VPC Flow Logs record a sample of network flows sent from and received by VM instances (/compute/docs/instances/), including instances used as GKE nodes (/kubernetes-engine/docs/). These logs can be used for network monitoring, forensics, real-time security analysis, and expense optimization.

You can view flow logs in Stackdriver Logging (/logging), and you can export logs to any destination that Stackdriver Logging export (/logging/docs/export/configure_export_v2) supports.

Flow logs are aggregated by connection from Compute Engine VMs and exported in real time. By subscribing to Pub/Sub, you can analyze flow logs using real-time streaming APIs.

- You can enable or disable VPC Flow Logs per VPC subnet. If enabled for a subnet, VPC flow logs collects data from all VM instances in that subnet.
- VMs report on all TCP and UDP flows. Each flow record includes the information described in the Record format (#record_format) section.
- Each VM samples the TCP and UDP flows it sees, inbound and outbound, whether the flow is to or from another VM, a host in your on-premises datacenter, a Google service, or a host on the Internet. If two GCP VMs are communicating, and both are in subnets that have VPC Flow Logs enabled, both VMs report the flows.
- You can use filters to select which flow logs are excluded (/logging/docs/exclusions) from Stackdriver Logging and which are exported (/logging/docs/export/) to external APIs.
- VPC Flow Logs is natively built into the networking stack of the VPC network infrastructure. There is no extra delay and no performance penalty in routing the logged IP packets to their destination.

VPC Flow Logs provides you with real-time visibility into network throughput and performance. You can:

- Monitor the VPC network
- Perform network diagnosis
- Filter the flow logs by VMs and by applications to understand traffic changes
- Understand traffic growth for capacity forecasting

You can analyze network usage with VPC Flow Logs. You can analyze the network flows for the following:

- Traffic between regions and zones
- Traffic to specific countries on the Internet
- Top talkers

Based on the analysis, you can optimize network traffic expenses.

You can utilize VPC Flow Logs for network forensics. For example, if an incident occurs, you can examine the following:

- Which IPs talked with whom and when
- Any compromised IPs by analyzing all the incoming and outgoing network flows

You can leverage the real-time streaming APIs (through Pub/Sub (/pubsub/docs/)), and integrate with SIEM (Security Information and Event Management (https://wikipedia.org/wiki/Security_information_and_event_management)) systems. This can provide real-time monitoring, correlation of events, analysis, and security alerts.

Flow logs are collected for each VM connection at specific intervals. All packets collected for a given interval for a given connection are aggregated for a period of time (aggregation interval) into a single flow log entry. This data is then sent to Stackdriver Logging (/logging).

Logs are stored in Stackdriver Logging for 30 days. If you wish to keep logs longer than that, you must export them (/logging/docs/export/configure_export_v2) to a supported destination.

Google Cloud samples packets that leave and enter a VM to generate flow logs. Not every packet is captured into its own log record. About 1 out of every 10 packets is captured, but this sampling rate might be lower depending on the VM's load. You cannot adjust this rate.

After the flow logs are generated, Google Cloud processes them according to the following procedure:

1. **Aggregation**: Information for sampled packets is aggregated over a configurable *aggregation interval* to produce a *flow log entry*.
2. **Flow log sampling**: This is a second sampling process. Flow log entries are further sampled according to a configurable *sample rate* parameter.
3. **Metadata**: If enabled, *Metadata annotations* are added.
4. **Write to logging**: The final log entries are written to Stackdriver Logging (/logging).

You cannot change how VPC Flow Logs *collect* samples; however, you can control the secondary flow log sampling v
g sampling parameter. Even with a flow log sampling of `1.0` or 100%, at most about 10% of packets at the VM level a
ssed due to the initial sample rate. If you need to analyze all packets, you can use Packet Mirroring
/docs/packet-mirroring) and collector instances running third-party software.

Even though Google Cloud doesn't capture every packet, log record captures can be quite large. You can balance your traffic visibility and storage cost needs by adjusting the following aspects of logs collection:

- **Aggregation interval**: Sampled packets for a time interval are aggregated into a single log entry. This time interval can be 5 sec (default), 30 sec, 1 min, 5 min, 10 min, or 15 min.
- **Sample rate**: Before being written to the database, the number of logs can be sampled to reduce their number. By default, the log entry volume is scaled by 0.50 (50%), which means that half of entries are kept. You can set this from `1.0` (100%, all log entries are kept) to `0.0` (0%, no logs are kept).
- **Metadata annotations**: By default, flow log entries are annotated with metadata information, such as the names of the source and destination VMs or the geographic region of external sources and destinations. This metadata annotation can be turned off to save storage space.

Log records contain base fields, which are the core fields of every log record, and metadata fields that add additional information. Metadata fields may be omitted to save storage costs.

Some log fields are in a multi-field format, with more than one piece of data in a given field. For example, the `connection` field is of the `IpConnection` format, which contains the source and destination IP address and port, plus the protocol, in a single field. These multi-field fields are described below the record format table.

n: The values for **metadata** fields aren't based on the data plane path; they are approximations and might be missin
ect. In contrast, the values for base fields are taken directly from packet headers.

Field	Field Format	Field type: Base or Optional metadata
connection	IpConnection (#IpConnection) 5-Tuple describing this connection.	Base
start_time	string Timestamp (RFC 3339 date string format) of the first observed packet during the aggregated time interval	Base
end_time	string Timestamp (RFC 3339 date string format) of the last observed packet during the aggregated time interval	Base
bytes_sent	int64 Amount of bytes sent from the source to the destination	Base
packets_sent	int64 Number of packets sent from the source to the destination	Base
rtt_msec	int64 Latency as measured (for TCP flows only) during the time interval. This is the time elapsed between sending a SEQ and receiving a corresponding ACK and it contains the network RTT as well as the application related delay.	Base
reporter	string The side which reported the flow. Can be either ‘SRC' or ‘DEST'.	Base

Field	Field Format	Field type: Base or Optional metadata
src_instance	InstanceDetails (#InstanceDetails) If the source of the connection was a VM located on the same VPC, this field is populated with VM instance details. In a Shared VPC configuration, `project_id` corresponds to the project that owns the instance, usually the service project.	Metadata
dest_instance	InstanceDetails (#InstanceDetails) If the destination of the connection was a VM located on the same VPC, this field is populated with VM instance details. In a Shared VPC configuration, `project_id` corresponds to the project that owns the instance, usually the service project.	Metadata
src_vpc	VpcDetails (#VpcDetails) If the source of the connection was a VM located on the same VPC, this field is populated with VPC network details. In a Shared VPC configuration, `project_id` corresponds to that of the host project.	Metadata
dest_vpc	VpcDetails (#VpcDetails) If the destination of the connection was a VM located on the same VPC, this field is populated with VPC network details. In a Shared VPC configuration, `project_id` corresponds to that of the host project.	Metadata
src_location	GeographicDetails (#GeographicDetails) If the source of the connection was external to the Google VPC, this field is populated with available location metadata.	Metadata
dest_location	GeographicDetails (#GeographicDetails) If the destination of the connection was external to the Google VPC, this field is populated with available location metadata.	Metadata

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	The IANA protocol number

Field	Type	Description
project_id	string	ID of the project containing the VM
vm_name	string	Instance name of the VM
region	string	Region of the VM
zone	string	Zone of the VM

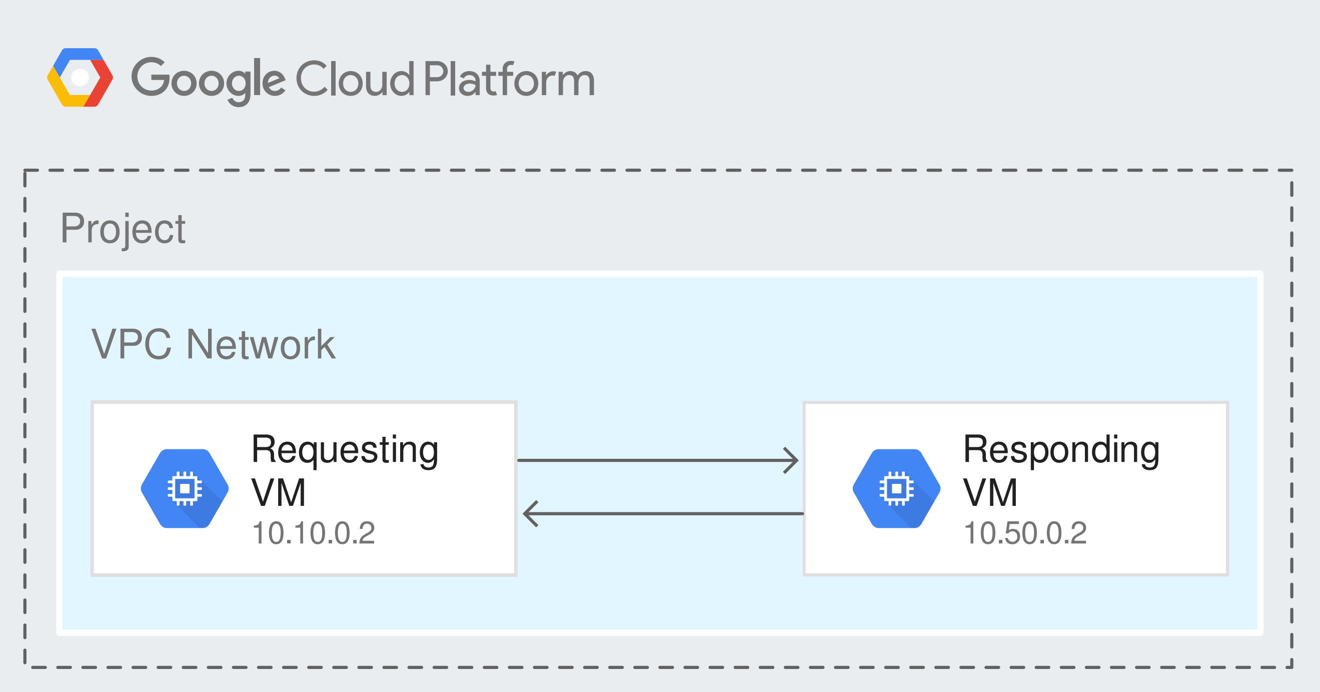
Field	Type	Description
project_id	string	ID of the project containing the VPC
vpc_name	string	VPC on which the VM is operating
subnetwork_name	string	Subnetwork on which the VM is operating

Field	Type	Description
continent	string	Continent for external endpoints
country	string	Country for external endpoints, represented as ISO 3166-1 Alpha-3 (https://en.wikipedia.org/wiki/List_of_ISO_3166_country_codes) country codes.
region	string	Region for external endpoints
city	string	City for external endpoints
asn	int32	The autonomous system number (ASN) of the external network to which this endpoint belongs.

This section demonstrates how VPC Flow Logs works for the following use cases:

- VM-to-VM flows in the same VPC

- VM-to-external flows
- VM-to-VM flows for Shared VPC (/vpc/docs/shared-vpc)
- VM-to-VM flows for VPC Peering (/vpc/docs/vpc-peering)
- VM-to-VM flows for Internal Load Balancing (/load-balancing/docs/internal)



(/vpc/images/flow-logs/flow-logs-1.svg)
VM flows within a VPC (click to enlarge)

For VM-to-VM flows in the same VPC, flow logs are reported from both requesting and responding VM, as long as both VMs are in subnets that have VPC FLow Logs enabled. In this example, VM `10.10.0.2` sends a request with 1224 bytes to VM `10.50.0.2`, which is also in a subnet that has logging enabled. In turn, `10.50.0.2` responds to the request with a reply containing 5342 bytes. Both the request and reply are recorded from both the requesting and responding VMs.

As reported by requesting VM (10.10.0.2)

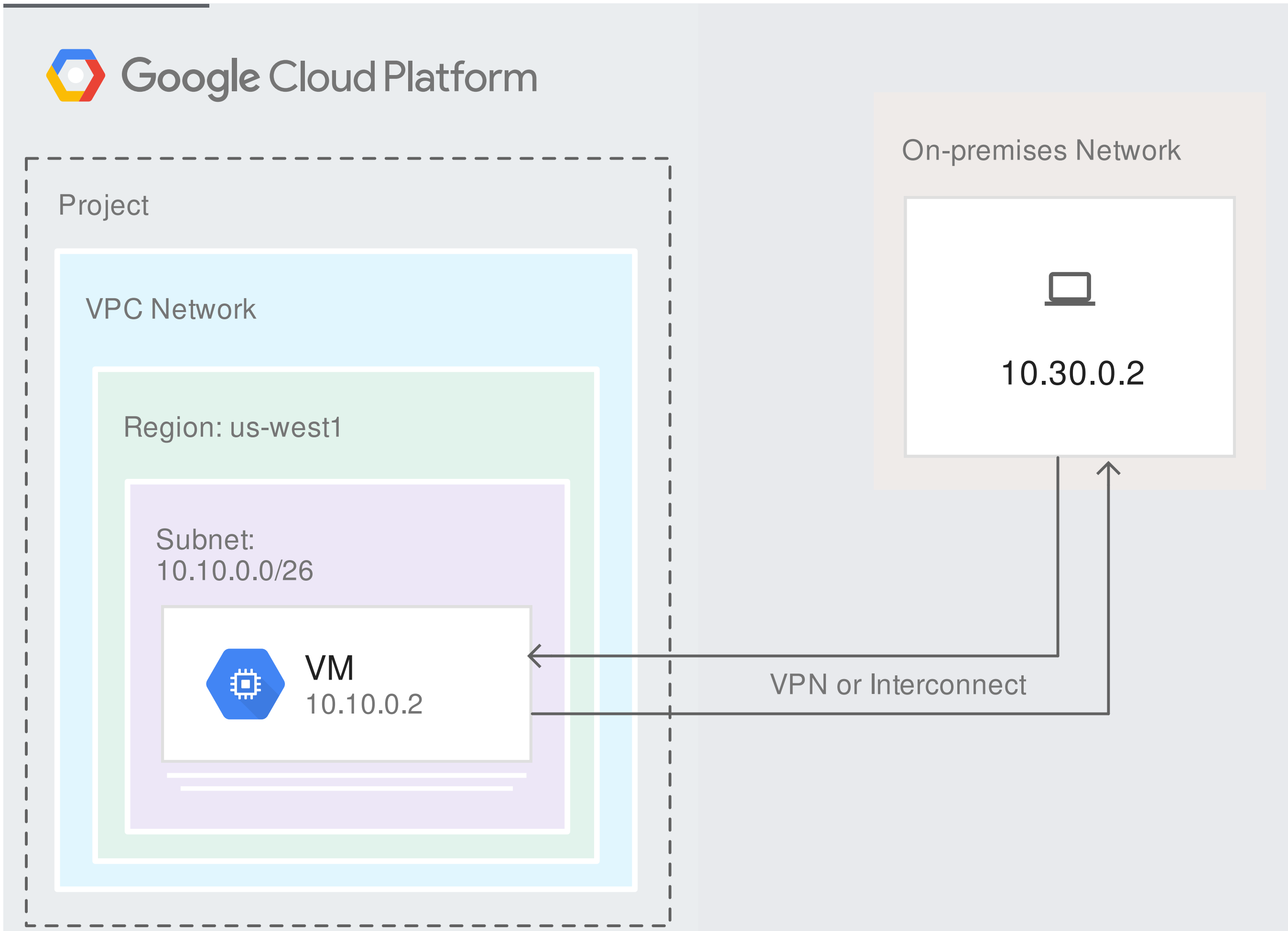
request/reply	connection.src_ip	connection.dest_ip	bytes_sent	VPC annotations

As reported by requesting VM (10.10.0.2)

request/reply	connection.src_ip	connection.dest_ip	bytes_sent	VPC annotations
request	10.10.0.2	10.50.0.2	1224	src_instance.* dest_instance.* src_vpc.* dest_vpc.*
reply	10.50.0.2	10.10.0.2	5342	src_instance.* dest_instance.* src_vpc.* dest_vpc.*

As reported by responding VM (10.50.0.2)

request/reply	connection.src_ip	connection.dest_ip	bytes	VPC annotations
request	10.10.0.2	10.50.0.2	1224	src_instance.* dest_instance.* src_vpc.* dest_vpc.*
reply	10.50.0.2	10.10.0.2	5342	src_instance.* dest_instance.* src_vpc.* dest_vpc.*



(/vpc/images/flow-logs/flow-logs-2.svg)
VM-to-external flows (click to enlarge)

For flows between a VM and an external entity, flow logs are reported from the VM only:

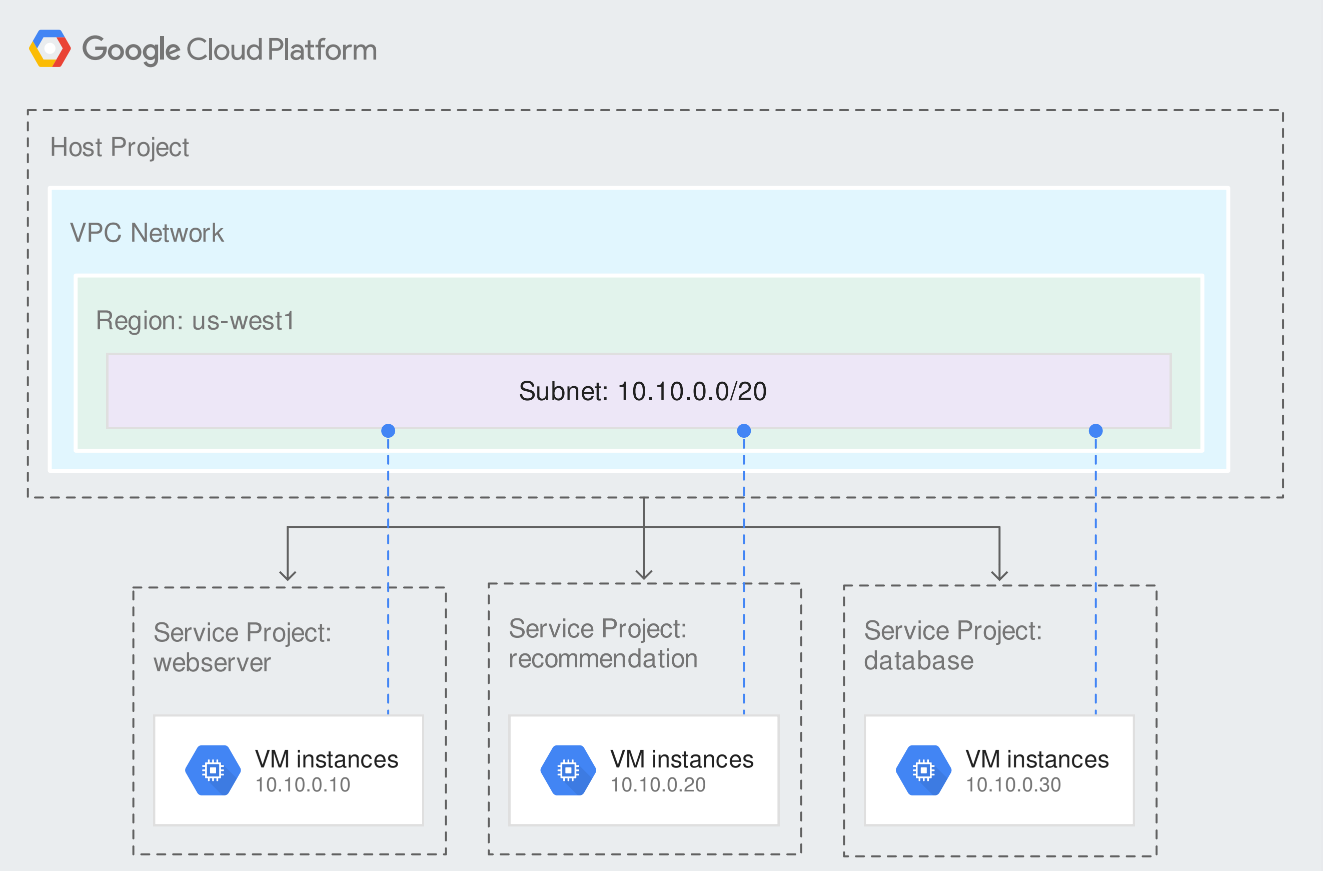
- For egress flows, the logs are reported from the VM that is the source of the traffic.
- For ingress flows, the logs are reported from the VM that is the destination of the traffic.

This applies to:

- Traffic between a VPC network and an on-premises network through VPN or Cloud Interconnect
- Traffic between VMs and locations on the Internet

In this example, VM `10.10.0.2` and on-premises endpoint `10.30.0.2` are connected through a VPN gateway or Cloud Interconnect. The outbound traffic of 1224 bytes sent from `10.10.0.2` to `10.30.0.2` is reported from the source VM, `10.10.0.2`. The inbound traffic of 5342 bytes sent from `10.30.0.2` to `10.10.0.2` is reported from the destination of the traffic, VM `10.10.0.2`.

request/reply	connection.src_ip	connection.dest_ip	bytes_sent	VPC annotations
request	10.10.0.2	10.30.0.2	1224	src_instance.* src_vpc.* dest_location.*
reply	10.30.0.2	10.10.0.2	5342	dest_instance.* dest_vpc.* src_location.*



(/vpc/images/flow-logs/flow-logs-3.svg)
Shared VPC flows (click to enlarge)

For VM-to-VM flows for Shared VPC, you can enable VPC Flow Logs for the subnet in the host project. For example, subnet 10.10.0.0/20 belongs to a Shared VPC Network defined in a host project. You can see flow logs from VMs belonging to this subnet, including ones created by service projects. In this example, the service projects are called "webserver", "recommendation", "database").

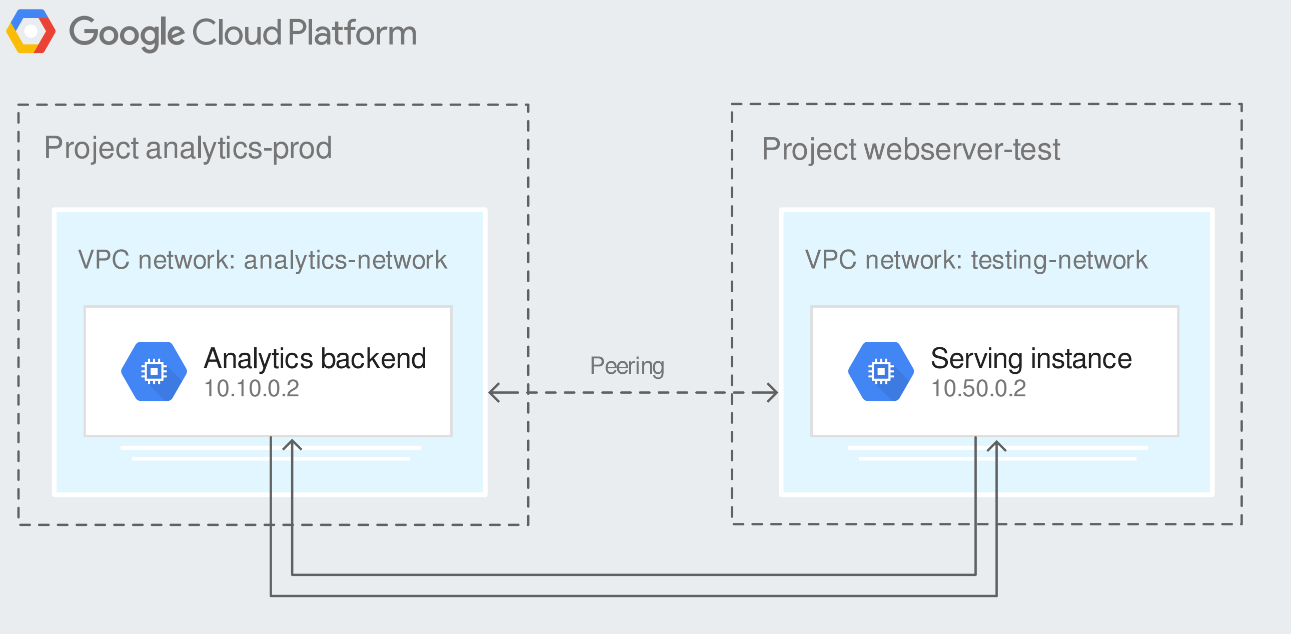
For VM-to-VM flows, if both VMs are in the same project, or in the case of a shared network, the same host project, annotations for project ID and the like are provided for the other endpoint in the connection. If the other VM is in a different project, then annotation for the other VM is not provided.

The following table shows a flow as reported by either `10.10.0.10` or `10.10.0.20`.

- `src_vpc.project_id` and `dest_vpc.project_id` are for the host project because the VPC subnet belongs to the host project.
- `src_instance.project_id` and `dest_instance.project_id` are for the service projects because the instances belong to the service projects.

connection .src_ip	src_instance .project_id	src_vpc .project_id	connection .dest_ip	dest_instance .project_id	dest_vpc .project_id
10.10.0.10	webserver	host_project	10.10.0.20	recommendation	host_project

Service projects do not own the Shared VPC network and do not have access to the flow logs of the Shared VPC network.



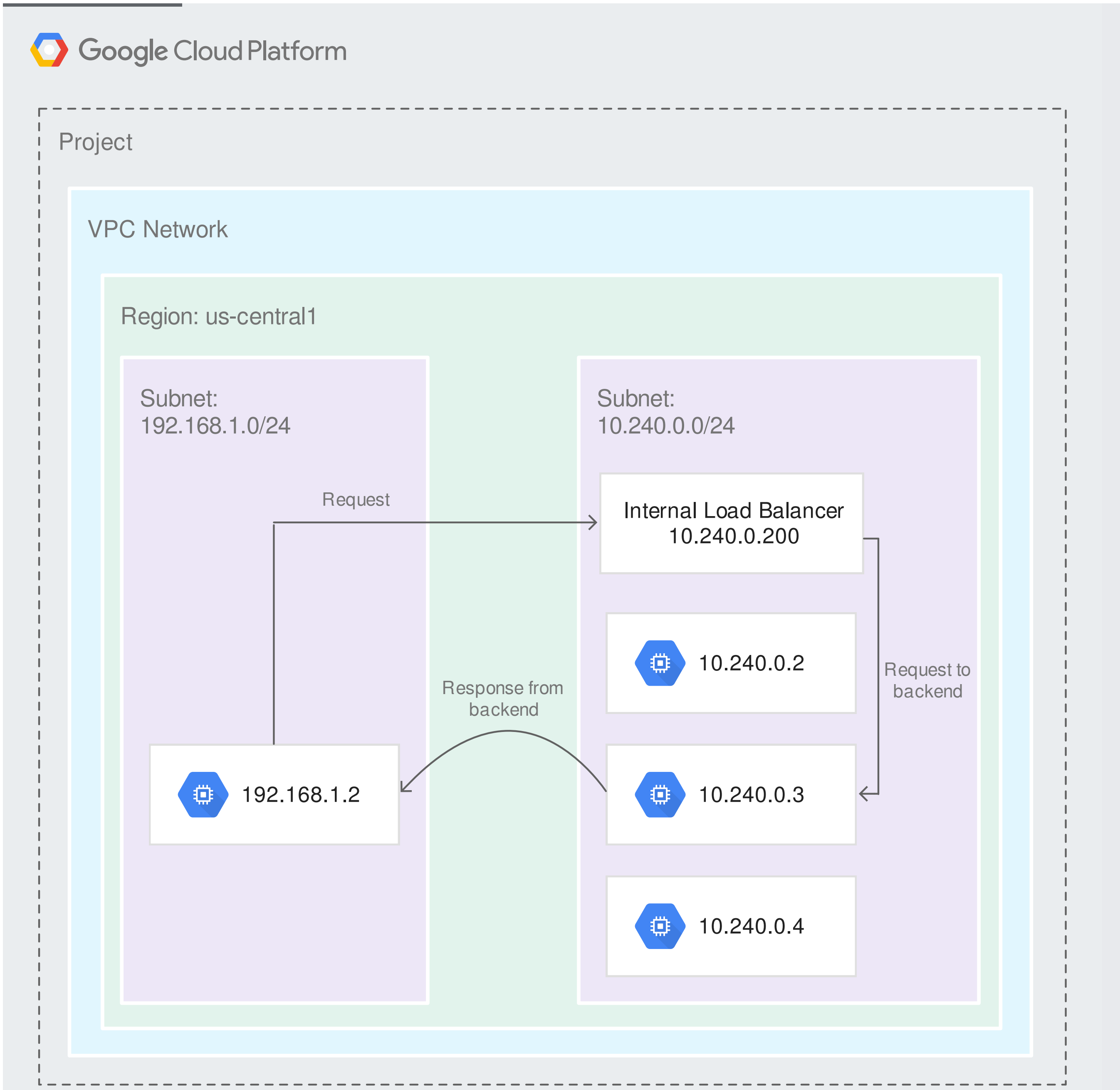
(/vpc/images/flow-logs/flow-logs-4.svg)
VPC Peering flows (click to enlarge)

Unless both VMs are in the same GCP project, VM-to-VM flows for peered VPCs are reported in the same way as for external endpoints—project and other annotation information for the other VM is not provided. If both VMs are in the same project, even if in different networks, then project and other annotation information is provided for the other VM as well.

In this example, the subnets of VM `10.10.0.2` in project analytics-prod and VM `10.50.0.2` in project webserver-test are connected through VPC Peering. If VPC Flow Logs is enabled in project analytics-prod, the traffic (1224 bytes) sent from `10.10.0.2` to `10.50.0.2` is reported from VM `10.10.0.2`, which is the source of the flow. The traffic (5342 bytes) sent from `10.50.0.2` to `10.10.0.2` is also reported from VM `10.10.0.2`, which is the destination of the flow.

In this example, VPC Flow Logs is not turned on in project webserver-test, so no logs are recorded by VM `10.50.0.2`.

reporter	connection.src_ip	connection.dest_ip	bytes_sent	VPC annotations
source	10.10.0.2	10.50.0.2	1224	src_instance.* src_vpc.*
destination	10.50.0.2	10.10.0.2	5342	dest_instance.* dest_vpc.*



(/vpc/images/flow-logs/flow-logs-5.svg)
Internal Load Balancing flows (click to enlarge)

When you add a VM to the backend service for an Internal Load Balancer, the Linux or Windows Guest Environment adds the IP address of the load balancer to the local routing table of the VM. This allows the VM to accept request packets with destinations set to the IP address of the load balancer. When the VM replies, it sends its response directly; however, the source IP address for the response packets is set to the IP address of the load balancer, not the VM being load balanced.

VM-to-VM flows sent through an internal load balancer are reported from both source and destination. For an example HTTP request / response pair, the following table explains the fields of the flow log entries observed. For the purpose of this illustration, consider the following network configuration:

- Browser instance at 192.168.1.2
- Internal load balancer at 10.240.0.200
- Webserver instance at 10.240.0.3

Traffic Direction	reporter	connection.src_ip	connection.dest_ip	connection.src_instance	connection.dest_instance
Request	SRC	192.168.1.2	10.240.0.200	Browser instance	
Request	DEST	192.168.1.2	10.240.0.3	Browser instance	Webserver instance
Response	SRC	10.240.0.3	192.168.1.2	Webserver instance	Browser instance
Response	DEST	10.240.0.200	192.168.1.2		Browser instance

The requesting VM does not know which VM will respond to the request. In addition, because the other VM sends a response with the internal load balancer IP as the source address, it does not know which VM *has* responded. For these reasons, the requesting VM cannot add `dest_instance` information to its report, only `src_instance` information. Because the responding VM does know the IP address of the other VM, it can supply both `src_instance` and `dest_instance` information.

When you enable VPC Flow Logs, you enable for all VMs in a subnet. You can also specify log sampling parameters when you turn on logging. See Log sampling and aggregation (#log-sampling) for details on the parameters you can control.

You can modify log sampling parameters. See Log sampling and aggregation (#log-sampling) for details on the parameters you can control.

Follow the access control guide for Stackdriver Logging (/logging/docs/access-control#permissions_and_roles).

View logs through the logs viewer page (/logging/docs/view/overview).

You need your project's project ID (/resource-manager/docs/creating-managing-projects#identifying_projects) for these commands.

1. Go to the Logs page in the Google Cloud Console.
   Go to the Logs page (https://console.cloud.google.com/logs/viewer)
2. Select **GCE Subnetwork** in the first pull-down menu.
3. Select **vpc_flows** in the second pull-down menu.
4. Click **OK**.

Alternatively:

1. Go to the Logs page in the Google Cloud Console.
   Go to the Logs page (https://console.cloud.google.com/logs/viewer)
2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.
3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID.
4. Click **Submit filter**.

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In the first pull-down menu, move the cursor to **GCE Subnetwork**, then move it to the right to open up the individual subnet selection menu.

3. In the second pull-down menu, select **vpc_flows**.

4. Click **OK**.

Alternatively:

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.

3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID and ***SUBNETWORK_NAME*** with your subnetwork.

4. Click **Submit filter**.

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.

3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID and ***VM_NAME*** with your VM.

4. Click **Submit filter**.

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.
3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID and ***SUBNETWORK_NAME*** with your subnetwork.
4. Click **Submit filter**.

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.
3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID, ***PORT*** with the port, and ***PROTOCOL*** with the protocol.
4. Click **Submit filter**.

1. Go to the Logs page in the Google Cloud Console.

   Go to the Logs page (https://console.cloud.google.com/logs/viewer)

2. In right side of the **Filter by label or text search** field, click the down arrow and select **Convert to advanced filter**.

3. Paste the following into the field. Replace ***PROJECT_ID*** with your project ID, ***PORT1*** and ***PORT2*** with the ports, and ***PROTOCOL*** with the protocol.

4. Click **Submit filter**.

You can export flow logs from Stackdriver Logging to a destination of your choice as described in the Stackdriver Logging documentation (/logging/docs/export/). Refer to the previous section for example filters.

- VPC flows are only supported for VPC network. If you have a legacy network (/vpc/docs/legacy), you will not see any logs.
- In Shared VPC (/vpc/docs/shared-vpc) networks, logs only appear in the host project, not the service projects. Make sure you look for the logs in the host project.
- Stackdriver Logging exclusion filters block specified logs. Make sure there are no exclusion rules that discard VPC Flow Logs.

1. Go to Resource usage (https://console.cloud.google.com/logs/usage).
2. Click the **Exclusions** tab.
3. Make sure there are no exclusion rules that might discard VPC Flow Logs.

- RTT measurements may be missing if not enough packets were sampled to capture RTT. This is more likely to happen for low volume connections.
- No RTT values are available for UDP flows.
- Some packets are sent with no payload. If header-only packets were sampled, the bytes value will be 0.

- Only UDP and TCP protocols are supported. VPC Flow Logs does not support any other protocols.
- Logs are sampled (#log-sampling). Some packets in very low volume flows might be missed.

Standard pricing for Stackdriver Logging, BigQuery, or Pub/Sub apply. VPC flow logs pricing is described in Network Telemetry pricing (/compute/network-pricing#network_telemetry).

- Does VPC Flow Logs include both allowed and denied traffic based on firewall rules?
  - VPC Flow Logs covers traffic from the perspective of a VM. All egress (outgoing) traffic from a VM is logged, even if it is blocked by an egress deny firewall rule. Ingress (incoming) traffic is logged if it is permitted by an ingress allow firewall rule. Ingress traffic blocked by an ingress deny firewall rule is not logged.
- Does VPC Flow Logs work with for VM instances with multiple interfaces?

  - Yes, you can enable VPC Flow Logs for all interfaces on a multiple interface VM (/vpc/docs/multiple-interfaces-concepts).

- Does VPC Flow Logs work with legacy networks?
  - No, VPC Flow Logs are not supported on legacy networks (/vpc/docs/legacy).

- View Stackdriver Logging (/logging/docs) documentation
- View Stackdriver Logging export (/logging/docs/export/configure_export_v2) documentation