



CONFIGURING PRIVATE GOOGLE ACCESS AND CLOUD NAT USING GCP



A MINI PROJECT REPORT

Submitted by

DEEPA R (731620104006)

DURGADEVI K S (731620104014)

KAVISHNI S (731620104028)

YAMUNA K (731620104061)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

K S R INSTITUTE FOR ENGINEERING AND TECHNOLOGY

TIRUCHENGODE – 637 215

ANNA UNIVERSITY : CHENNAI 600 025

MAY 2023

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report “**CONFIGURING PRIVATE GOOGLE ACCESS AND CLOUD NAT USING GCP**” is the bonafide work of “**DEEPA R (731620104006), DURGADEVI K S(731620104014), KAVISHNI S (731620104028), YAMUNA K (731620104061)**” who carried out the project work under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr.M.VIMALADEVI M.E., Ph.D

ASSOCIATE PROFESSOR & HEAD

Department of Computer Science and
Engineering,

K S R Institute for Engineering and
Technology,

Tiruchengode-637215.

SIGNATURE

SUPERVISOR

Mrs.P.NITHYA M.E.,

ASSISTANT PROFESSOR

Department of Computer Science and
Engineering,

K S R Institute for Engineering and
Technology,

Tiruchengode-637215.

Submitted for the Mini Project work Viva-Voce held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved founder and chairman, **Theivathiru. Dr. K. S. RANGASAMY, MJF**, K S R Educational Institutions. We thank our vice chairman **Thiru. R. SRINIVASAN, B.B.M., M.I.S.T.E.**, K S R Institute for Engineering and Technology, for his inspiration and moral support.

We express our heartfelt thanks to our principal, **Dr. M. VENKATESAN M.E., Ph.D., M.I.S.T.E.**, of K S R Institute for Engineering and Technology for his Valuable support.

We express our sincere thanks to **Dr. M.VIMALADEVI M.E., Ph.D.**, Head of the Department of Computer Science and Engineering, K S R Institute for Engineering and Technology, for her encouragement and support throughout the course and for her invaluable guidance, immense help, encouragement and provide us necessary facilities throughout this project.

We express our extreme gratitude to our project coordinator **Mr.V. PRAKASHAM M.Tech.**, Assistant Professor, Department of Computer Science and Engineering, for providing us kind advice during the development of the project.

We wish to express our profound gratitude and thanks to our Guide **Mrs.P.NITHYA M.E.**, Assistant Professor, Department of Computer Science and Engineering, K S R Institute for Engineering and Technology, for her valuable guidance, immense help, encouragement and providing us necessary facilities throughout this project.

ABSTRACT

Private Google Access allows resources within a VPC network to access Google Cloud services privately without requiring public IP addresses. Cloud NAT enables instances within private networks to communicate with the internet while using private IP addresses. To configure Private Google Access, the first step involves creating a Virtual Private Cloud (VPC) network. Within the VPC network, subnets need to be created, specifying the appropriate IP ranges for the private addresses. To enable Private Google Access, the "Private Google Access" option should be enabled for each subnet that requires access to Google APIs and services.

In order to establish internet connectivity for instances within the private network, Cloud NAT can be set up. This involves creating a NAT gateway and configuring NAT services within the VPC network. The NAT gateway serves as a bridge between the private instances and the internet, translating private IP addresses to public ones for outbound traffic. To configure Cloud NAT, a subnet needs to be designated as the NAT subnet, and an external IP address must be allocated for the NAT gateway. Firewall rules can be defined to control traffic between the private instances and the internet. Additionally, Cloud Router can be configured to allow dynamic routing for the NAT gateway.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	TABLE OF CONTENTS	v
	LIST OF FIGURES	vii
	LIST OF ABBREVIATIONS	viii
1	INTRODUCTION	9
	1.1 AIM OF THE PROJECT	9
	1.2 PROBLEM STATEMENT	10
	1.3 OBJECTIVES	11
2	LITERATURE SURVEY	12
3	SYSTEM ANALYSIS	14
	3.1 EXISTING SYSTEM	14
	3.1.1 Disadvantages	15
	3.2 PROPOSED SYSTEM	15
	3.2.1 Advantages	15
4	SYSTEM SPECIFICATION	17
	4.1 HARDWARE REQUIREMENTS	17
	4.2 SOFTWARE REQUIREMENTS	17
5	MODULE DESCRIPTION	18
	5.1 MODULES	18
	5.2 SCREENSHOTS	32
6	SYSTEM TESTING	37
	6.1 TESTING	37

	6.2 TYPES OF TESTING	37
	6.2.1 Connectivity Test	37
	6.2.2. Private Google Access Test	38
	6.2.3. Traffic Inspection Test	38
	6.2.4. Load and Scalability Test	38
	6.2.5. Failover Test	38
	6.2.6. Security Test	38
	6.2.7. Logging and Monitoring Test	39
	6.3 BLOCK DIAGRAM	40
7	CONCLUSION	41
	7.1 FUTURE ENHANCEMENT	42
	7.2 REFERENCES	43

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
5.2.1	OPEN GOOGLE CONSOLE	32
5.2.2	CREATING THE VM INSTANCE	33
5.2.3	ENABLING PRIVATE GOOGLE ACCESS	34
5.2.4	CONFIGURING A CLOUD NAT GATEWAY	35
5.2.5	CONFIGURING AND VIEWING LOGS WITH CLOUD NAT LOGGING	36
6.3.1	INTERNET ACCESSSS OF CLOUD NAT IN GCP	40
6.3.2	CLOUD NAT WITH VPC PEERING	40

LIST OF ABBREVIATIONS

GCP	GOOGLE CLOUD PLATFORM
NAT	NETWORK ADDRESS TRANSLATION
VPC	VIRTUAL PRIVATE CLOUD
VM	VIRTUAL MACHINE
IP	INTERNET PROTOCOL
API	APPLICATION PROGRAM INTERFACE
SSH	SECURE SHELL
SaaS	SOFTWARE AS A SERVICE

CHAPTER 1

INTRODUCTION

1.1 AIM OF THE PROJECT

Google Cloud Platform (GCP) offers a robust set of networking features that allow you to configure secure and efficient communication within your cloud environment. Two important components of this networking infrastructure are Private Google Access and Cloud NAT (Network Address Translation). In this introduction, we will explore these two concepts and discuss how to configure them in GCP. Private Google Access enables your virtual machine (VM) instances to access Google APIs and services without requiring a public IP address. By default, VMs in GCP use public IP addresses to access Google services, but this can pose security risks and increase exposure to the public internet. Private Google Access provides an alternative by allowing VMs to access Google services privately using internal IP addresses.

When we enable Private Google Access on a subnet, the VMs within that subnet can access Google Cloud services such as Cloud Storage, Big Query, and Cloud SQL over an internal network connection. This helps to maintain a more secure network environment by reducing the attack surface and limiting exposure to the internet. Cloud NAT allows your VMs that do not have public IP addresses to access the internet for updates, package installations, and other outbound connectivity needs. It provides outbound NAT capabilities, enabling private instances to communicate with external resources over the internet without exposing their internal IP addresses. By using Cloud NAT, you can conserve public IP addresses, reduce network management overhead, and simplify networking configurations. Cloud NAT automatically scales based on your needs, ensuring high availability performance for outbound connections.

1.2 PROBLEM STATEMENT

Organizations implementing firewalls as part of their network security infrastructure face several challenges in effectively utilizing and managing these security devices. Configuring firewalls can be complex and time-consuming, especially when dealing with large networks and multiple security zones. Organizations struggle to define and implement firewall rules that accurately reflect their security policies and adequately protect their network assets. The complexity often leads to misconfigurations, rule conflicts, and unintended security gaps. Organizations often face challenges in gaining comprehensive visibility into network traffic and understanding the specific requirements of different applications and services. This lack of visibility hinders the ability to define granular firewall rules that appropriately allow or restrict traffic based on application, user, or other contextual factors.

As a result, there may be overly permissive rules or unnecessary restrictions that impact network functionality and security. As organizations adopt cloud services and hybrid infrastructure models, integrating firewalls into these environments becomes complex. Ensuring consistent security policies across on-premises and cloud environments, managing traffic flows between different networks, and maintaining visibility and control over distributed resources pose significant challenges. Addressing these challenges requires a comprehensive approach to firewall management. Organizations need tools, processes, and expertise to simplify firewall configuration, enhance visibility into network traffic, streamline rule management, and ensure consistent security policies across diverse environments. It is essential to strike a balance between robust security measures and maintaining network performance and user experience.

1.3 OBJECTIVES

The primary objective of configuring Private Google Access and Cloud NAT in Google Cloud Platform (GCP) is to establish secure and efficient networking capabilities within the cloud environment. The specific objectives include: By configuring Private Google Access, the objective is to enable virtual machine (VM) instances to securely access Google services using internal IP addresses. This reduces exposure to the public internet, minimizing potential attack vectors and enhancing the overall security posture of the cloud infrastructure. To ensure that VMs within specific subnets can communicate privately with various Google Cloud services such as Cloud Storage, BigQuery, and Cloud SQL.

Private Google Access enables VMs to access these services securely without requiring public IP addresses, enhancing data privacy and confidentiality. The objective of configuring Cloud NAT is to enable private instances without public IP addresses to access the internet for outbound connectivity needs. Cloud NAT provides outbound Network Address Translation capabilities, allowing private instances to communicate with external resources over the internet while masking their internal IP addresses. The goal is to establish a reliable and efficient outbound connectivity solution while conserving public IP addresses. The objective is to streamline network configurations and reduce management overhead. Private Google Access and Cloud NAT provide mechanisms to simplify networking setups by eliminating the need for public IP addresses for internal communication and centralizing outbound connectivity for private instances.

CHAPTER 2

LITERATURE SURVEY

[1] ENHANCING THE SECURITY OF A PRIVATE NETWORK BY USING A MULTI-LEVEL HIERARCHICAL NAT SCHEME (Shie-Yuan Wang ,Yu-Hsun Yuan in 2022)

Nowadays, attacks coming from the Internet are posing serious threats to the hosts in an institution, campus, company, etc. The Network Address Translator (NAT) is a device that allows a host in a private network to interact with the hosts on the public Internet. Due to the property of NAT, unless a host that is behind a NAT actively contacts a host on the Internet, hosts on the Internet cannot actively reach the host behind the NAT. In this work, we exploit NATs and propose a multi-level hierarchical NAT scheme to protect and enhance the security of a private network. We have designed and implemented our scheme over P4 programmable hardware switches. Experimental results show that our scheme functions correctly and provides high throughput, low latency, and high stability. In addition, according to our tests, our scheme works correctly with most existing network applications .

[2] NAT EXPERIMENTAL DESIGN AND RESULT ANALYSIS BASED ON SIMULATION TECHNOLOGY (Cao Huamei School of Management, Tianjin University of Technology, Tianjin, China in 2022)

NAT technology can solve the problem of IP address resource shortage, and it is widely used in the actual network construction. This paper first completes the DESIGN of NAT experimental project of an enterprise network, then realizes the simulation design by using Cisco simulator, and finally analyzes the results such as connectivity test, NAT address translation table analysis and IP data report analysis, and expands

the working principle of NAT intuitively. This research can provide students with the opportunity of self-exploration, help students to understand the theoretical knowledge in depth, is a useful exploration of undergraduate education reform.

[3] HYPERNAT:SCALING UP NETWORK ADDRESS TRANSLATION WITH SMARTNICs FOR CLOUDS (ShaokeFang Qingsong Liu Wenfei Wu in 2021)

Network address translation (NAT) is a basic functionality in cloud gateways. With the increasing traffic volume and number of flows introduced by the cloud tenants, the NAT gateway needs to be implemented on a cluster of servers. We propose to scale up the gateway servers, which could reduce the number of servers so as to reduce the capital expense and operation expense. We design HyperNAT, which leverages smartNICs to improve the server's processing capacity. In HyperNAT, the NAT functionality is distributed on multiple NICs, and the flow space is divided and assigned accordingly. HyperNAT overcomes the challenge that the packets in two directions of one connection need to be processed by the same NAT rule (named two-direction consistency, TDC) by cloning the rule to both data paths of the two directions.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

A firewall is a network security device that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as a barrier between internal networks (such as a company's private network) and external networks (such as the Internet) to protect the internal network from unauthorized access and potential threats. Firewalls have been a first line of defense in network security for over 25 years. They establish a barrier between secured and controlled internal networks that can be trusted and untrusted outside networks, such as the Internet. A firewall can be hardware, software, software-as-a service (SaaS), public cloud, or private cloud (virtual). A firewall is a security product that filters out malicious traffic. Traditionally, firewalls have run in between a trusted internal network and an untrusted network – e.g., between a private network and the Internet. Early firewalls were physical appliances that connected to an organization's on-premise infrastructure.

Firewalls block and allow network traffic according to an internal set of rules. Most firewalls allow administrators to customize these rules. Typically, a firewall allows no direct connection between the internal network and the Internet. Instead, external connection requests, or digital packets, may be routed to a heavily secured “bastion host” server designed to withstand attack or to a larger “demilitarized zone,” a controlled network between the internal network and the outside. A firewall can regulate access going either to or from the internal network; for instance, some companies use a firewall to block employee access to certain public websites.

3.1.1 Disadvantages

Firewalls are an essential component of network security, but they also have some disadvantages and limitations. Here are a few disadvantages of firewalls:

- 1.False sense of security
- 2.Incomplete protection
- 3.Over-restrictive or under-restrictive policies
- 4.Performance impact
- 5.Encrypted traffic challenges
- 6.Advanced evasion techniques

3.2 PROPOSED SYSTEM

Google Cloud's Network Address Translation (NAT) service enables you to provision your application instances without public IP addresses while also allowing them to access the internet for updates, patching, config management, and more in a controlled and efficient manner. In this project, we will configure Private Google Access and Cloud NAT for a VM instance that doesn't have an external IP address. Then, you will verify access to public IP addresses of Google APIs and services and other connections to the internet. Finally, we will use Cloud NAT logging to record connections made in your gateway.

3.2.1 Advantages

- Cloud NAT allows you to easily manage your IP addresses by providing a translation layer between your private IP addresses and the public internet.
- By using Cloud NAT, you can hide the private IP addresses of your resources from the public internet. This adds an extra layer of security by reducing the exposure of your internal network infrastructure to potential attacks and unauthorized access.
- Cloud NAT enables instances that have private IP addresses to

communicate with the internet. It provides outbound internet connectivity for instances in private subnets without requiring public IP addresses on each instance, which can help improve network security.

- Cloud NAT can help reduce costs associated with IP address allocation. Instead of assigning a public IP address to each instance, you can use a smaller pool of shared public IP addresses provided by Cloud NAT.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

This section gives the details and Specification of the hardware on which the system is expected to work.

Processor	:	Intel dual core processor
RAM	:	2GBSD RAM
Monitor	:	17"Color
Harddisk	:	500GB
Keyboard	:	Standard102 Keys
Mouse	:	Optical mouse

4.2 SOFTWARE REQUIREMENTS

This section gives the details of the software that are used for the development.

Operating System	:	Windows10
Environment	:	Google Cloud Skills Boost
Commands	:	Google Cloud Commands

CHAPTER 5

MODULE DESCRIPTION

5.1 MODULES

Module 1 Create the VM instances

We will now create one VM instance that has no external IP address and another VM instance to serve as a bastion host.

Create a VPC network and firewall rules

First, create a VPC network for the VM instances and a firewall rule to allow SSH access.

1. In the Cloud Console, on the **Navigation menu** (≡), click **VPC network > VPC networks**.
2. Click **Create VPC Network**.
3. For **Name**, type **privatenet**.
4. For **Subnet creation mode**, click **Custom**.
5. Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Name	privatenet-us
Region	us-central1
IP address range	10.130.0.0/20

6. Click **Done**.

7. Click **Create** and wait for the network to be created.
8. In the left pane, click **Firewall rules**.
9. Click **Create Firewall Rule**.
10. Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Name	privatenet-allow-ssh
Network	Privatenet
Targets	All instances in the network
Source filter	IP ranges
Source IP ranges	0.0.0.0/0
Protocols and ports	Specified protocols and ports

11. For **tcp**, specify port **22**.

12. Click **Create**.

Create the VM instance with no public IP address

1. In the Cloud Console, on the **Navigation menu** (≡), click **Compute Engine > VM instances**.
2. Click **Create**.

3.Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Name	vm-internal
Region	us-central1
Zone	us-central1-c
Machine type	n1-standard-1(1vCPU, 3.75 GB memory)

4.Click **Management, security, disks, networking, sole tenancy**.


5.Click **Networking**.

6.For **Network interfaces**, click the pencil icon to edit.

7.Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Network	Privatenet
Subnetwork	privatenet-us
External IP	None

8. Click **Done**.
9. Click **Create**, and wait for the VM instance to be created.
10. On the **VM instances** page, verify that the **External IP** of **vm-internal** is **None**.

VM instances							SHOW INFO PANEL
Filter VM instances							Columns
<input type="checkbox"/> Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect	
<input type="checkbox"/>  vm-internal	us-central1-c			10.130.0.2 (nic0)	None	SSH	⋮

Create the bastion host

Because **vm-internal** has no external IP address, it can only be reached by other instances on the network or via a managed VPN gateway. This includes SSH access to **vm-internal**, which is grayed out (unavailable) in the Cloud Console. In order to connect via SSH to **vm-internal**, create a bastion host **vm-bastion** on the same VPC network as **vm-internal**.

1. In the Cloud Console, on the **VM instances** page, click **Create Instance**.
2. Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Name	vm-bastion
Region	us-central1

Zone	us-central1-c
Machine type	f1-micro (1vCPU)
Identity and API access > Access scopes	Set access for each API
Compute Engine	Read Write

1. Click **Management, security, disks, networking, sole tenancy**.
2. Click **Networking**.
3. For **Network interfaces**, click the pencil icon to edit.
4. Specify the following, and leave the remaining settings as their defaults:

Property	Value (type value or select option as specified)
Network	privatenet
Subnetwork	privatenet-us
External IP	Ephemeral

5. Click **Done**.

6. Click **Create**, and wait for the VM instance to be created.

SSH to **vm-bastion** and verify access to **vm-internal**

Verify to access **vm-internal** through **vm-bastion**.

1. For **vm-bastion**, click **SSH** to launch a terminal and connect.

<input type="checkbox"/>	Name ^	Zone	Recommendation	In use by	Internal IP	External IP	Connect
<input type="checkbox"/>	 vm-bastion	us-central1-c			10.130.0.3 (nic0)	35.193.84.221	SSH ▼
<input type="checkbox"/>	 vm-internal	us-central1-c			10.130.0.2 (nic0)	None	SSH ▼

2. From the **vm-bastion** SSH terminal, verify external connectivity by running the following command:

```
ping -c 2 www.google.com
```

3. Connect to **vm-internal** by running the following command:

```
gcloud compute sshvm-internal --zone=us-central1-c --internal-ip
```

When asked if you want to continue, enter Y.

4. When prompted for a passphrase, press **ENTER** for no passphrase, then **ENTER** again.

5. Test the external connectivity of **vm-internal** by running the following command:

```
ping -c 2 www.google.com
```

6. This should not work because **vm-internal** has no external IP address!

7. Wait for the ping command to complete.

8. Close the connection to **vm-internal** by running the following command:

```
exit
```

9. Close the SSH terminal of **vm-bastion** by running the following command

```
exit
```

Module 2 Enable private Google access

VM instances that have no external IP addresses can use Private Google Access to reach external IP addresses of Google APIs and services. By default, Private Google Access is disabled on a VPC network.

Create a Cloud Storage bucket

Create a Cloud Storage bucket to test access to Google APIs and services.

1. In the Cloud Console, on the **Navigation menu** (≡), click **Cloud Storage** > **Bucket**.
2. Click **Create bucket**.

Property	Value (type value or select option as specified)
Name	Enter a globally unique name
Default storage class	Multi-Regional

3. Specify the following, and leave the remaining settings as their defaults.
4. Click **Create**.
5. Note the name of your storage bucket for the next subtask. It will be referred to as [my_bucket].

Copy an image file into your bucket

Copy an image from a public Cloud Storage bucket to your own bucket.

1. Run the following command in Cloud Shell, replacing [my_bucket] with your bucket's name:

```
gsutil cp gs://cloud-training/gcpnet/private/access.png gs://[my_bucket]
```

2. In the Cloud Console, click **Refresh Bucket** to verify that the image was copied.

We can click on the name of the image in the Cloud Console to view an example of how Private Google Access is implemented.

Access the image from your VM instances

1. In the Cloud Console, on the **Navigation menu** (☰), click **Compute Engine > VM instances**.
2. For **vm-bastion**, click **SSH** to launch a terminal and connect.
3. Try to copy the image to **vm-bastion** by running the following command, replacing [my_bucket] with your bucket's name:

```
gsutil cp gs://[my_bucket]/*.png .
```

This should work because **vm-bastion** has an external IP address!

4. Connect to **vm-internal** by running the following command:

```
gcloud compute sshvm-internal --zone=us-central1-c --internal-ip
```

5. If prompted, type **Y** to continue.
6. Try to copy the image to **vm-internal** by running the following command, replacing [my_bucket] with your bucket's name:

```
gsutil cp gs://[my_bucket]/*.png .
```

7. To terminate the request after the first attempt, press CTRL+C.
8. Close the SSH terminal.

Enable private Google access

Private Google access is enabled at the subnet level. When it is enabled, instances in the subnet that only have private IP addresses can send traffic to Google APIs and services through the default route (0.0.0.0/0) with a next hop to the default internet gateway.

1. In the Cloud Console, on the **Navigation menu** (☰), click **VPC network > VPC networks**.
2. Click **private net > private net-us** to open the subnet.
3. Click **Edit**.

4. For **Private Google access**, select **On**.
5. Click **Save**.
6. In the Cloud Console, on the **Navigation menu** (≡), click **Compute Engine > VM instances**.
7. For **vm-bastion**, click **SSH** to launch a terminal and connect.
8. Connect to **vm-internal** by running the following command:

```
gcloud compute ssh vm-internal --zone=us-central1-c --internal-ip
```
9. If prompted, type **Y** to continue.
10. Try to copy the image to **vm-internal** by running the following command:

```
gsutil cp gs://[my_bucket]/*.png .
```

This should work as **vm-internal**'s subnet has **Private Google Access** enabled!

11. Close the SSH terminal.

Module 3 Configure a Cloud NAT gateway

Although **vm-internal** can now access certain Google APIs and services without an external IP address, the instance cannot access the internet for updates and patches. You will now configure a Cloud NAT gateway, which allows **vm-internal** to reach the internet.

Try to update the VM instances

1. For **vm-bastion**, click **SSH** to launch a terminal and connect.
2. Try to re-synchronize the package index of **vm-bastion** by running the

Following:

```
sudo apt-get update
```

The output should look like this:

```
...
Reading package lists... Done
```

This should work because **vm-bastion** has an external IP address!

3. Connect to **vm-internal** by running the following command:

4. If prompted, type **Y** to continue.

```
gcloud compute sshvm-internal --zone=us-central1-c --internal-ip
```

5. Try to re-synchronize the package index of **vm-internal** by running the following:

```
sudo apt-get update
```

This should only work for Google Cloud packages because **vm-internal** only has access to Google APIs and services!

6. Press CTRL+C to stop the request.

8. Close the SSH terminal.

Configure a Cloud NAT gateway

Cloud NAT is a regional resource. You can configure it to allow traffic from all ranges of all subnets in a region, from specific subnets in the region only, or from specific primary and secondary CIDR ranges only.

1. In the Cloud Console, on the **Navigation menu** (≡), click **Network services > Cloud NAT**.

2. Click **Get started** to configure a NAT gateway.

3. Specify the following:

Property	Value (type value or select option as specified)
Gateway name	nat-config
VPC network	privatenet
Region	us-central1

4. For **Cloud Router**, select **Create new router**.
5. For **Name**, type **nat-router**.
6. Click **Create**.
7. Click **Create**.
8. Wait for the gateway's **Status** to change to **Running**.

Verify the Cloud NAT gateway

It may take up to 3 minutes for the NAT configuration to propagate to the VM, so wait at least a minute before trying to access the internet again.

1. In the Cloud Console, on the **Navigation menu** (≡), click **Compute Engine > VM instances**.
2. For **vm-bastion**, click **SSH** to launch a terminal and connect.
3. Connect to **vm-internal** by running the following command:

```
gcloud compute sshvm-internal --zone=us-central1-c --internal-ip
```

4. If prompted, type **Y** to continue.
5. Try to re-synchronize the package index of **vm-internal** by running the following:

```
sudo apt-get update
```

The output should look like this:

```
...
```

```
Reading package lists... Done
```

This should work because **vm-internal** is using the NAT gateway!

6. Close the SSH terminal.

Module 4 Configure and view logs with Cloud NAT Logging

Cloud NAT logging allows you to log NAT connections and errors. When Cloud NAT logging is enabled, one log entry can be generated for each of the following scenarios:

- When a network connection using NAT is created.

- When a packet is dropped because no port was available for NAT.

We can opt to log both kinds of events, or just one or the other. Created logs are sent to Cloud Logging.

Enabling logging

If logging is enabled, all collected logs are sent to Stackdriver by default. You can filter these so that only certain logs are sent. We can also specify these values when you create a NAT gateway or by editing one after it has been created. The following directions show how to enable logging for an existing NAT gateway.

1. In the Cloud Console, on the **Navigation menu** (≡), click **Network services > Cloud NAT**.
2. Click on the nat-config gateway and then click **Edit**.
3. Click the Logging, minimum ports, timeout dropdown to open that section.
4. Under Stackdriver logging, select **Translation and errors** and then click **Save**.

NAT logging in Stackdriver

Now that we have set up Cloud NAT logging for the nat-config gateway, let's find out where we can view our logs. You should have left off on the following page with your gateway updated:

Cloud NAT

+

CREATE NAT GATEWAY

DELETE

REFRESH

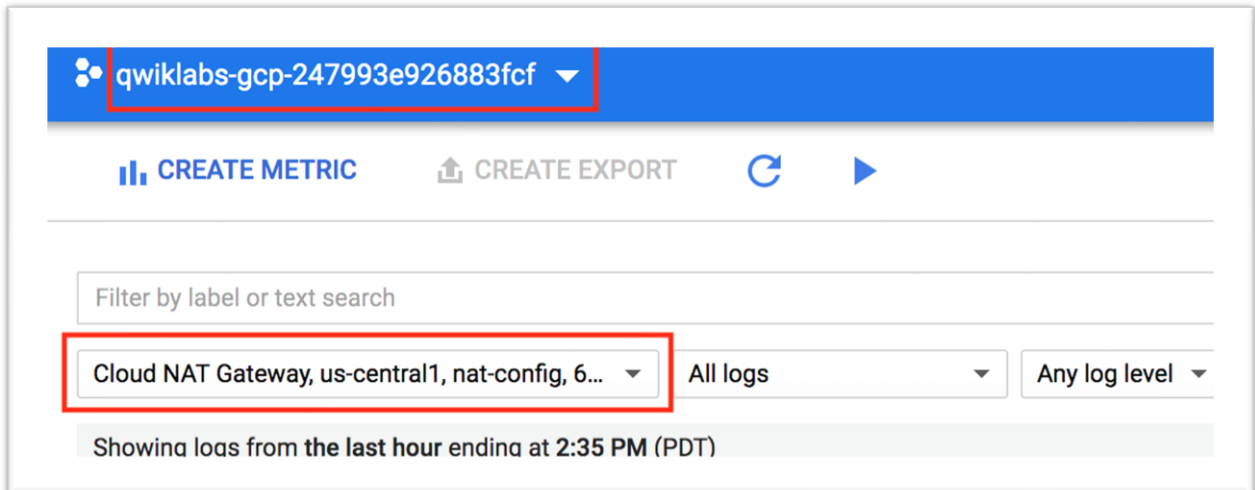
Filter by NAT gateway properties

?

Columns

<div><div></div> Gateway name ^</div>	Region	Cloud Router	Status	High availability
<div><div></div> nat-config</div>	us-central1	<u>nat-router</u>	<div><div></div>Running</div>	Yes <div><div></div></div>

1. Click on nat-config to expose its details. Then click on the **Logs** tab. Then click the link to **Cloud Logging**.
2. This will open a new tab with Stackdriver logging—ensure that the top of your page resembles the following:



We will see that there aren't any logs yet—that's because we just enabled this feature for the gateway. **Keep this tab open** and return to your other Cloud Console tab.

Generating logs

As a reminder, Cloud NAT logs are generated for the following sequences:

- When a network connection using NAT is created.
- When a packet is dropped because no port was available for NAT.

Let's connect the bastion host to the internal VM again to see if any logs are generated.

1. In the Cloud Console, on the **Navigation menu** (☰), click **Compute Engine > VM instances**.
2. For **vm-bastion**, click **SSH** to launch a terminal and connect.
3. Connect to **vm-internal** by running the following command:

```
gcloud compute sshvm-internal --zone=us-central1-c --internal-ip
```

4.If prompted, type Y to continue.

5.Try to re-synchronize the package index of **vm-internal** by running the following:

```
sudo apt-get update
```

The output should look like this:

```
...  
Reading package lists... Done
```

6. Close the SSH terminal.

Let's see if opening up this connection revealed anything new in our logs.

Viewing logs

1.Return to your NAT Logging tab and click the **Load newer logs** button.

2.We should see two new logs that were generated after connecting the bastion host to the internal VM.

As we see, the logs give us details on the VPC network we connected to and the connection method we used.

5.2 SCREENSHOTS

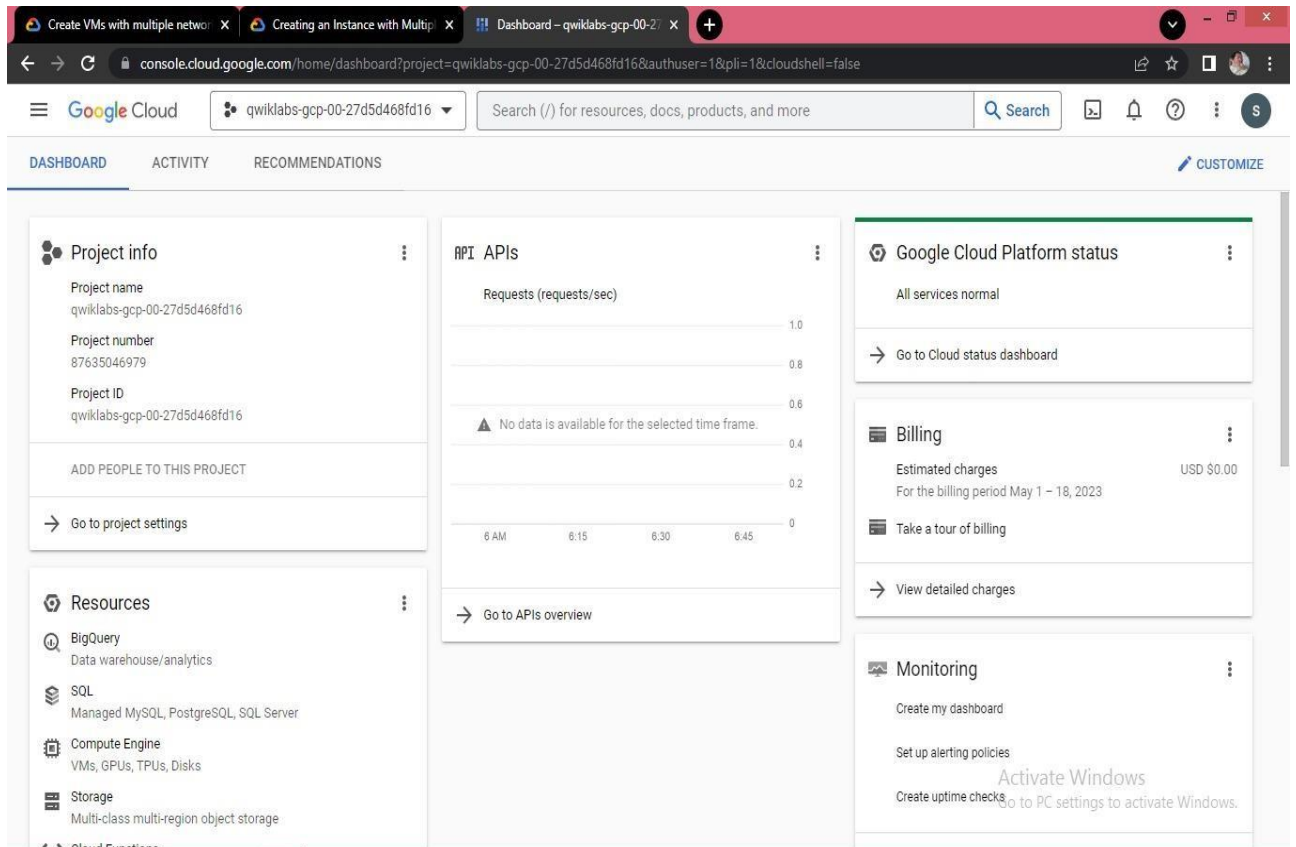


Fig 5.2.1 OPEN GOOGLE CONSOLE

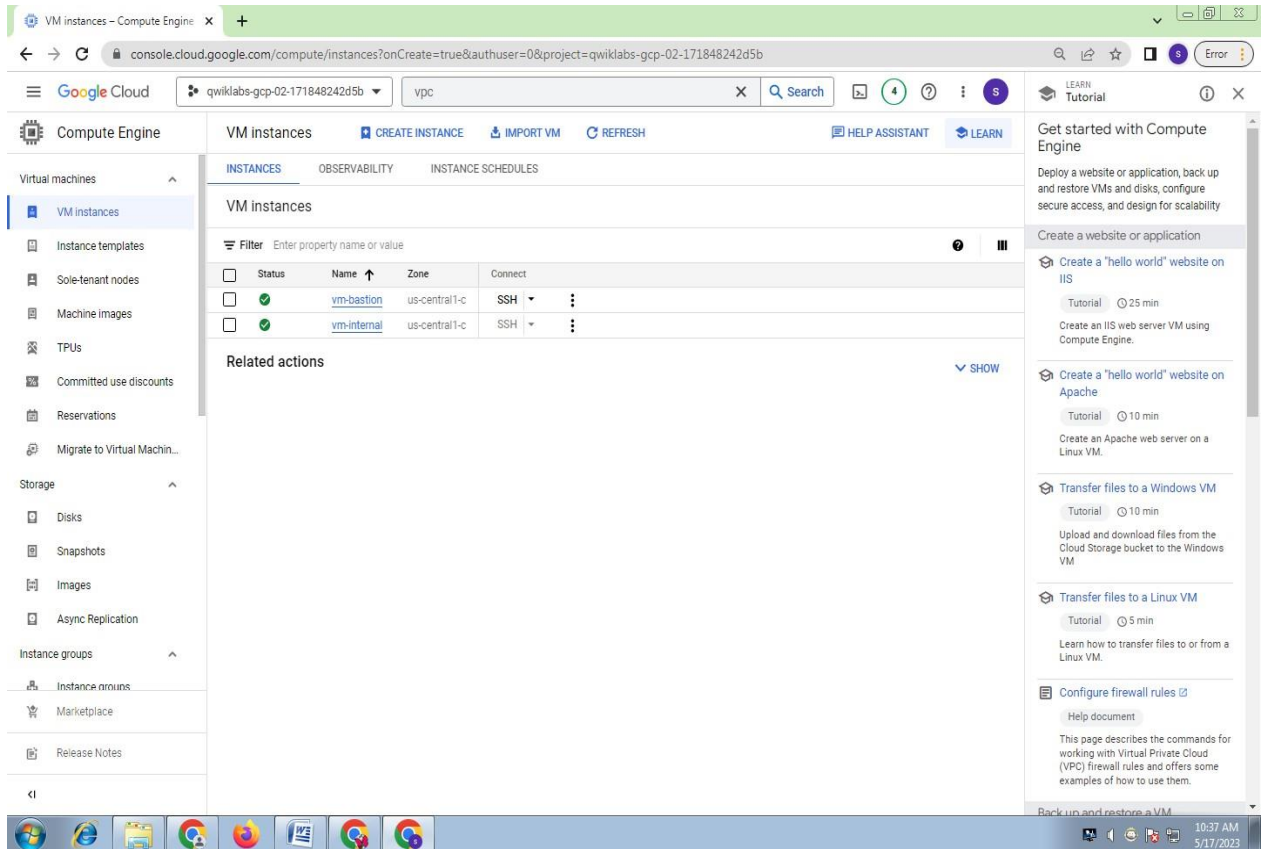


Fig 5.2.2 CREATING THE VM INSTANCE

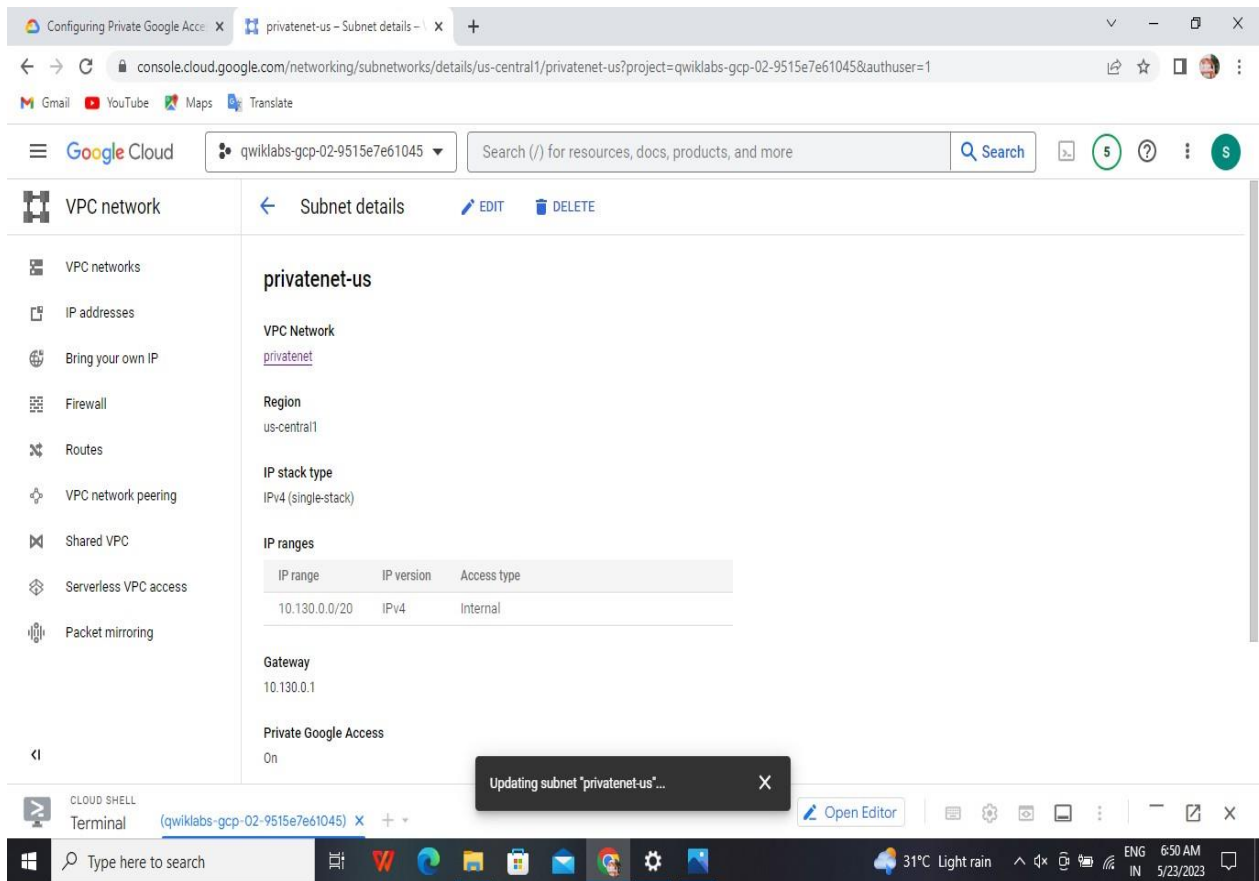


Fig 5.2.3 ENABLING PRIVATE GOOGLE ACCESS

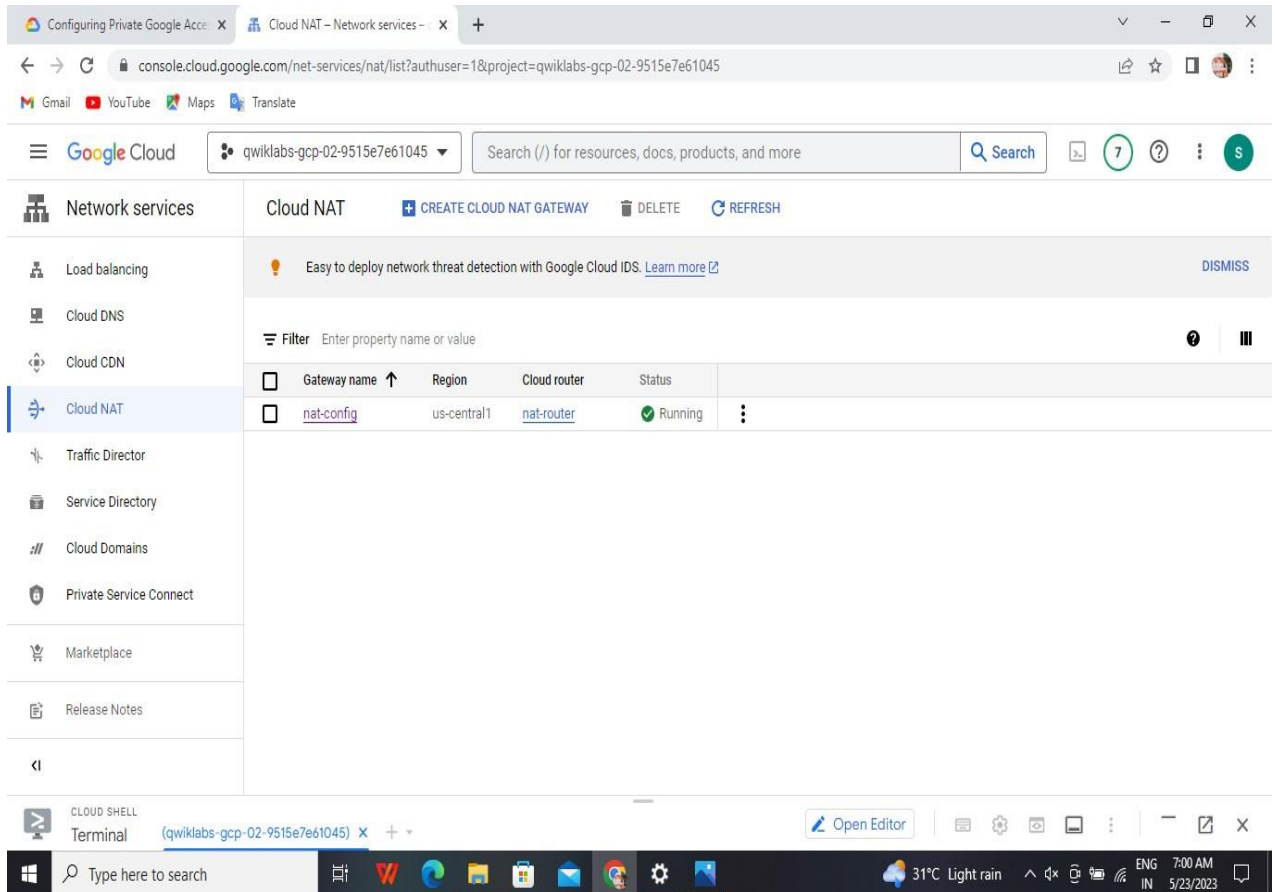


Fig 5.2.4 CONFIGURING A CLOUD NAT GATEWAY

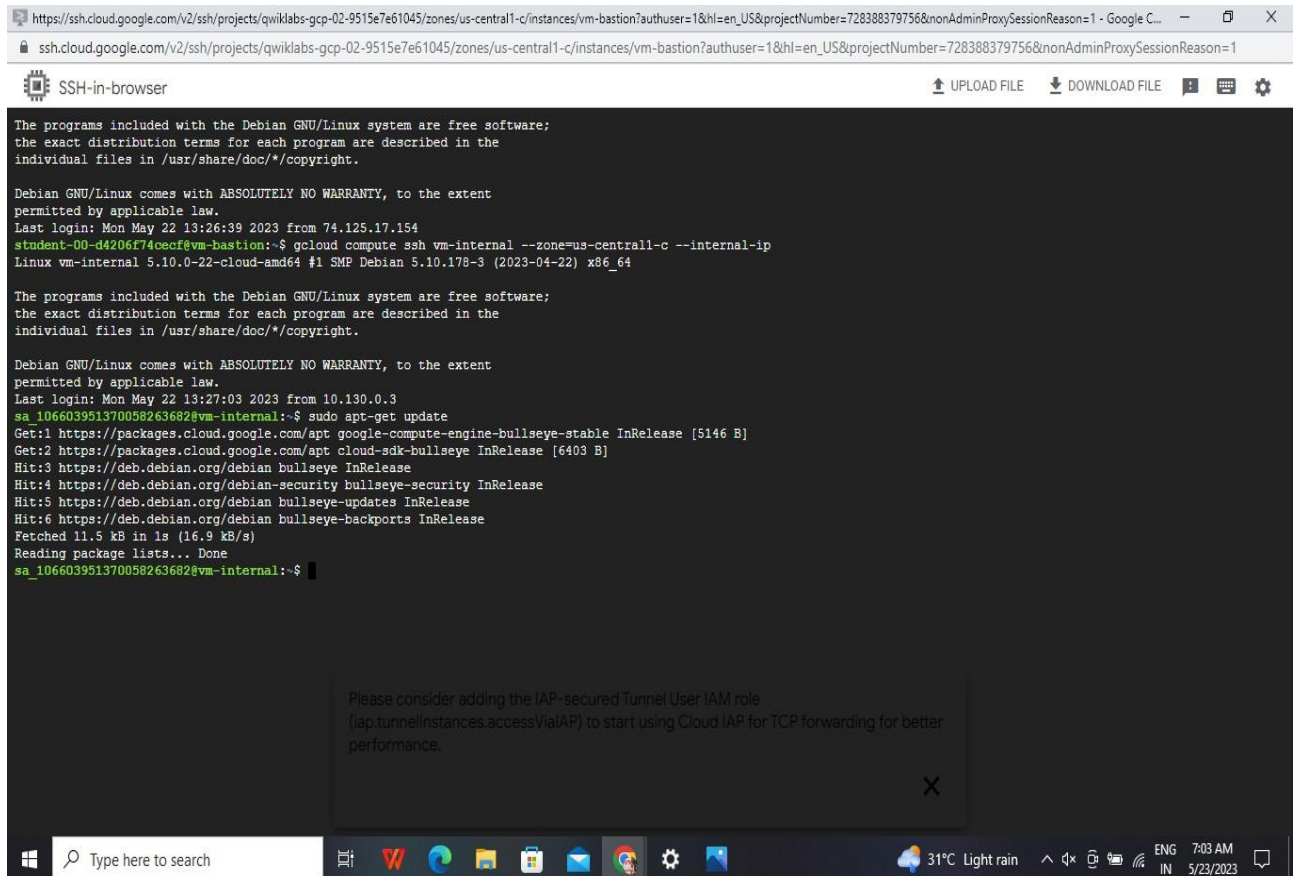


Fig 5.2.5 CONFIGURE AND VIEW LOGS WITH CLOUD NAT LOGGING

CHAPTER 6

SYSTEM TESTING

6.1 TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include various types of testing, but are not limited to the process of executing a program or application with the intent of finding software bugs(errors or other defects).

6.2 TYPES OF TESTING

- 1.Connectivity Test
- 2.Private Google Access Test
- 3.Traffic Inspection Test
- 4.Load and Scalability Test
- 5.Failover Test
- 6.Security Test
- 7.Logging and Monitoring Test

6.2.1 Connectivity Test

Verify that your instances in the private subnet can connect to the internet through Cloud NAT. You can perform a simple test by pinging an external IP address or running a command to access a public website from your instance.

6.2.2 Private Google Access Test

Check if your instances in the private subnet can reach Google APIs and services without requiring a public IP address. Test the connectivity by accessing various Google services like Google Cloud Storage or Big Query from our instance.

6.2.3 Traffic Inspection Test

Monitor and inspect the network traffic to ensure that traffic leaving the instances in the private subnet is properly translated by Cloud NAT. You can use packet capture tools or network monitoring solutions to examine the traffic and verify the translation.

6.2.4 Load and Scalability Test

Simulate a high load scenario by generating increased traffic from instances in the private subnet that use Cloud NAT. Test if the Cloud NAT setup can handle the increased workload and scale as per your requirements without any performance degradation.

6.2.5 Failover Test

Validate the resiliency and failover capabilities of Cloud NAT. Intentionally cause a failure of a Cloud NAT gateway or simulate a network disruption to ensure that failover mechanisms are functioning as expected and traffic is properly redirected.

6.2.6 Security Test

Assess the security of your Cloud NAT setup by conducting vulnerability scans or penetration tests. Ensure that the configuration is resilient against common security threats and no unauthorized access or exposure of private

instances or resources occurs.

6.2.7 Logging and Monitoring Test

Enable logging and monitoring for Cloud NAT and review the logs and metrics to ensure that the system is operating correctly. Check for any errors, warnings, or anomalies in the logs and validate that the monitoring metrics are within acceptable ranges.

These tests help verify the proper configuration and functionality of Private Google Access and Cloud NAT in GCP. It is essential to perform thorough testing to ensure a reliable and secure network setup.

6.3 BLOCK DIAGRAM

Cloud NAT provides internet access to private instances

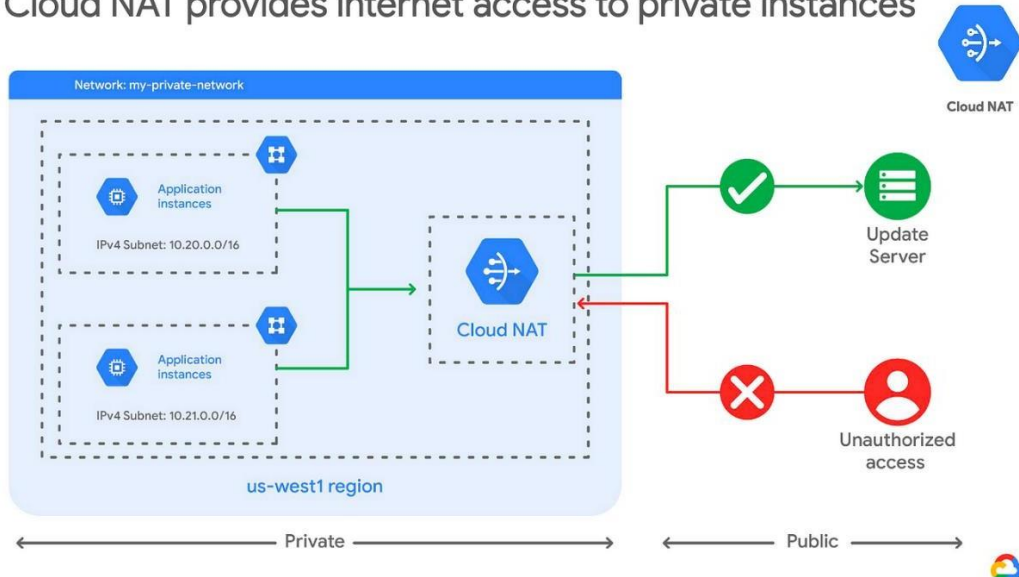


Fig 6.3.1 INTERNET ACCESS OF CLOUD NAT IN GCP

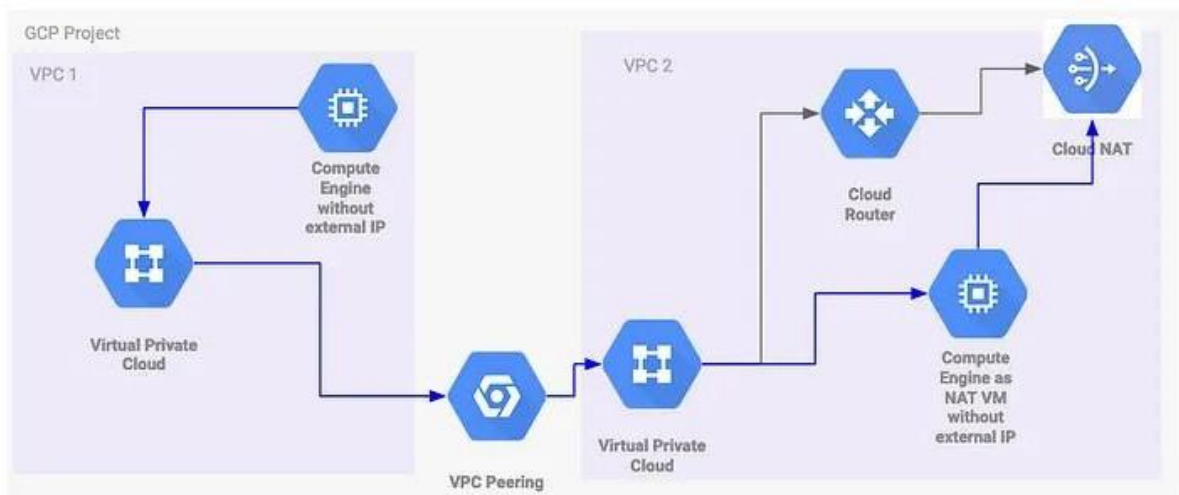


Fig 6.3.2 CLOUD NAT WITH VPC PEERING

CHAPTER 7

CONCLUSION

In conclusion, configuring private Google Access and Cloud NAT in Google Cloud Platform (GCP) offers several benefits for our cloud environment. Private Google Access allows our virtual machine instances to access Google APIs and services using internal IP addresses, without exposing them to the public internet. This enhances security by minimizing exposure to potential threats. We created `vm-internal`, an instance with no external IP address, and `vm-bastion`, a bastion host to securely connect to `vm-internal`. Then we enabled Private Google Access, configured a NAT gateway, and verified that `vm-internal` can access Google APIs and services and other public IP addresses.

Finally, we did how to configure NAT Logging and how to generate and view those logs in Stackdriver. VM instances without external IP addresses are isolated from external networks. Using Cloud NAT, these instances can access the internet for updates and patches, and in some cases, for bootstrapping. As a managed service, Cloud NAT provides high availability without user management and intervention. Overall, by leveraging the capabilities of Private Google Access and Cloud NAT, you can enhance the security and control of your GCP environment while allowing your instances to access necessary resources over the internet in a controlled manner.

7.1 FUTURE ENHANCEMENT

Cloud NAT (Network Address Translation) is a service provided by cloud providers that enables instances within a private network to communicate with the internet using a shared public IP address. While Cloud NAT already offers valuable functionality, there are potential future enhancements that could further improve its capabilities. Cloud NAT could be enhanced to handle even larger-scale deployments. This could involve optimizations in terms of performance, throughput, and the number of concurrent connections it can support. Improved scalability would enable organizations with rapidly growing networks to efficiently manage their traffic and accommodate increasing demands. Currently, Cloud NAT assigns a single public IP address to a NAT gateway.

Future enhancements could provide more flexible IP management options. For example, allowing dynamic allocation and deallocation of public IP addresses based on traffic demand or allowing the use of multiple public IP addresses for load balancing and redundancy purposes. Cloud NAT primarily supports outbound connectivity using TCP and UDP protocols. Future enhancements might include support for additional protocols, such as ICMP (Internet Control Message Protocol) for better troubleshooting and diagnostics, or other specialized protocols that are commonly used in specific industries or applications. As IPv6 adoption continues to grow, future enhancements to Cloud NAT might include support for translating IPv6 addresses to IPv4 addresses, enabling seamless communication between IPv6-only networks and IPv4 networks. It's important to note that these are potential future enhancements and may vary depending on the cloud provider and their roadmap for Cloud NAT. Organizations should stay updated with their cloud provider's announcements and documentation to learn about the latest features and improvements.

REFERENCES

- [1] "Enhancing the Security of a Private Network by Using A Multi-level Hierarchical NAT Scheme" by Shie-Yuan Wang, Yu-Hsun Yuan in 2022.
- [2] "Telerobotic Operations Using Network Address Translation (NAT) Traversal" by Xinyu Liu, Chun Ho So, Burak Kizilkaya in 2022.
- [3] "NAT experimental design and result analysis based on simulation technology" by Cao Huamei in 2022.
- [4] "Hyper NAT: Scaling Up Network Address Translation with SmartNICs , for Clouds" by Shaoke Fang, Qing song Liu, Wenfei Wu in 2021
- [5] "CASS-NAT: CTC Alignment-Based Single Step Non-Autoregressive Transformer for Speech Recognition" by Ruchao Fan, Wei Chu, Peng Chang, Jing Xiao in 2021.
- [6] "Identifying NAT Devices to Detect Shadow IT: A Machine Learning Approach" by Reem Nassar, Imad Elhajj, Ayman Kayssi, Samer Salam in 2021.
- [7] "Probe Delay Based Adaptive Port Scanning for IoT Devices with Private IP Address Behind NAT" by Fengxiao Tang, Yuichi Kawamoto, Nei Kato, Kazuto Yano, Yoshinori Suzuki in 2020.
- [8] "The P2P Solution Research and Design Based on NAT Traversing Technology" by Fangliang Huang, Tongping Shen, Sheng Hu in 2019.
- [9] "Blockchain-Based NAT Management for Smart Mobility" by Youchan Jung, Marnel Peradilla, Ronnel Agulto in 2018.
- [10] "NAT-Aware Peer Grouping and Chunk Scheduling for Mesh-Pull P2P Live Streaming Systems" by Wen -Kang Jia, Gen- Hen Liu, Yaw-Chung Chen in 2015.