Cloud NAT overview

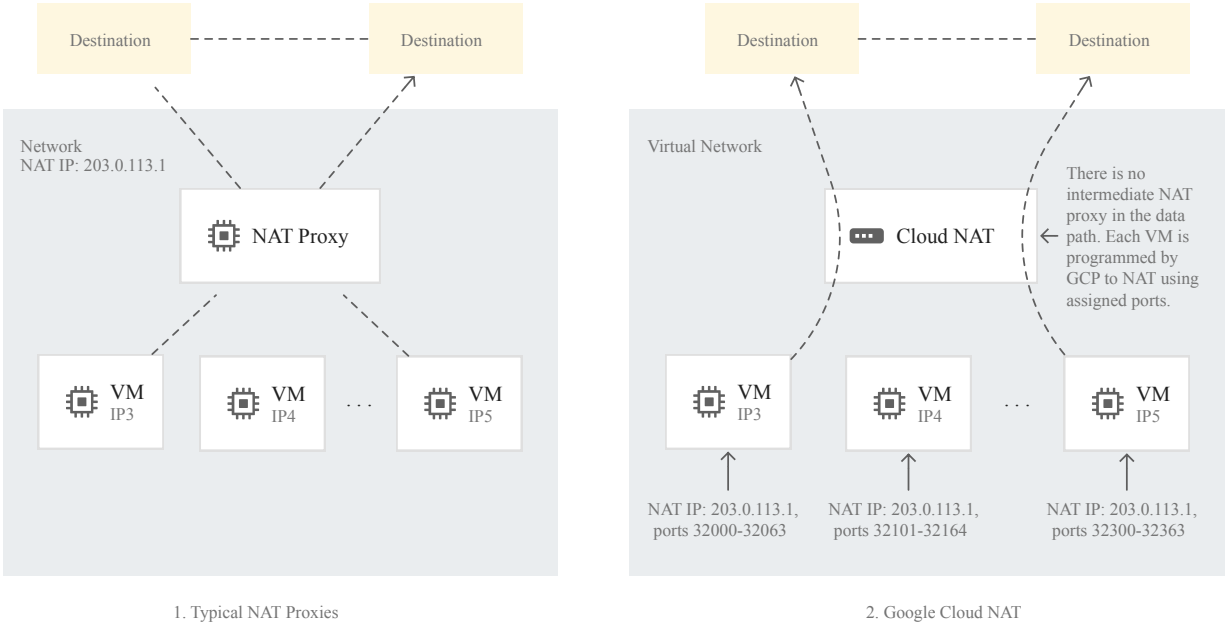
Cloud NAT (network address translation (https://www.wikipedia.org/wiki/Network_address_translation)) lets certain resources without external IP addresses create outbound connections to the internet.

Cloud NAT provides outgoing connectivity for the following resources:

- Compute Engine virtual machine (VM) instances without external IP addresses
- Private Google Kubernetes Engine (GKE) clusters
- Cloud Run instances through Serverless VPC Access (/vpc/docs/configure-serverless-vpc-access)
- Cloud Functions instances through Serverless VPC Access (/vpc/docs/configure-serverless-vpc-access)
- App Engine standard environment instances through Serverless VPC Access (/vpc/docs/configure-serverless-vpc-access)

Architecture

Cloud NAT is a distributed, software-defined managed service. It's not based on proxy VMs or appliances. Cloud NAT configures the Andromeda software (https://cloudplatform.googleblog.com/2014/04/enter-andromeda-zone-google-cloud-platforms-latest-networking-stack.html) that powers your Virtual Private Cloud (VPC) network so that it provides *source network address translation (source NAT or SNAT)* for VMs without external IP addresses. Cloud NAT also provides *destination network address translation (destination NAT or DNAT)* for established inbound response packets.



(/static/nat/images/07.svg)
Traditional NAT versus Cloud NAT (click to enlarge)

Cloud NAT implements outbound NAT in conjunction with static routes (/vpc/docs/routes#static_routes) in your VPC network whose next hops are the *default internet gateway*. In a basic configuration, a default route (/vpc/docs/routes#routingpacketsinternet) in your VPC network meets this requirement.

Cloud NAT does *not* implement unsolicited inbound connections from the internet. DNAT is only performed for packets that arrive as responses to outbound packets.

Benefits

Cloud NAT provides the following benefits:

- **Security**

  You can reduce the need for individual VMs to each have external IP addresses. Subject to egress firewall rules (/vpc/docs/firewalls), 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.

  If you use manual NAT IP address assignment (/nat/docs/ports-and-addresses#addresses) to configure a Cloud NAT gateway, you can confidently share a set of common external source IP addresses with a destination party. For example, a destination service might only allow connections from known external IP addresses.

- **Availability**

Cloud NAT is a distributed, software-defined managed service. It doesn't depend on any VMs in your project or a single physical gateway device. You configure a NAT gateway on a Cloud Router, which provides the control plane for NAT, holding configuration parameters that you specify. Google Cloud runs and maintains processes on the physical machines that run your Google Cloud VMs.

- **Scalability**

  Cloud NAT can be configured to automatically scale the number of NAT IP addresses that it uses, and it supports VMs that belong to managed instance groups, including those with autoscaling (/compute/docs/autoscaler) enabled.

- **Performance**

  Cloud NAT does not reduce the network bandwidth per VM. Cloud NAT is implemented by Google's Andromeda software-defined networking. For more information, see Network bandwidth (/compute/docs/network-bandwidth) in the Compute Engine documentation.

Specifications

Cloud NAT can be configured to provide NAT to the internet for packets sent from the following:

- The Compute Engine VM's network interface's primary internal IP address, provided that the network interface doesn't have an external IP address assigned to it. If the network interface has an external IP address assigned to it, Google Cloud automatically performs one-to-one NAT for packets whose sources match the interface's primary internal IP address because the network interface meets the Google Cloud internet access requirements (/vpc/docs/vpc#internet_access_reqs). The existence of an external IP address on an interface always takes precedence and always performs one-to-one NAT, without using Cloud NAT.
- An alias IP range (/vpc/docs/alias-ip) assigned to the VM's network interface. Even if the network interface has an external IP address assigned to it, you can configure a Cloud NAT gateway to provide NAT for packets whose sources come from an alias IP range of the interface. An external IP address on an interface never performs one-to-one NAT for alias IP addresses.
- For GKE clusters, Cloud NAT can provide service even if the cluster has external IP addresses in certain circumstances. For details, see GKE interaction (#NATwithGKE).

Cloud NAT allows outbound connections and the inbound responses to those connections. Each Cloud NAT gateway performs source NAT on egress, and destination NAT for established response packets.

Cloud NAT does *not* permit unsolicited inbound requests from the internet, even if firewall rules would otherwise permit those requests. For more information, see Applicable RFCs (#specs-rfcs).

Each Cloud NAT gateway is associated with a single VPC network, region, and Cloud Router. The Cloud NAT gateway and the Cloud Router provide a control plane—they are not involved in the data plane, so packets do not pass through the Cloud NAT gateway or Cloud Router.

Note: Even though a Cloud NAT gateway is managed by a Cloud Router, Cloud NAT does not use or depend on the Border Gateway Protocol (BGP).

Routes and firewall rules

Cloud NAT relies on custom static routes whose next hops are the default internet gateway. To fully utilize a Cloud NAT gateway, your VPC network needs a default route whose next hop is the default internet gateway. For more information, see routes interactions (#interaction-routes).

Cloud NAT does not have any Google Cloud firewall rule requirements. Firewall rules are applied directly to the network interfaces of Compute Engine VMs, not Cloud NAT gateways.

You don't have to create any special firewall rules that allow connections to or from NAT IP addresses. When a Cloud NAT gateway provides NAT for a VM's network interface, applicable egress firewall rules are evaluated as packets for that network interface *before* NAT. Ingress firewall rules are evaluated *after* packets have been processed by NAT.

Subnet IP address range applicability

A Cloud NAT gateway can provide NAT services for packets sent from a Compute Engine VM's network interface as long as that network interface doesn't have an external IP address assigned to it. For GKE clusters, Cloud NAT can provide service even if the cluster nodes have external IP addresses in certain circumstances. For details, see GKE interaction (#NATwithGKE).

The Cloud NAT gateway can be configured to provide NAT for the VM network interface's primary internal IP address, alias IP ranges, or both. You make this configuration by choosing the subnet IP address ranges (/vpc/docs/subnets#manually_created_subnet_ip_ranges) to which the gateway should apply.

Important: In Google Cloud, the terms *subnet* and *IP address range* are not synonyms. Each subnet has one primary IP address range, and, optionally, multiple secondary IP address ranges. For background information essential to understanding subnet IP address range applicability, see the VPC network overview (/vpc/docs/vpc).

You can configure a Cloud NAT gateway to provide NAT for the following:

- **Primary and secondary IP address ranges of all subnets in the region.** A single Cloud NAT gateway provides NAT for the primary internal IP addresses and all alias IP ranges of eligible VMs whose network interfaces use a subnet in the region. This option uses exactly one NAT gateway per region.
- **Primary IP address ranges of all subnets in the region.** A single Cloud NAT gateway provides NAT for the primary internal IP addresses and alias IP ranges from subnet primary IP address ranges of eligible VMs whose network interfaces use a subnet in the region. You can create additional Cloud NAT gateways in the region to provide NAT for alias IP ranges from subnet secondary IP address ranges of eligible VMs.
- **Custom subnet IP address ranges.** You can create as many Cloud NAT gateways as necessary, subject to Cloud NAT quotas and limits (/nat/quota#limits). You choose which subnet primary or secondary IP address ranges should be served by each gateway.

Bandwidth

Using a Cloud NAT gateway does not change the amount of outbound or inbound bandwidth that a VM can use. For bandwidth specifications, which vary by machine type, see Network bandwidth (/compute/docs/network-bandwidth) in the Compute Engine documentation.

VMs with multiple network interfaces

If you configure a VM with multiple network interfaces (/vpc/docs/multiple-interfaces-concepts), each interface must be in a separate VPC network. Consequently, the following is true:

- A Cloud NAT gateway can only apply to a single network interface of a VM. Separate Cloud NAT gateways can provide NAT to the same VM, where each gateway applies to a separate interface.
- One interface of a multiple network interface VM can have an external IP address, and thus be ineligible for Cloud NAT, while another one of its interfaces can be eligible for NAT if that interface doesn't have an external IP address, and you've configured a Cloud NAT gateway to apply to the appropriate subnet IP address range.

NAT IP addresses and ports

When you create a Cloud NAT gateway, you can choose to have the gateway automatically allocate regional external IP addresses. Alternatively, you can manually assign a fixed number of regional external IP addresses to the gateway. For details about each method, see NAT IP addresses (/nat/docs/ports-and-addresses#addresses).

You can configure the number of source ports that each Cloud NAT gateway reserves to each VM for which it should provide NAT services. You can configure static port allocation (/nat/docs/ports-and-addresses#static-port), where the same number of ports are reserved for each VM, or dynamic port allocation (/nat/docs/ports-and-addresses#dynamic-port), where the number of reserved ports can vary between the minimum and maximum limits that you specify.

The VMs for which NAT should be provided are determined by the subnet IP address ranges (#specs-subnet-ranges) that the gateway is configured to serve.

For more information, see Ports (/nat/docs/ports-and-addresses#ports) and the Port reservation procedure.

Applicable RFCs

Cloud NAT supports *Endpoint-Independent Mapping* and *Endpoint-Dependent Filtering* as defined in RFC 5128 (https://tools.ietf.org/html/rfc5128). You can enable or disable Endpoint-Independent Mapping. By default, Endpoint-Independent Mapping is disabled when you create a NAT gateway.

Endpoint-Independent Mapping means that if a VM sends packets from a given internal IP address and port pair to multiple different destinations, then the gateway maps all of those packets to the same NAT IP address and port pair, regardless of the destination of the packets. For details and implications pertinent to Endpoint-Independent Mapping, see

Simultaneous port reuse and Endpoint-Independent Mapping (/nat/docs/ports-and-addresses#ports-reuse-endpoints).

Endpoint-Dependent Filtering means that response packets from the internet are allowed to enter only if they are from an IP address and port that a VM had already sent packets to. The *filtering* is endpoint-dependent regardless of Endpoint Mapping type. This feature is always on and not user configurable.

For more information about the relationship between ports and connections, see Ports and connections (/nat/docs/ports-and-addresses#ports-and-connections) and the NAT flow example (/nat/docs/ports-and-addresses#snat-flow-example).

Cloud NAT is a *Port Restricted Cone NAT* as defined in RFC 3489 (https://www.ietf.org/rfc/rfc3489.txt).

NAT traversal

If Endpoint-Independent Mapping is enabled, Cloud NAT is compatible with common NAT traversal protocols such as STUN and TURN, if you deploy your own STUN or TURN servers:

- STUN (Session Traversal Utilities for NAT, RFC 5389 (https://tools.ietf.org/html/rfc5389)) allows direct communication between VMs behind NAT when a communication channel is established.
- TURN (Traversal Using Relays around NAT, RFC 5766 (https://tools.ietf.org/html/rfc5766)) permits communication between VMs behind NAT by way of a third server where that server has an external IP address. Each VM connects to the server's external IP address, and that server relays communication between the two VMs. TURN is more robust, but consumes more bandwidth and resources.

NAT timeouts

Cloud NAT gateways use the following timeouts. You can modify the default timeout values to decrease or increase the rate at which ports are reused. Each timeout value is a balance between efficient use of Cloud NAT resources and possible disruption to active connections, flows, or sessions.

Timeout	Description	Cloud NAT default	Configurable
UDP Mapping Idle Timeout RFC 4787 (https://tools.ietf.org/html/rfc4787)	Specifies the time in seconds after which UDP flows must stop sending traffic to endpoints so	30 seconds	Yes

REQ-5	that the Cloud NAT mappings are removed. UDP Mapping Idle Timeout affects two endpoints that stop sending traffic to each other. It also affects endpoints that take longer to respond, or if there is increased network latency. You can increase the specified timeout value to decrease the rate at which ports can be reused. The larger timeout value means that the ports are held for longer connections and also protects against pauses in traffic over a specific UDP socket.		
TCP Established Connection Idle Timeout RFC 5382 (https://tools.ietf.org/html/rfc5382) REQ-5	Specifies the time in seconds that a connection is idle before the Cloud NAT mappings are removed. TCP Established Connection Idle Timeout affects endpoints that take longer to respond, or if there is increased network latency. You can increase the timeout value when you want to open TCP connections and keep the connections open for a long time without a keepalive mechanism in place.	1200 seconds (20 minutes)	Yes
TCP Transitory Connection Idle Timeout RFC 5382 (https://tools.ietf.org/html/rfc5382) REQ-5	Specifies the time in seconds that TCP connections can remain in the half-open state before the Cloud NAT mappings can be deleted. TCP Transitory Connection Idle Timeout affects an endpoint when an external endpoint takes a longer period than the specified time, or when there is increased network latency. Unlike the TCP Established	30 seconds **Note:** Regardless of the value that you set for this timeout, Cloud NAT might require up to an additional 30 seconds before a Cloud NAT source IP address and source port tuple can be used to	Yes

	Connection Idle Timeout, the TCP Transitory Connection Idle Timeout affects only half-open connections.	process a new connection.	
TCP TIME_WAIT Timeout RFC 5382 (https://tools.ietf.org/html/rfc5382) REQ-5	Specifies the time in seconds that a fully closed TCP connection is retained in the Cloud NAT mappings after the connection expires. TCP TIME_WAIT Timeout protects your internal endpoints from receiving invalid packets that belong to a closed TCP connection that are retransmitted. You can decrease the timeout value to improve the reuse of Cloud NAT ports at the cost of possibly receiving retransmitted packets from an unrelated, previously closed connection.	120 seconds **Note:** Regardless of the value that you set for this timeout, Cloud NAT might require up to an additional 30 seconds before a Cloud NAT source IP address and source port tuple can be used to process a new connection.	Yes
ICMP Mapping Idle Timeout RFC 5508 (https://tools.ietf.org/html/rfc5508) REQ-2	Specifies the time in seconds after which Internet Control Message Protocol (ICMP) Cloud NAT mappings that don't have any traffic flows are closed. ICMP Mapping Idle Timeout affects an endpoint when the endpoint takes a longer to respond than the specified time, or when there is increased network latency.	30 seconds	Yes

Product interactions

The following sections describe important interactions between Cloud NAT and other Google Cloud products.

Routes interactions

A Cloud NAT gateway can only use routes whose next hops are the default internet gateway. Each VPC network starts with a default route whose destination is `0.0.0.0/0` and whose next hop is the default internet gateway. For important background information, see the routes overview (/vpc/docs/routes).

The following examples illustrate situations that could cause Cloud NAT gateways to become inoperable:

- If you create a custom static route with next hops set to any other type of custom static route next hop (/vpc/docs/routes#static-route-next-hops), packets with destination IP addresses matching the destination of the route are sent to that next hop instead of to the default internet gateway. For example, if you use VM instances running NAT, firewall, or proxy software, you would necessarily create custom static routes to direct traffic to those VMs as the next hop. The next-hop VMs require external IP addresses. Thus, neither traffic from the VMs that rely upon the next-hop VMs nor the next-hop VMs themselves could use Cloud NAT.
- If you create a custom static route whose next hop is a Cloud VPN tunnel, Cloud NAT does not use that route. For example, a custom static route with destination `0.0.0.0/0` and a next hop Cloud VPN tunnel directs traffic to that tunnel, not to the default internet gateway. Consequently, Cloud NAT gateways would not be able to use that route. This holds true even for more specific destinations, including `0.0.0.0/1` and `128.0.0.0/1`.
- If an on-premises router advertises a custom dynamic route to a Cloud Router managing a Cloud VPN tunnel or Cloud Interconnect attachment (VLAN), Cloud NAT gateways cannot use that route. For example, if your on-premises router advertises a custom dynamic route with destination `0.0.0.0/0`, `0.0.0.0/0` would be directed to the Cloud VPN tunnel or Cloud Interconnect attachment (VLAN). This holds true even for more specific destinations, including `0.0.0.0/1` and `128.0.0.0/1`.

Private Google Access interaction

Cloud NAT never performs NAT for traffic sent to the select external IP addresses for Google APIs and services (/vpc/docs/configure-private-google-access#config-domain). Instead, Google Cloud automatically enables Private Google Access for a subnet IP address range when you configure a Cloud NAT gateway to apply to that subnet range (#specs-subnet-ranges), either primary or secondary. As long as the gateway provides NAT for a subnet's range, Private Google Access is in effect for that range and cannot be disabled manually.

A Cloud NAT gateway does *not* change the way that Private Google Access works. For more information, see Private Google Access (/vpc/docs/private-access-options).

Shared VPC interaction

Shared VPC (/vpc/docs/shared-vpc) enables multiple service projects in a single organization to use a common, Shared VPC network in a host project. To provide NAT for VMs in service projects that use a Shared VPC network, you must create Cloud NAT gateways in the host project.

VPC Network Peering interaction

Cloud NAT gateways are associated with subnet IP address ranges in a single region and a single VPC network. A Cloud NAT gateway created in one VPC network cannot provide NAT to VMs in other VPC networks connected by using VPC Network Peering (/vpc/docs/vpc-peering), even if the VMs in peered networks are in the same region as the gateway.

GKE interaction

A Cloud NAT gateway can perform NAT for nodes and Pods in a private cluster (/kubernetes-engine/docs/concepts/private-cluster-concept), which is a type of VPC-native cluster. The Cloud NAT gateway must be configured to apply to at least the following subnet IP address ranges for the subnet that your cluster uses:

- Subnet primary IP address range (used by nodes)
- Subnet secondary IP address range used for Pods in the cluster
- Subnet secondary IP address range used for Services in the cluster

The simplest way to provide NAT for an entire private cluster is to configure a Cloud NAT gateway to apply to all of the cluster's subnet's IP address ranges.

For background information about how VPC-native clusters utilize subnet IP address ranges, see IP ranges for VPC-native clusters (/kubernetes-engine/docs/concepts/alias-ips#cluster_sizing).

When a Cloud NAT gateway is configured to provide NAT for a private cluster, it reserves NAT source IP addresses and source ports for each node VM. Those NAT source IP addresses and source ports are usable by Pods because Pod IP addresses are implemented as alias IP ranges assigned to each node VM.

GKE VPC-native clusters always assign each node an alias IP range that contains more than one IP address (netmask smaller than `/32`).

- If static port allocation (/nat/docs/ports-and-addresses#static-port) is configured, the Cloud NAT port reservation procedure (/nat/docs/ports-and-addresses#port-reservation-procedure) reserves at least 1,024 source ports per node. If the specified value for minimum ports per VM is greater than 1,024, that value is used.
- If dynamic port allocation (/nat/docs/ports-and-addresses#dynamic-port) is configured, the specified value for minimum ports per VM is initially allocated per node. The number of allocated ports subsequently varies between the specified values for minimum and maximum ports per VM, based on demand.

For information about Pod IP address ranges and VPC-native clusters, see Subnet secondary IP address range for Pods (/kubernetes-engine/docs/concepts/alias-ips#cluster_sizing_secondary_range_pods).

Independent of Cloud NAT, GKE performs SNAT by using software running on each node when Pods send packets to the internet, unless you've changed the cluster's IP masquerade configuration (/kubernetes-engine/docs/how-to/ip-masquerade-agent). If you need granular control over egress traffic from Pods, you can use a network policy (/kubernetes-engine/docs/how-to/network-policy).

Under certain circumstances, Cloud NAT can be useful to non-private VPC-native clusters as well. Because the nodes in a non-private cluster have external IP addresses, packets sent from the node's primary internal IP address are never processed by Cloud NAT. However, packets sent from Pods in a non-private cluster can be processed by a Cloud NAT gateway if both of the following are true:

- For VPC-native clusters, the Cloud NAT gateway is configured to apply to the secondary IP address range for the cluster's Pods.
- The cluster's IP masquerade configuration is not configured to perform SNAT within the cluster for packets sent from Pods to the internet.

Cloud Load Balancing interactions

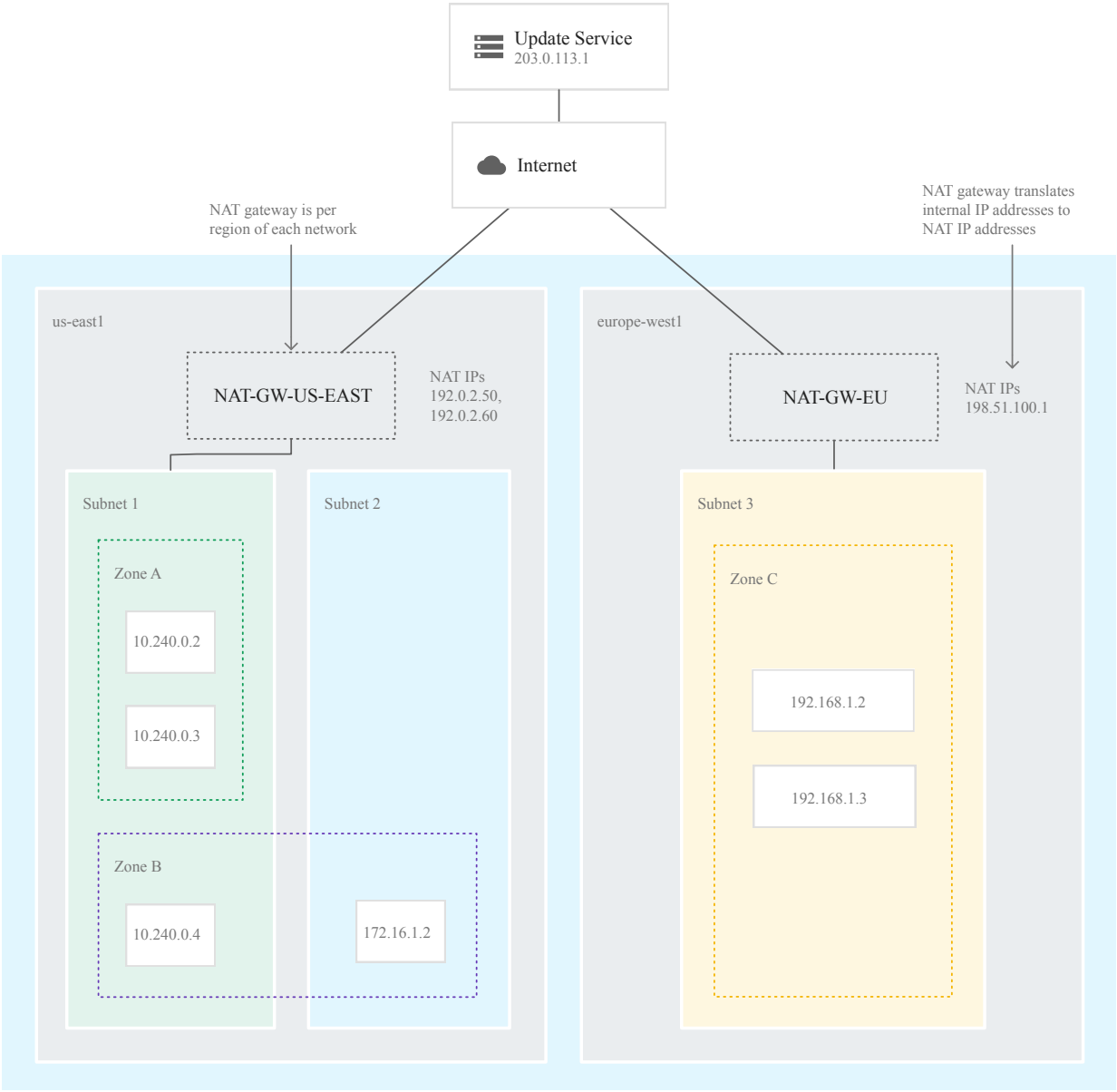
Google Cloud external load balancers and health check systems communicate with VMs by using special routes (/vpc/docs/routes#special-lb-paths). Backend VMs do not require external IP addresses nor does a Cloud NAT gateway manage communication for load balancers and health checks. For more information, see Cloud Load Balancing overview

(/load-balancing/docs/load-balancing-overview) and Health checks overview (/load-balancing/docs/health-check-concepts).

Examples

The following examples illustrate Cloud NAT concepts.

Basic NAT configuration

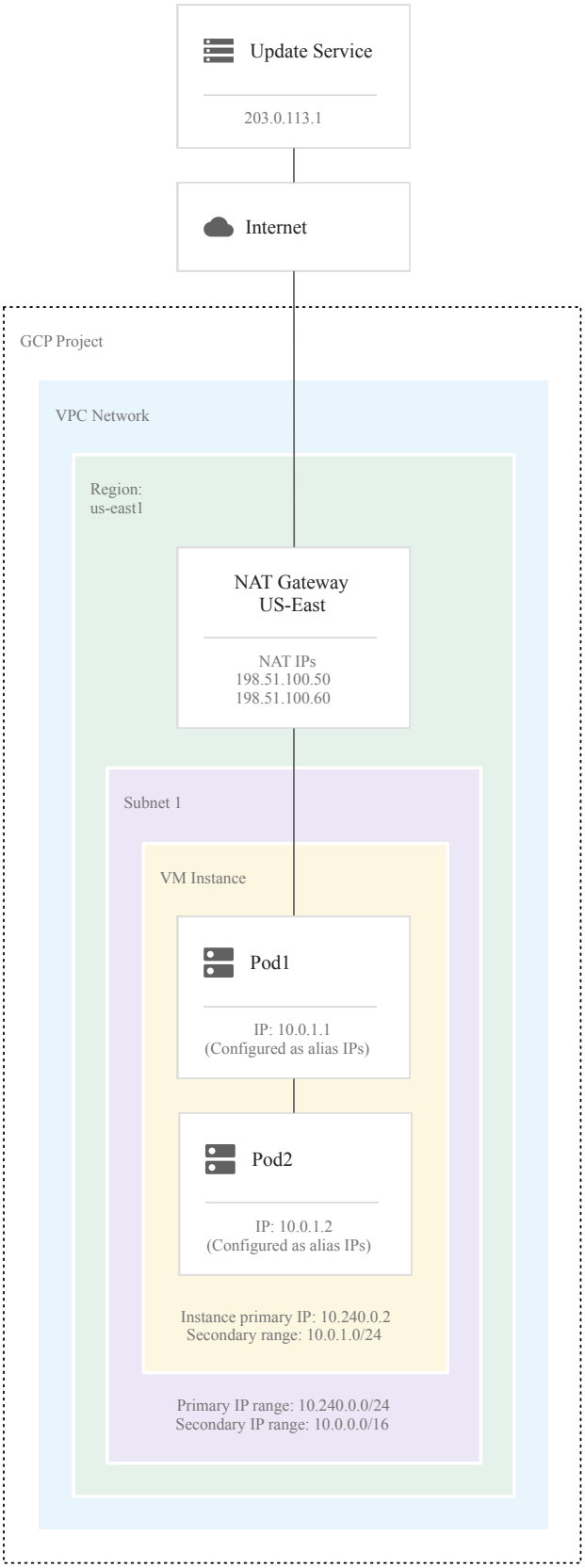


(/static/nat/images/01.svg)
Cloud NAT (click to enlarge)

In this example:

- The `nat-gw-us-east` gateway is configured to apply to the primary IP address range of `subnet-1` in the `us-east1` region. A VM whose network interface does not have an external IP address can send traffic to the internet by using either its primary internal IP address or an alias IP range from the primary IP address range of `subnet-1`, `10.240.0.0/16`.
- A VM whose network interface does not have an external IP address and whose primary internal IP address is located in `subnet-2` cannot access the internet because no Cloud NAT gateway applies to any IP address range of that subnet.
- The `nat-gw-eu` gateway is configured to apply to the primary IP address range of `subnet-3` in the `europe-west1` region. A VM whose network interface does not have an external IP address can send traffic to the internet by using either its primary internal IP address or an alias IP range from the primary IP address range of `subnet-3`, `192.168.1.0/24`.

GKE example



(/static/nat/images/03.svg)

Cloud NAT with GKE (click to enlarge)

In this example, you want your containers to be NAT-translated. To enable NAT for all the containers and the GKE node, you must choose all the IP address ranges of Subnet 1 as the

NAT candidates. It is not possible to enable NAT for only `container1` or `container2`.

What's next

- Learn about Cloud NAT addresses and ports (/nat/docs/ports-and-addresses).
- Create your own Cloud NAT gateway (/nat/docs/set-up-network-address-translation).
- Learn about Cloud NAT rules (/nat/docs/nat-rules-overview).
- Create an example Compute Engine setup (/nat/docs/gce-example).
- Create an example GKE setup (/nat/docs/gke-example).
- Troubleshoot common issues (/nat/docs/troubleshooting).

Last updated 2023-01-24 UTC.