

# Implementación de Seguridad Zero Trust en Google Cloud: Identity-Aware Proxy (IAP)

**Autor:** Christhian Rodriguez

**Fecha:** Noviembre 2025

**Lab:** "Protege máquinas virtuales con Chrome Enterprise Premium" (GSP1036)

## Resumen Ejecutivo

En este laboratorio implementé una arquitectura de seguridad **Zero Trust** utilizando **Google Cloud Identity-Aware Proxy (IAP)**. El objetivo principal fue eliminar la necesidad de direcciones IP públicas y hosts bastión tradicionales (Jumpboxes) expuestos a internet, permitiendo la administración remota segura de instancias Windows y Linux mediante túneles TCP encriptados y autenticación IAM.

## 1. Arquitectura y Conceptos Clave

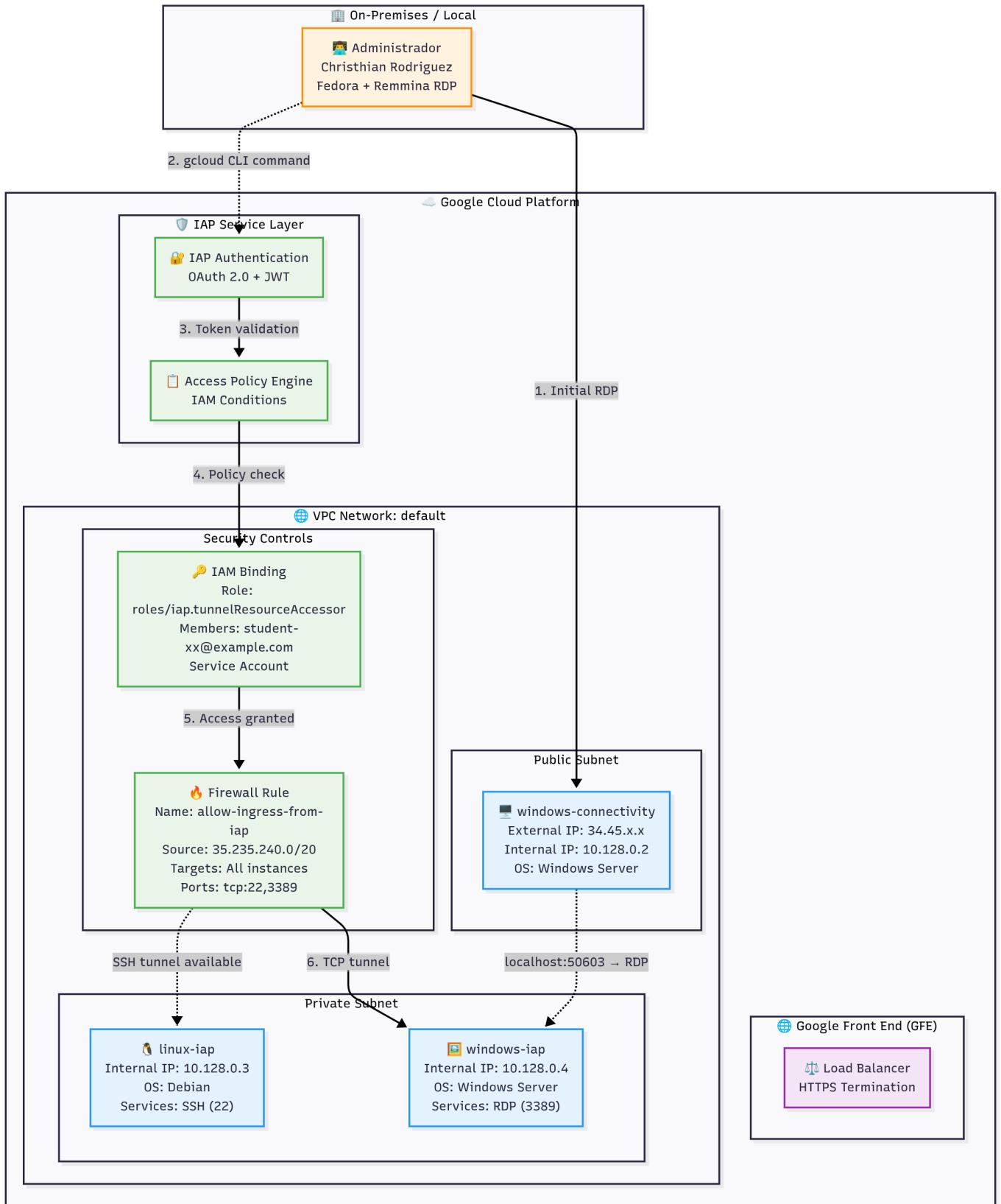
El desafío principal de la seguridad en la nube es reducir la superficie de ataque.

- **El Problema:** Exponer puertos de administración (SSH: 22, RDP: 3389) a internet (`0.0.0.0/0`) invita a ataques de fuerza bruta.
- **La Solución (IAP):** Google actúa como un "proxy" intermedio. Verifica mi identidad (Gmail/Cloud Identity) y si tengo permisos, reenvía el tráfico a través de su red interna hacia la VM. La VM nunca toca internet público.

## Esquema visual

Flujo de Conexión:

Mi Laptop (Fedora) -> Túnel IAP (HTTPS) -> Red de Google -> VM Privada



## 2. Implementación Paso a Paso

### Paso 1: Creación de Infraestructura Aislada

Se aprovisionaron tres máquinas virtuales. El punto crítico aquí fue la configuración de red de las máquinas objetivo (`linux-iap` y `windows-iap`).

- **Configuración:** En la sección de Networking, se estableció `External IPv4 address: None`.

- **Resultado:** Estas máquinas son invisibles desde internet.

## Paso 2: Intento Fallido (Validación de Aislamiento)

Para confirmar que la seguridad funciona, intenté conectarme vía SSH/RDP tradicional. El intento falló con un error de conexión, confirmando que no hay "puerta trasera" abierta.

1. Después de crear las instancias, probarás el acceso a `linux-iap` y `windows-iap` para asegurarte de que no puedes acceder a las VMs sin la IP externa.

2. Para `linux-iap`, haz clic en el botón SSH para acceder a la máquina y asegúrate de recibir un mensaje similar al siguiente:

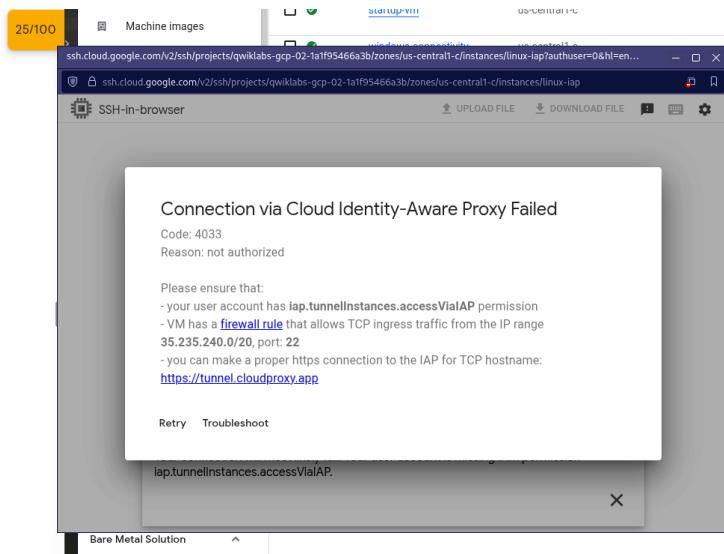


**Nota:** Es posible que el botón SSH siga apareciendo como disponible para hacer clic en la página de la lista de VMs, a pesar de que la IPv4 externa esté configurada como Ninguna. Para confirmar que la instancia no tiene una IP externa, haz clic en su nombre y, luego, coloca el cursor sobre el botón SSH en la página de detalles, que muestra el mensaje: *Esta instancia no tiene una dirección IP externa*.

4. Para `windows-iap`, haz clic en el botón RDP y asegúrate de recibir un mensaje similar al siguiente:



Los siguientes pasos para configurar y usar IAP te permitirán conectar a las instancias que no tienen IPs externas.



## Paso 3: Configuración del Firewall (Pregunta de Examen )

Para que IAP funcione, la red VPC debe confiar en el tráfico proveniente de los balanceadores de carga de Google.

- **Nombre:** `allow-ingress-from-iap`
- **Rango de Origen (Source):** `35.235.240.0/20` (Este es el rango reservado de Google IAP).
- **Protocolos:** TCP 22, 3389.
- **Targets:** Todas las instancias de la red.

The screenshot shows the Google Cloud Network Security Firewall policies section. A new rule is being created with the name "allow-ingress-from-iap". The rule is set to "On" for logs. It is configured for the "default" network with a priority of 1000. The direction of traffic is "Ingress". The action on match is "Allow". Targets are specified as "Specified target tags". The source filter is set to "IPv4 ranges" with the value "35.235.240.0/20". The destination filter is set to "None". Under "Protocols and ports", "TCP" is selected with port 22,3389.

Muestra la configuración correcta de la regla de firewall con el rango 35.235.240.0/20.

## Paso 4: Gestión de Identidad (IAM)

IAP es "Identity-Aware" (Consciente de la identidad). No basta con abrir el puerto; hay que tener permiso.

- Rol:** IAP-Secured Tunnel User (`roles/iap.tunnelResourceAccessor`).

- Principales:**

1. Mi usuario (`student-xx...`).
2. La Service Account de la VM bastión (para permitir saltos entre máquinas).

Captura recomendada: Inserta aquí `image_77b63a.png`.

Muestra el panel derecho de IAP donde agregaste los correos y el rol específico.

## 3. Creación del Túnel ("La Magia Negra")

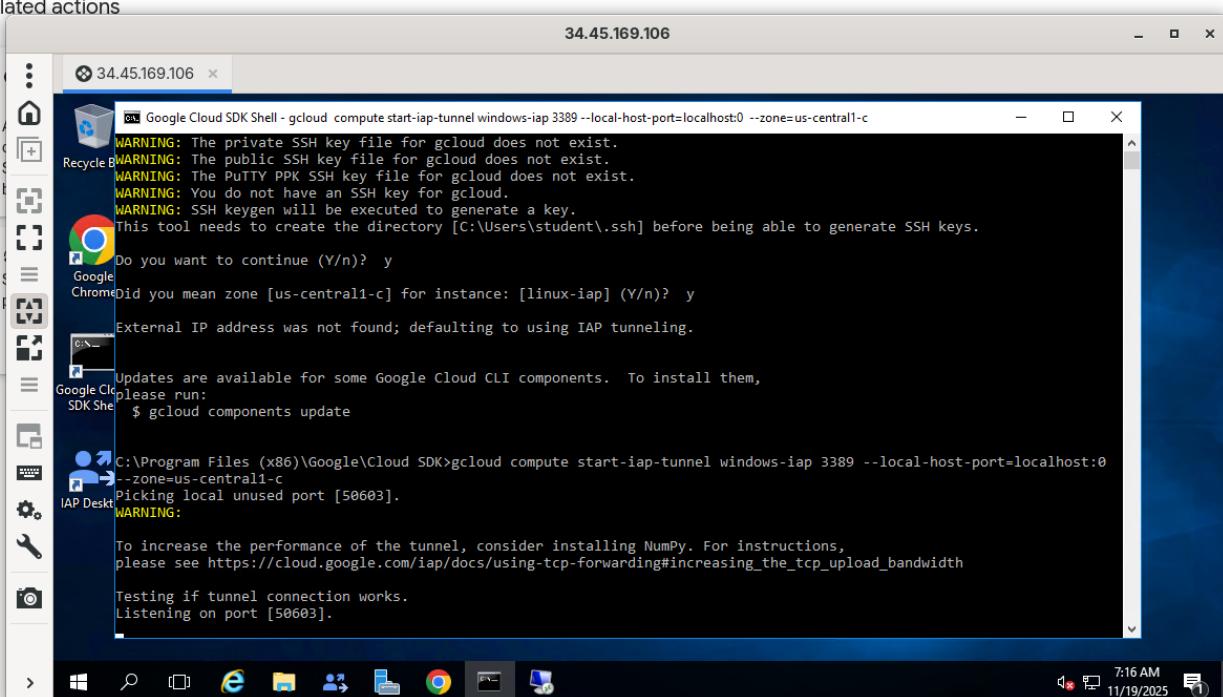
Aquí es donde la teoría se vuelve práctica. Desde una sesión RDP en el bastión ( windows-connectivity ), utilicé el SDK de Google Cloud para abrir un túnel manual hacia la máquina privada.

### Comando ejecutado:

```
gcloud compute start-iap-tunnel windows-iap 3389 --local-host-port=localhost:0  
--zone=us-central1-c
```

### Explicación técnica:

Este comando abre un puerto aleatorio en la máquina local (ej. 50603) y lo conecta mágicamente con el puerto 3389 de la VM privada a través de IAP.



```
WARNING: The private SSH key file for gcloud does not exist.  
WARNING: The public SSH key file for gcloud does not exist.  
WARNING: The PuTTY PPK SSH key file for gcloud does not exist.  
WARNING: You do not have an SSH key for gcloud.  
WARNING: SSH keygen will be executed to generate a key.  
This tool needs to create the directory [C:\Users\student\.ssh] before being able to generate SSH keys.  
Do you want to continue (Y/n)? y  
Did you mean zone [us-central1-c] for instance: [linux-iap] (Y/n)? y  
External IP address was not found; defaulting to using IAP tunneling.  
Updates are available for some Google Cloud CLI components. To install them,  
please run:  
$ gcloud components update  
C:\Program Files (x86)\Google\Cloud SDK>gcloud compute start-iap-tunnel windows-iap 3389 --local-host-port=localhost:0  
--zone=us-central1-c  
Picking local unused port [50603].  
WARNING:  
To increase the performance of the tunnel, consider installing NumPy. For instructions,  
please see https://cloud.google.com/iap/docs/using-tcp-forwarding#increasing_the_tcp_upload_bandwidth  
Testing if tunnel connection works.  
Listening on port [50603].
```

Muestra la terminal negra (SDK Shell) ejecutando el comando y diciendo "Listening on port [50603]".

## 4. Validación Final ("Inception")

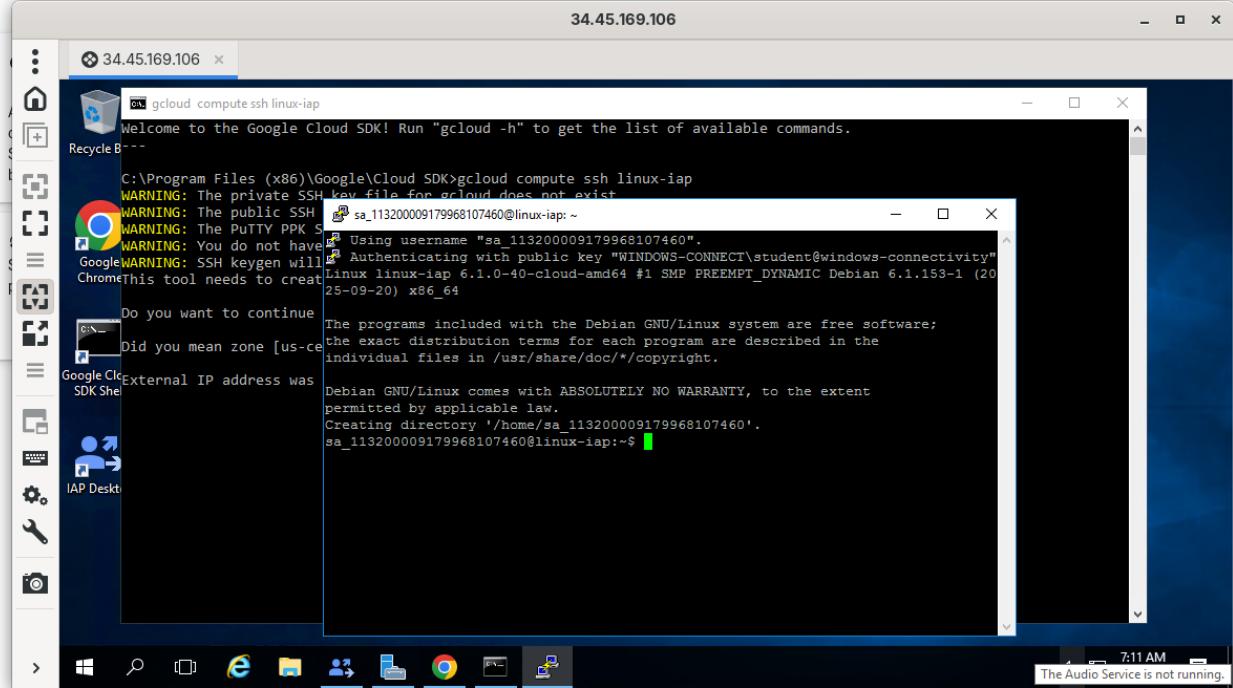
El resultado final es una conexión anidada que demuestra el flujo completo:

- 1. Host:** Fedora Linux (usando Remmina).
- 2. Salto 1:** Conexión RDP a windows-connectivity (IP Pública).

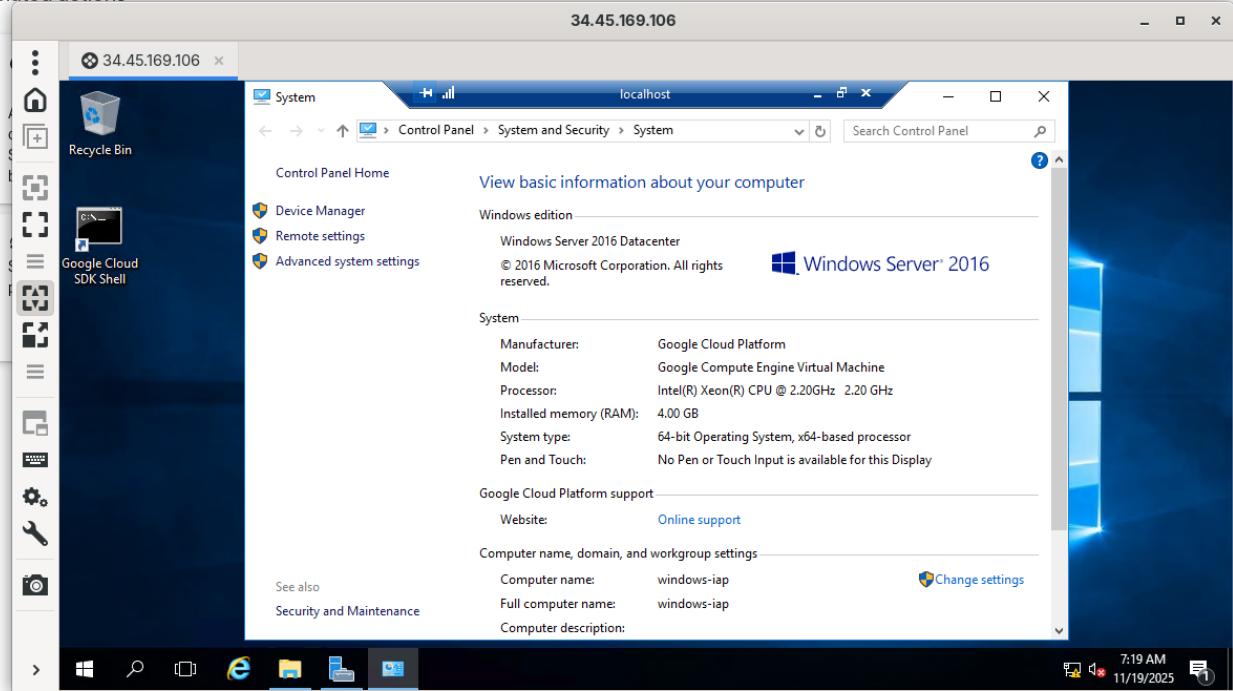
3. Salto 2: Conexión RDP a `localhost:50603` (dentro del Salto 1).

4. Destino: Escritorio de `windows-iap` (Sin IP Pública).

Related actions



Related actions



Esta es la "Money Shot". Muestra el escritorio de Fedora, conteniendo el escritorio de Windows Bastion, conteniendo el escritorio de Windows Privado.

## Cheat Sheet de Comandos Útiles

Durante el lab utilicé estos comandos de `gcloud` para inspección y troubleshooting:

```
# Ver lista de VMs e IPs (Internas/Externas)
gcloud compute instances list

# Crear el túnel IAP (SSH o RDP)
gcloud compute start-iap-tunnel [INSTANCE_NAME] [PORT] --local-host-
port=localhost:0 --zone=[ZONE]

# Consultar el Metadata Server (Desde dentro de una VM) - ¡Tip de Examen!
Invoke-RestMethod -Headers @{"Metadata-Flavor"="Google"} -Uri "
[http://metadata.google.internal/computeMetadata/v1/instance/]
(http://metadata.google.internal/computeMetadata/v1/instance/)"
```

## Conclusiones

---

- **Seguridad:** He aprendido a administrar servidores sin exponerlos a internet, bloqueando escaneos de puertos y ataques externos.
- **Eficiencia:** No fue necesario configurar una VPN Site-to-Site compleja.
- **Interoperabilidad:** Logré gestionar entornos Windows desde una estación de trabajo Linux (Fedora) usando herramientas estándar como RDP y SDK.