-rule basis. In the slide screenshot, you're editing the firewall rule named *enable-rdp*. Selecting the radio button will enable firewall rules.

Note: Firewall Rules Logging can only record TCP and UDP connections. For other protocols, use Packet Mirroring.

Caution: Firewall Rules Logging can generate a lot of data, which might have a cost implication.

Enabling Firewall Rules Logging in the CLI

✓ To activate

```
$ gcloud compute firewall-rules update [NAME] --enable-logging
```

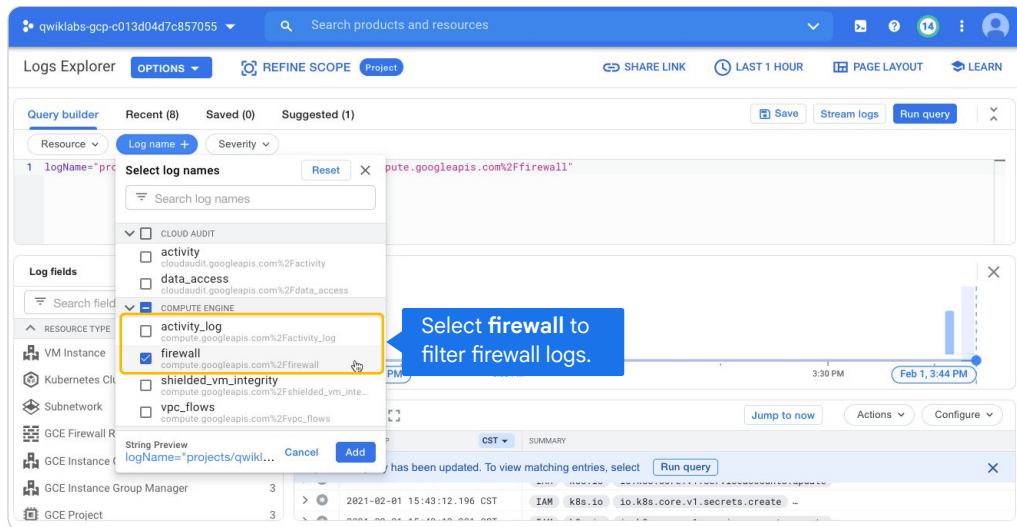
✓ To deactivate

```
$ gcloud compute firewall-rules update [NAME] --no-enable-logging
```

Firewall Rules Logging can also be activated on existing firewall rules by using the CLI.

See these two examples on this slide. In both, the [NAME] tag will be the name of your firewall rule.

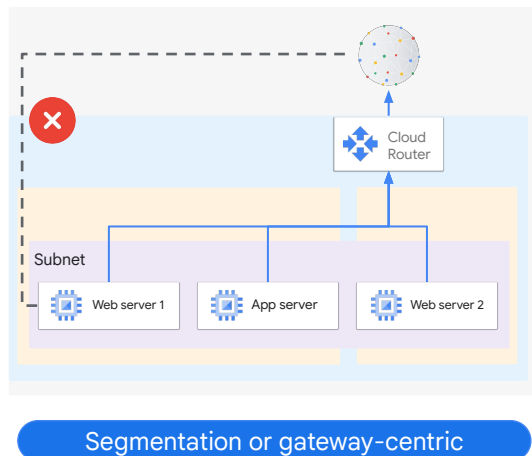
Viewing the firewall rules logs



Like all Google Cloud logs, use Logs Explorer to view logs in real time or to configure exports.

To filter for firewall logs and network policy firewall logs, below the Compute Engine resource, select **firewall**.

Firewall rules provide microsegmentation



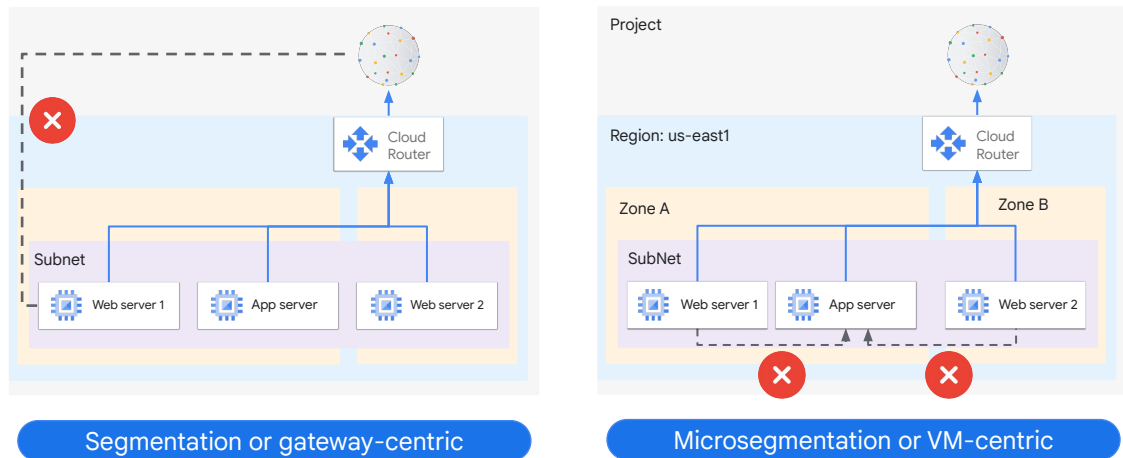
Many are familiar with classic segmentation or gateway-centric firewalls.

In this example, you can see a private network, possibly at your office or home.

At the network boundary, where the private network meets the outside internet, sits a firewall.

A segmentation firewall is designed to segment and secure a protected network from an outside insecure network.

Firewall rules provide microsegmentation



Google Cloud VPC firewalls are micro-segmentation firewalls.

These firewalls function more like a bunch of micro-firewalls, each operating over the Network Interface Controller (NIC) of every VM connected to the VPC.

The micro-firewalls can then grant or deny any configured incoming or outgoing traffic.

Now, imagine we have an issue.

We have two different web servers. After some configuration changes by a particular DevOps team, the web servers can no longer access the application server they both share.

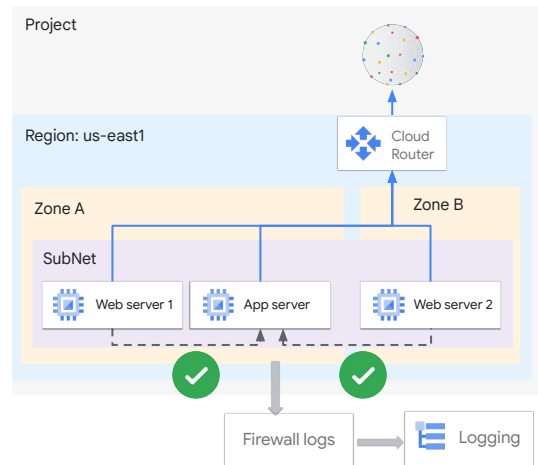
How can we tell if the issue is firewall-related? Let's see.

Troubleshooting: Using rules to catch incorrect traffic

Logging all denied connections creates too many log entries.

Temporarily create a high-priority rule to allow traffic to the server.

If traffic now gets through, examine the logs to find the root cause.



If the connectivity issue is related to a firewall, then there are two major possibilities:

A firewall rule is actively blocking the incoming connections from the web servers.

Or

Network traffic is blocked by default in most networks. A firewall rule might not be allowing the traffic from the web servers as it should.

Logging all denied connections could generate significant data that would take time and effort to monitor. So, instead of starting with option one, start with option two.

Create a temporary high-priority rule designed to allow the web server traffic through to the app server. Enable Cloud logging on it so you can examine the entries.

Suddenly the traffic is getting through, so you know it's firewall related. Now examine the log entries. Also, find the existing rule supposed to be allowing the traffic and see what you can find.

Hey, look at that! The rule that's supposed to allow the traffic is based on a network tag named *webserver*. The web server machines are actually using the network tag *web-server*.

There it is, that's your problem.

In this section, you explore



- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ **Load Balancer Logs**
- ✓ Cloud NAT logs
- ✓ Packet Mirroring
- ✓ Network Intelligence Center

Several of Google Cloud load balancers support monitoring or logging.

Load balancers support for Cloud Logging



All the Google Cloud load balancers support Cloud Logging and Cloud Monitoring:



Internal and External HTTP(S) load balancers



Internal and External TCP/UDP Network load balancers



External SSL Proxy and TCP Proxy load balancers



Internal TCP Proxy load balancers



The log type, log fields, and metrics supported vary based on the the load balancer type.



Load balancing logs is used to debug and analyze user traffic.

While all the Google Cloud load balancers support Cloud Logging and Cloud Monitoring, the log type and log fields supported vary based on the type of the load balancers. These include:

- Internal and external HTTP(S) load balancers
- Internal and External TCP/UDP Network load balancers
- External SSL Proxy and TCP Proxy load balancers
- Internal TCP Proxy load balancers

Cloud Logging for load balancing logs all the load balancing requests sent to your load balancer. These logs can be used for debugging and analyzing your user traffic. You can view request logs and export them to Cloud Storage, BigQuery, or Pub/Sub for analysis. For example, in network load balancer, per-connection logging gives you insight into how each connection is routed to serving backends.

The internal and external HTTP(S) load balancers support logging



Activated and deactivated on a per backend service basis

For external HTTP(S) load balancers with backend buckets, logging is automatically enabled and cannot be deactivated.

Logging can be enabled on a per backend service basis.

URL map might reference more than one backend service.

Use exclusion, if you do not want the logs to be stored in Cloud Logging.

For external HTTP(S) load balancers with backend buckets, logging is automatically enabled and cannot be disabled. You can activate logging on a per backend service basis. A single internal HTTP(S) load balancer URL map can reference more than one backend service. You might need to enable logging for more than one backend service, depending on your configuration. It will be enabled by default for all new load balancer backends. But backends created before the Globally Available (GA) release of load balancer logging might require manual configuration.

Fields in a log record

LogEntry

Contains severity, project ID, project number, and timestamp information.

HttpRequest

Contains a method, URL, status, remote IP address, latency string.

resource

Contains the monitored resource associated with a log entry.

jsonPayload

Contains `statusDetails` field that includes a string that explains why the load balancer returned the HTTP status, cache, and failure information.

HTTP(S) load balancing log entries contain information useful for monitoring and debugging your HTTP(S) traffic. Make sure to [check the documentation for further details](#).

Log entries contain the following types of information:

- LogEntry format includes general information shown in most logs, such as severity, project ID, project number, timestamp, and so on.
- However, *HttpRequest.protocol* is not populated for HTTP(S) load balancing logs. This can include a method, a URL, remote IP address, a protocol, a latency string or a user agent.
- resource contains the monitored resource type associated with the log entry.
- jsonPayload contains the *statusDetails* field. This field holds a string that explains why the load balancer returned the HTTP status that it did.
- Redirects (such as HTTP response status code 302 Found) issued from the load balancer are *not* logged. Redirects issued from the backend instances are logged.

Example of using a load balancing log record

HTTP error response code (5XX)



Refer to Load Balancer logs to determine the source of error.



statusDetails field: response_sent_by_backend indicates it is a backend issue.



statusDetails field: failed_to_pick_backend indicates that the load balancer failed to pick a healthy backend to handle a request.

Let's take an example of how to use log record information to troubleshoot a load balancing issue. Consider a scenario where the load balancer generates an HTTP error resource code 5XX and sends the same error code to the client.

Refer to the load balancer logs to determine the source of an error:

- Within the statusDetails field: the response_sent_by_backend indicates it is a backend issue.
- Whereas, failed_to_pick_backend indicates that the load balancer failed to pick a healthy backend to handle a request.

In this section, you explore

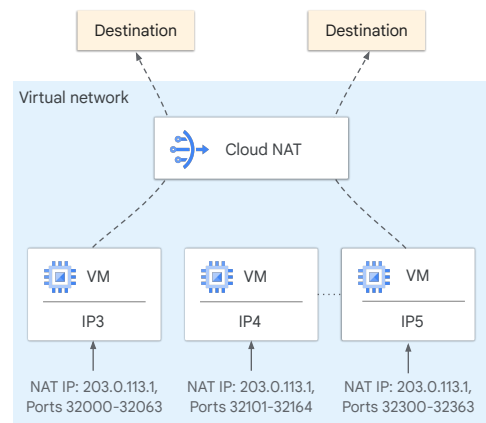


- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ Load Balancer logs
- ✓ **Cloud NAT logs**
- ✓ Packet Mirroring
- ✓ Network Intelligence Center

Another piece of the network telemetry features in Google Cloud is Cloud NAT logs.

Cloud NAT overview

- It allows Google Cloud Compute workload with no external IP to send packets to the internet.
- It's a fully managed, proxyless NAT service in Andromeda.
- These are some of its benefits:
 - Reduces the need for individual VMs to each have external IP addresses.
 - Automatically scales the number of NAT IP addresses that it uses.
 - Is not dependent on a single physical gateway device.



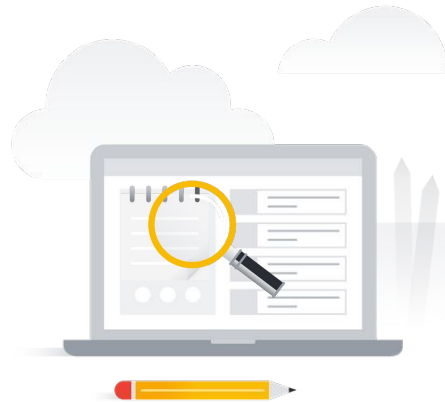
Cloud NAT is the Google-managed Network Address Translation service. It lets you provision your application instances without public IP addresses, and it also lets them access the internet in a controlled and efficient manner. With Cloud NAT, your private instances can access the internet for updates, patching, configuration management, and more.

There are many Cloud NAT benefits.

- VMs without external IP addresses can access destinations on the internet. For example, you might have VMs that only need internet access to download updates or complete provisioning. Cloud NAT lets you configure these VMs with an internal IP address. Thus, your organization needs fewer external IP addresses.
- Cloud NAT can be configured to automatically scale the number of NAT IP addresses that it uses. Cloud NAT supports VMs that belong to managed instance groups, including those with autoscaling enabled.
- Cloud NAT is not dependent on a single, physical gateway device. Cloud NAT is a distributed, software-defined managed service. You configure a NAT gateway on a Cloud Router, which provides the control plane for Cloud NAT. Cloud Router contains the NAT configuration parameters.

Cloud NAT logging

- ✓ A NAT log is created when:
 - A network connection using NAT is created.
 - A packet is dropped due to port unavailability.
- ✓ It lets you log NAT connections and/or errors.
- ✓ Logs contain TCP and UDP traffic only.
- ✓ The log rate threshold will reach a maximum of 50-100 entries per second, per vCPU.



Cloud NAT logging lets you log NAT TCP and UDP connections and errors. When Cloud NAT logging is enabled, a log entry can be generated when a network connection that uses Cloud NAT is created, and/or when an egress packet is dropped because no port was available for Cloud NAT.

You can opt to log both kinds of events, or just one or the other. Logs contain TCP and UDP traffic only, and the log rate threshold will reach a maximum of 50-100 log events per vCPU before log filtering.

Cloud NAT logging might be enabled when a new Cloud NAT gateway is first created, or by editing the settings of an existing gateway.

To view the collected logs in Logs Explorer, filter to the Cloud NAT Gateway resource and optionally, restrict to a particular region or Gateway.

In this section, you explore

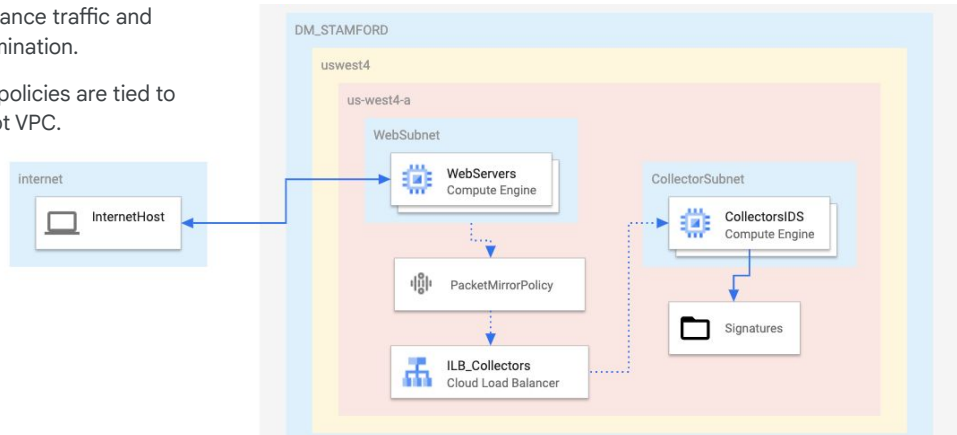


- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ Load Balancer logs
- ✓ Cloud NAT logs
- ✓ [Packet Mirroring](#)
- ✓ Network Intelligence Center

Another way to monitor the network traffic flowing in and out of your Compute Engine virtual machines is to use Packet Mirroring.

Packet Mirroring: Visualize and protect your network

- It clones VPC instance traffic and forwards for examination.
- Packet Mirroring policies are tied to workloads and not VPC.



Packet Mirroring clones the traffic of specific instances in your Virtual Private Cloud (VPC) network and forwards it for examination. Packet Mirroring captures all ingress and egress traffic and packet data, such as payloads and headers.

The mirroring happens on the virtual machine (VM) instances, not on the network. Therefore, Packet Mirroring consumes additional bandwidth on the hosts.

Packet Mirroring is useful when you need to monitor and analyze your security status. It exports all traffic, not only the traffic between sampling periods. For example, you can use security software that analyzes mirrored traffic to detect all threats or anomalies.

Also, you can inspect the full traffic flow to detect application performance issues and to provide network forensics for Payment Card Industry Data Security Standards (PCI DSS) compliance and other regulatory use cases. We will elaborate on this further in the next few slides.

Obviously, Packet Mirroring can generate significant data, so collector destination is generally an instance group behind a TCP/UDP load balancer or equivalent technology.

Packet Mirroring: Overcoming bandwidth limitations

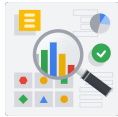
- Packet Mirroring consumes the egress bandwidth of the mirrored instances.
 - Use filters to reduce the bandwidth on mirrored instances.
 - Filters can be based on protocol, IP ranges, traffic directions, etc.
 - The current maximum of filters for Packet Mirroring is 30.

One of the major limitations of Packet Mirroring is bandwidth consumption. Packet Mirroring consumes the egress bandwidth of the mirrored instances. However, there is a work around. Use filters to reduce the traffic collected for mirrored instances. This filter can be used for IP address ranges, protocols, traffic directions and lot more.

The current maximum number of filters that can be used for Packet mirroring is 30.

For more information, refer to the [documentation](#).

Packet Mirroring: Use cases



Network and
application monitoring



Security and
compliance



Network forensics for
PCI compliance

Two main use cases where Packet Mirroring is useful in security and monitoring. Let's explore each of these use cases in detail.

Network and application monitoring: Network engineers can use the data from Packet Mirroring to:

- Maintain integrity of deployment.
- Troubleshoot packet loss issues by analyzing protocols.
- Troubleshoot reconnection and latency issues by analyzing real time traffic patterns.

Security and compliance: Implement zero-trust by monitoring network traffic across and within the trust boundaries without any network re-architecture. Packet Mirroring helps capture multiple packets for a single flow. This information can be quite useful for the implementation and usage of the following security tools:

- Intrusion detection systems match signatures with multiple packets of a single flow.
- Deep Packet Inspection engines inspect payloads for anomalies.

Network forensics for PCI compliance: Packet mirroring help capture, process and preserve forensic of different attack vectors.

Monitoring Packet Mirroring

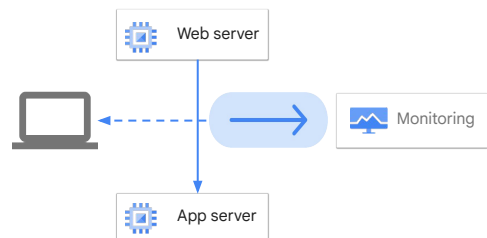
✓ Use metrics to verify that instances are being mirrored as intended.

● Mirrored Packets count

● Mirrored Bytes count

● Dropped Packets count

✓ Monitoring can also spot where packet mirroring shouldn't be happening.



- Packet Mirroring exports monitoring data about mirrored traffic to Cloud Monitoring. You can use monitoring metrics to check whether traffic from a VM instance is being mirrored as intended. For example, you can view the mirrored packet or byte count for a specific instance.
- You can also view the monitoring metrics of mirrored VM instances or instances that are part of the collector destination (internal load balancer). For mirrored VM instances, Packet Mirroring provides metrics specific to mirrored packets such as mirrored packets count, mirrored bytes count and dropped packets count.
- Monitoring can also spot where packet mirroring is being used unnecessarily or unexpectedly. Remember that, as noted earlier, mirroring generates significant data that requires storage and processing. Also, note that it slows the network throughput of the virtual machines being monitored and might accidentally expose sensitive data.

In this section, you explore



- ✓ VPC Flow Logs
- ✓ Firewall Rules Logging
- ✓ Load Balancer logs
- ✓ Cloud NAT logs
- ✓ Packet Mirroring
- ✓ [Network Intelligence Center](#)

This section is a bit of a detour, but let's at least mention the Network Intelligence Center and how it helps with network analysis.

Network Intelligence Center

Centralized network monitoring and visibility

Reduced troubleshooting time and effort

Improved overall user experience



Network Intelligence



Network Topology



Connectivity Tests



Performance Dashboard



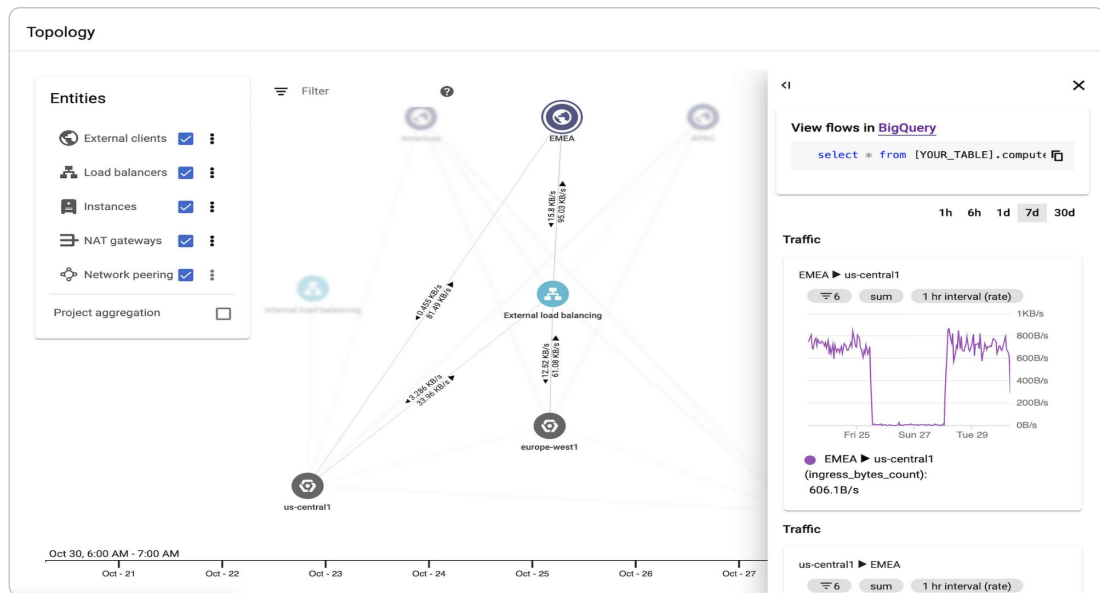
Firewall Insights



Network Analyzer

Network Intelligence Center gives you centralized monitoring and visibility into your network. It reduces troubleshooting time and effort and increases network security, all while improving the overall user experience. Currently, it offers five modules: Network Topology, Connectivity Tests, Performance Dashboard, Firewall Insights, and Network Analyzer.

Network topology



Network Topology visualizes your Google Cloud network as a graph.

You can use the graph to explore your existing configurations and quickly troubleshoot networking issues.

You can select network entities, filter, see lines of communication with bandwidth information, expand and collapse hierarchies, and select time boundaries.

Connectivity tests

Connectivity Tests

CREATE CONNECTIVITY TEST

RERUN

DELETE

This test lets you check connectivity between network endpoints. It analyzes your configuration and, if the configuration is eligible, sends packets through the live data plane. [Learn more](#)

Filter

Filter by test name or protocol

<input type="checkbox"/>	Name	Protocol	Source	Destination	Destination port	Last test time
<input type="checkbox"/>	http	tcp	10.150.0.3 (default)	10.150.0.2 (default)	80	2023-05-16 (14:04:09)
<input type="checkbox"/>	test	tcp	10.0.0.1 (default)	10.1.1.1 (default)	80	2023-05-16 (13:31:27)
<input type="checkbox"/>	vm-test1	icmp	grafana-ent (default, 10.150.0.3)	ray (default, 10.150.0.2)	-	2023-05-16 (14:19:15)

- Quickly diagnose connectivity issues and prevent outages.
- Verify the configuration change effect to help prevent outages.

The [Connectivity Tests](#) tool in Network Intelligence Center helps you to quickly diagnose connectivity issues and prevent outages.

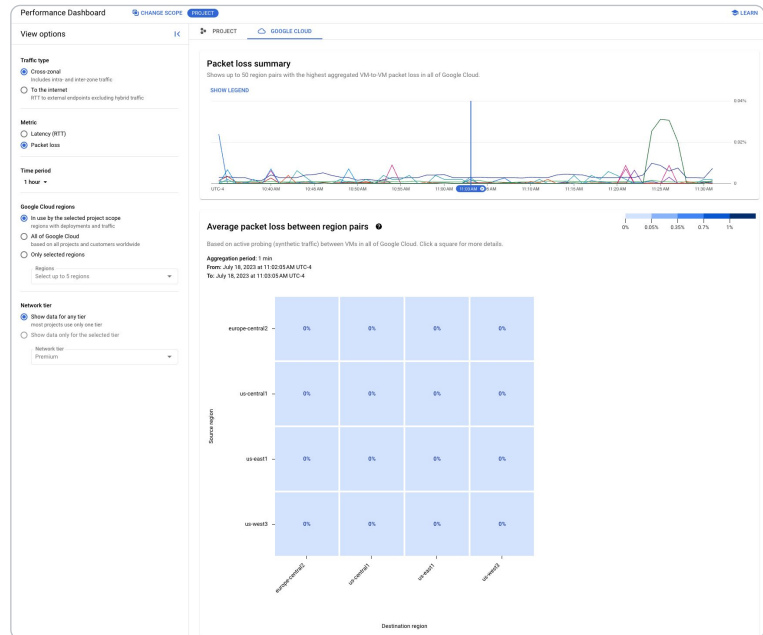
These tests let you self-diagnose connectivity issues within Google Cloud or from Google Cloud to an external IP address (the connectivity issue could be on-premises or in another cloud). The results help to isolate whether the issue is in Google Cloud.

Run tests to help verify the effect of configuration changes and ensure that network intent captured by these tests is not violated, proactively preventing network outages.

These tests also help assure network security and compliance.

Performance dashboard

Packet loss metrics aggregated across zones



Performance Dashboard gives you visibility into the performance of your VPC.

The **Packet Loss** tab shows the results of active probing between your VMs in a given VPC.

To get this data, it runs workers on the physical hosts that house your VMs.

These workers insert and receive probe packets that run on the same network as your traffic, revealing issues on that network.

Workers run on the physical host and not on your VM. Therefore, these workers do not consume VM resources and the traffic is not visible on your VMs.

Packet loss is aggregated for all zone pairs.

Performance dashboard

Median Latency summaries aggregated across zones



The **Latency** tab aggregates latency information based on a sample of your actual Transmission Control Protocol (TCP) VM traffic. The method used is similar to the one used for [VPC Flow Logs](#).

The latency is calculated as the time that elapses between sending a TCP sequence number (SEQ) and receiving a corresponding Acknowledgement (ACK) that contains the network Round Trip Time (RTT) and TCP stack related delay.

The latency metric is only available if TCP traffic is around 1,000 packets per minute or higher.

Firewall Insights helps you understand and optimize your firewall rules



Firewall Insights, a component product of Network Intelligence Center, produces metrics and insights that let you make better decisions about your firewall rules. It provides data about how your firewall rules are being used, exposes misconfigurations, and identifies rules that could be made more strict.

Firewall Insights uses Cloud Monitoring metrics and Recommender insights.

Cloud Monitoring collects measurements to help you understand how your applications and system services are performing. A collection of these measurements is generically called a metric. The applications and system services being monitored are called monitored resources. Measurements might include the latency of requests to a service, the amount of disk space available on a machine, the number of tables in your SQL database, the number of widgets sold, and so forth. Resources might include virtual machines, database instances, disks, and so forth.

Recommender is a service that provides recommendations and insights for using resources on Google Cloud. These recommendations and insights are per-product or per-service, and are generated based on heuristic methods, machine learning, and current resource usage. You can use insights independently from recommendations. Each insight has a specific insight type. Insight types are specific to a single Google Cloud product and resource type. A single product can have multiple insight types, where each provides a different type of insight for a different resource.

Firewall Insights metrics let you analyze how your firewall rules are used

- ✓ Verify that firewall rules are being used in the intended way.
- ✓ Verify that firewall rules allow or block their intended connections.
- ✓ Perform live debugging of connections that are inadvertently dropped.
- ✓ Discover malicious attempts to access your network.

Firewall Insights metrics let you analyze the way that your firewall rules are being used. Firewall Insights metrics are available through Cloud Monitoring and the Google Cloud console. Metrics are derived through Firewall Rules Logging.

With Firewall Insights metrics, you can perform the following tasks:

- Verify that firewall rules are being used in the intended way.
- Over specified time periods, verify that firewall rules allow or block their intended connections.
- Perform live debugging of connections that are inadvertently dropped because of firewall rules.
- Discover malicious attempts to access your network, in part by getting alerts about significant changes in the hit counts of firewall rules.

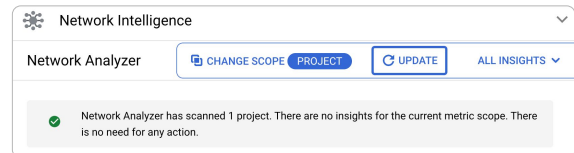
Network Analyzer

Automatically monitors your VPC network configurations.

Detects misconfigurations and suboptimal configurations.

Provides insights on network topology, firewall rules, routes, configuration dependencies, and connectivity.

Identifies network failures, provides root cause information, and suggests possible resolutions.



Network Analyzer automatically monitors your VPC network configurations and detects misconfigurations and suboptimal configurations. It provides insights on Network Topology, firewall rules, routes, configuration dependencies, and connectivity to services and applications. It identifies network failures, provides root cause information, and suggests possible resolutions.

Network Analyzer runs continuously and triggers relevant analyses based on near real-time configuration updates in your network. If a network failure is detected, it tries to correlate the failure with recent configuration changes to identify root causes. Wherever possible, it provides recommendations to suggest details on how to fix the issues.

Insights help surface issues and identify the root cause

The screenshot displays the Network Analyzer interface. On the left, a table lists various insights with columns for Priority, Resource name, Resource type, Project, Insight type, and Network insight. The table shows several insights, including errors for GKE clusters and network issues. On the right, a detailed view of a specific insight is shown, detailing the root cause and providing a link to dismiss the insight.

Priority	Resource name	Resource type	Project	Insight type	Network insight
High	cluster-2	GKE cluster	configcheck-newsnapshot-tests	Error	Node to control plane connectivity is blocked by a routing issue or missing network peering
High	cluster-2	GKE cluster	configcheck-newsnapshot-tests	Error	Control plane blocked by a routing issue or missing network peering
High	cluster-1	GKE cluster	configcheck-newsnapshot-tests	Error	Traffic from cluster blocked by ingress instance
High	cluster-3	GKE cluster	configcheck-newsnapshot-tests	Error	Traffic to public endpoint blocked by egress firewall
High	mysql-2	SQL instance	configcheck-newsnapshot-tests	Error	Connectivity issue
High	mysql-1	SQL instance	configcheck-newsnapshot-tests	Error	Connectivity issue
Medium	dynamic-routes-peer	Network	configcheck-newsnapshot-tests	Error	Dynamic route not found or a peering static route
Medium	dynamic-routes-peer	Network	configcheck-newsnapshot-tests	Error	Dynamic route not found or a subnet route

Network insight
Node to control plane connectivity is blocked by a routing issue or missing network peering

Priority
High

Insight type
Error

GKE cluster
[cluster-2](#)

Control plane endpoint
10.17.0.2

Network
[gke](#)

Project
configcheck-newsnapshot-tests

First report time
Apr 20, 11:00 PM

Documentation
[View related product documentation](#)

[DISMISS INSIGHT](#)

Network Analyzer provides insights that help identify common issues such as connectivity blockage, load balancing errors, external IP address that are not used but allocated, invalid next hop, GKE network misconfiguration and lot more. It also identifies the root cause of the insights and also provides recommended fixes.

In the example above, an insight of the type *Error*, a GKE node to control plane connectivity is generated. The insight page also describes the following:

- The root cause: an ingress firewall rule is blocking the connection between the node and the plane. This indicated that the default drywall rules were modified, removed, or shadowed by another firewall rule.
- A solution: if the root of the problem is a deleted firewall, create a new firewall rule. If it's a shadowed firewall rule, then increase the priority.

Recap

- 01 Collect and analyze VPC Flow Logs, Firewall Rules Logging, load balancer logs, and Cloud NAT logs.
- 02 Enable and monitor Packet Mirroring.
- 03 Explain the capabilities of the Network Intelligence Center.



After completing this module, you know how to:

- Collect and analyze VPC Flow Logs, Firewall Rules Logging, load balancer logs, and Cloud NAT logs so you can see what's happening to the traffic across your network.
- Enable Packet Mirroring so you can replicate packets at the virtual machine network interface and forward it for further analysis.
- And explain the capabilities of the Network Intelligence Center.