INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
Special Issue, March-April 2023, Vol 3 (02), 112-116.
International conference on Innovation towards Sustainable Development Goals-ICISDG'23 ISSN No: 2583 0368

# Securing Access to Applications with IAP and Compute Engine

**Deebika R[1], Gobi B[2], Logeshwaran G[3], Sinthana G[4], Shaheen M[5]**
*[1]Assistant professor, Department of Computer Science and Engineering, KSR Institute For Engineering and Technology Namakkal, Tamil Nadu, India.*
*[2,3,4,5]UG Students, Department of Computer Science and Engineering, KSR Institute For Engineering and Technology Namakkal, Tamil Nadu, India.*

**Article Type:** Research

OPENACCESS
**Article Citation:**
Deebika R[1], Gobi B[2], Logeshwaran G[3], Sinthana G[4], Shaheen M[5], "Securing Access to Applications with IAP and Compute Engine", International Journal of Recent Trends In Multidisciplinary Research, March-April 2023, Vol 3(02), 112-116.

Check for updates

https://www.doi.org/10.59256/ijrtmr.20230402c24



**Abstract:** Cloud Identity- Aware Proxy (IAP) is a security service handed by Google Cloud Platform that allows directors to authenticate and authorize access to web operations and VMs running on Google Cloud. This service provides a fresh subcaste of security to operations and VMs by vindicating the stoner's identity and determining if they've authorization to pierce the resource. Setting up Cloud IAP for Compute Engine involves configuring access programs, OAuth2.0 customer IDs, and firewall rules. Once configured, IAP allows only authorized druggies to pierce the operation, grounded on their Google identity and class in specific Google groups. In this process, directors must produce an access policy, which determines which druggies or groups can pierce the operation. They must also produce an OAuth2.0 customer ID, which is used for authentication with IAP. Eventually, the director must modernize the Compute Engine case firewall rules to allow business from IAP. The benefits of setting up Cloud IAP for Compute Engine include enhanced security for the operation and the capability to manage access to coffers centrally. It also eliminates the need to manage access control within the operation and simplifies the process of granting or repealing access for druggies. While setting up Cloud IAP for Compute Engine can be a complex process, following the recommended way ensures that the operation is secure and only accessible to authorized druggies.

**Key Words:** Cloud IAP, Load Balancer, Firewall Rules.

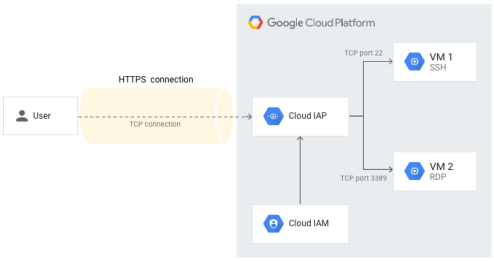
## 1. Introduction

In this design, a company needs to give secure access to their commercial operations running on Compute Engine cases, while also icing that access is confined to authorized druggies and bias. The design would involve setting up IAP to control access to the Compute Engine cases, with access control programs that apply strong authentication and authorization conditions. The design could include tasks similar as configuring IAP to work with the company's being identity and access operation system, setting up VPN access to the Compute Engine cases, and enforcing monitoring and waking to insure that any implicit security pitfalls are snappily detected and eased. The end result would be a secure, scalable, and dependable pall structure that provides authorized druggies with easy and secure access to the company's operations and services.

## 2.Concepts

### 2.1 CLOUD IAP

Cloud Identity-Aware Proxy (IAP) is a service provided by Google Cloud that allows you to control access to your web applications and VMs hosted on Google Cloud Platform (GCP) by verifying user identity and context. When a user attempts to access a protected resource, IAP intercepts the request and verifies the user's identity based on their Google account or a third-party identity provider. IAP then checks the user's access level and permission against the Access Control Lists (ACLs) that you configure for the application, and if the user is authorized, IAP forwards the request to the application.




**2.2 Cloud IAM**

Cloud IAM (Identity and Access Management) is a Google Cloud Platform service that provides centralized control and visibility over the permissions of resources within a Google Cloud project. It allows administrators to manage permissions for users, groups, and service accounts, as well as set policies that determine who can perform actions on specific resources. Cloud IAM supports fine-grained access control, allowing administrators to specify permissions at the individual resource level. Some key features of Cloud IAM include role-based access control, which allows administrators to assign roles to users and service accounts, and the ability to grant permissions to users outside of the Google Cloud project, such as customers or partners. It also integrates with other Google Cloud Platform services such as Cloud Storage, Compute Engine, and BigQuery.

**2.3 Compute Engine**

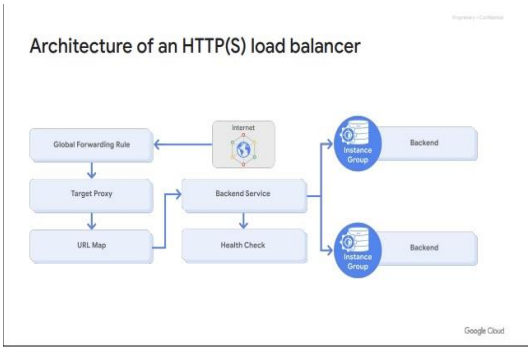
Google Cloud Compute Engine is a part of the Google Cloud Platform that enables users to launch and manage virtual machines (VMs) in the cloud. Compute Engine provides infrastructure as a service (IaaS) and allows users to choose from a range of virtual machine types, from small micro-instances to large machine types with many CPU cores and lots of memory. Users can select the operating system for their VMs, configure the network, attach persistent disks, and manage their VMs through the Compute Engine API or the web-based console. Compute Engine also offers features like autoscaling, load balancing, and preemptible VMs, allowing users to build scalable and fault-tolerant applications in the cloud.

**2.4 Firewall Rules**

Firewall rules are defined at the project level in Google Cloud and apply to all instances within that project. They can be used to limit access to certain ports or protocols, block traffic from specific IP ranges, or allow traffic only from specific sources, among other use cases. Google Cloud provides default firewall rules for each project, which allow incoming traffic on certain ports and protocols by default. These rules can be modified or deleted as needed, and new custom firewall rules can be created to meet specific requirements.

**2.5 Load Balancer**

In Google Cloud, a load balancer is a managed service that can distribute incoming traffic among multiple virtual machine instances or other backend resources to improve the availability and scalability of your applications. Load balancers can be configured with health checks to monitor the status of backend resources and ensure that traffic is only sent to healthy instances. They can also perform SSL offloading, which allows SSL/TLS encryption and decryption to be handled by the load balancer instead of the backend instances, which can help reduce the workload on those instances. Load balancers can also be configured with features like session affinity and URL mapping to help optimize traffic distribution. HTTP(S) Load Balancing is one of type of loads balancing and it is a global load balancer that can distribute traffic across multiple regions and backend services, and supports HTTP and HTTPS traffic.




## 3. Proposed System

Google has recently launched a new feature called "IAP for TCP forwarding," which simplifies the process of setting up IAP for Compute Engine. With this feature, you can protect Compute Engine instances that are listening on arbitrary TCP ports, such as HTTP or HTTPS.

To set up IAP for TCP forwarding, you can follow these steps:
1. Enable IAP: First, you need to enable IAP for your Google Cloud project.
2. Create a TCP forwarding target: Next, you need to create a TCP forwarding target for the Compute Engine instance that you want to protect.
3. Configure IAP for TCP forwarding: Once you have created the TCP forwarding target, you can configure IAP for TCP forwarding by creating a Cloud IAP resource and adding a policy that specifies which TCP forwarding targets you want to protect.
4. Set up the VM: Finally, you need to set up the VM that you want to protect by installing the IAP daemon and configuring the VM's firewall to allow traffic from IAP.

## 4. Implementation

The following are fulfilled to achieve the ideal of this paper.There are five phases in the following method. They are Create a Compute Engine Instance, create a Managed Instance Groups, get a domain name and certificate, create a Load Balancer, Set up IAP.

In google cloud console under Compute Engine we are create the instance template. The template is with a N1 series machine, f1-micro with vCPU, setting the compute engine access as Read-only, allow the HTTP traffic. The template also includes a startup script that installs git, virtualenv, and dependencies for a Python application. It then clones a Google Cloud Platform repository, creates a virtual environment, and installs dependencies for the application. Finally, it runs the application using Gunicorn on port 80.

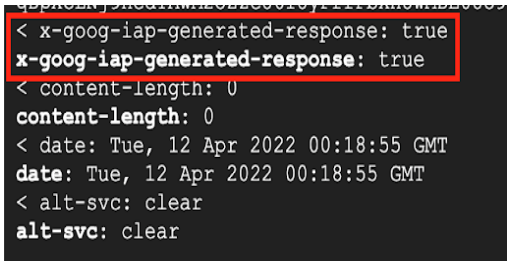
After creating the instance template our next phase is to create a Managed Instance Group (MIG) in Google Cloud Console. The Managed Instance Group is configured with a name of "my-managed-instance-group", an instance template that was created in a previous step, and is set to be located across multiple zones. The user can also set the number of instances to 3 by turning off autoscaling mode. Finally, the Managed Instance Group is created.

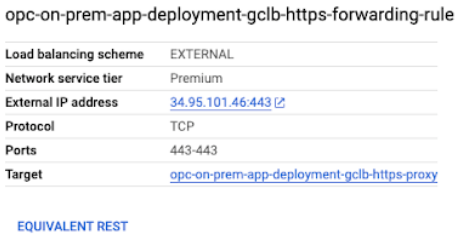
Next phase is creating an SSL certificate and private key in Google Cloud. First, create a private key and certificate signing request (CSR) using OpenSSL. Then, the CSR can be signed by a trusted certificate authority (CA) to create a publicly trusted certificate, or by a self-managed CA to create an internally trusted certificate. If neither option is available, a self-signed certificate can be created for testing purposes. Next, to create a Google Cloud SSL certificate resource, use the gcloud compute ssl-certificates create command with the appropriate parameters, including the private key and certificate files. Finally, a global SSL certificate can be created with the --global flag.

In fourth phase we create a load balancer, In the Cloud Console, go to Network Services and select the project. Click on HTTP(S) Load Balancing and select "From Internet to my VMs or serverless services". Enter a name for the load balancer and create a backend service with a specific name ("my-backend-service"). Set the instance group and port to 80 and uncheck "ENABLE CLOUD CDN". Create a health check and save it. In frontend services set the protocol to HTTPS and create a new static IP address. Reserve the IP address and click "Done". Finally, click "Create" and note the external IP address under Details > Frontend, click "Create" to create the load balancer.
To authenticate requests from IAP, the VMs in the Managed Instance Group need to be restarted. To do this, go to the Compute Engine > Instance groups page in the Cloud Console, select the relevant instance group, click Restart/Replace VMs, and set the values for the operation, maximum unavailable instances, and minimum wait time. Finally, click Restart VMs to restart the VMs.

In fifth phase by block access to underlying VMs and only allow access through IAP, you need to delete the default-allow-internal firewall rule and create a new firewall rule. You can do this by going to the Cloud Console VPC Network > Firewall rules and selecting the checkbox next to the default-allow-internal rule, then clicking Delete and confirming the deletion. Next, you need to create a new firewall rule called allow-iap-traffic with the target set to all instances in the network and source IPv4 ranges of 130.211.0.0/22 and 35.191.0.0/16. Finally, you need to specify the protocols and ports as TCP on port 80 and 78, and then click Create to finish creating the new firewall rule.

Once the firewall is created then we enable Identity-Aware Proxy (IAP) in Google Cloud Console. It involves enabling the API, configuring the OAuth consent screen, and toggling on IAP for a specific project. It warns users not to enter any confidential information on the OAuth consent screen as it may be publicly visible. Once the necessary steps are completed, users can turn on IAP for their backend service.

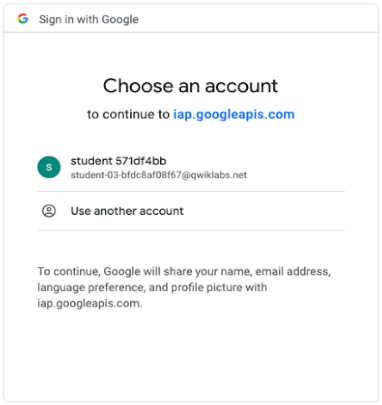



After configuring iap and Oauth add principals to the IAP access list for their project. They need to go to the IAM & admin > Identity-Aware Proxy page, select the relevant service, and click Add Principal. Then, they need to enter the email address of the user to grant access to, select the "IAP-secured Web App User" role, and save. Finally, they need to confirm access by configuring IAM.To confirm that IAP is configured and protecting traffic for a GCE instance, you need to go to the Network Services > Load balancing page and select Frontends. Then, select the forwarding rule and run a curl command to hit the external IP address. We can see an IAP-generated response that is true. If you click on the External IP address link, you should see the 302 redirections to accounts.google.com. This confirms that IAP is configured and protecting traffic, even though you won't be able to access the application itself due to the use of a self-signed certificate.

## 5.Results

After configuring IAP for a Google Compute Engine Instance, we will see an IAP-generated response that confirms the successful configuration. Clicking on the External IP address link should show 302 redirections to




accounts.google.com, and following display a page similar to the one show.

## 6.Future Scope

Cloud Identity-Aware Proxy (IAP) is a Google Cloud Platform service that provides secure access to web applications and VMs hosted on Google Cloud. If you are planning to set up a cloud IAP for Compute Engine, here are some potential future scopes that you could consider:

1. Multi-factor Authentication: You can enhance security by adding multi-factor authentication to the IAP login process. This feature can be configured to prompt users for additional verification, such as a fingerprint scan, in addition to their username and password.
2. Integration with Cloud Security Services: You can integrate Cloud IAP with other Google Cloud security services like Cloud Armor, Cloud Audit Logging, and Cloud Security Command Centre to enhance security and monitoring capabilities.
3. Scaling: As your organization grows, the number of users and applications that require access to your Compute Engine instances will increase. You can plan to scale your IAP deployment to accommodate future growth.
4. API Access Management: You can use Cloud IAP to provide secure access to APIs that are hosted on Compute Engine. This feature can help you control access to sensitive data and resources.
5. Identity Federation: You can integrate Cloud IAP with identity federation services like Google Workspace, Okta, or Active Directory to enable seamless authentication and access management across your organization's systems and applications.
6. Compliance: If your organization operates in a regulated industry, you can leverage Cloud IAP's compliance capabilities to help meet your regulatory requirements. Cloud IAP is compliant with various security standards, including SOC 2, HIPAA, and PCI DSS.

Overall, setting up Cloud IAP for Compute Engine provides a secure and convenient way to manage access to your organization's applications and data. As you plan for the future, you can consider these and other potential areas of expansion to further enhance the security and usability of your IAP deployment.

## 7. Conclusion

In conclusion, setting up an Identity-Aware Proxy (IAP) for Compute Engine on Google Cloud Platform (GCP) can help you control and secure access to your Compute Engine cases. By using IAP, you can circumscribe access to authorized druggies and groups, and reduce the threat of unauthorized access to your cases. To set up IAP for Compute Engine, you need to have a GCP design with a running Compute Engine case, a Cloud cargo Balancer to frontal- end the case, and a Google account to subscribe in to pierce the case. You also need to insure that your website or operation is duly configured to use HTTPS business. It's important to precisely follow the recommended stylish practices and review the attestation to insure a secure and dependable IAP setup. also, it's important to duly secure and configure your Compute Engine case before enabling IAP.

## References

*[1] Securing access to applications with Identity-Aware Proxy (IAP): https://cloud.google.com/iap/docs/app-engine-quickstart*
*[2] Using Identity-Aware Proxy (IAP) for Compute Engine: https://cloud.google.com/iap/docs/compute-engine-quickstart*
*[3] Setting up SSL for an App Engine Application: https://cloud.google.com/appengine/docs/standard/java11/securing-custom-domains-with-ssl*
*[4] Cloud Identity and Access Management (IAM) Documentation: https://cloud.google.com/iam/docs*
*[5] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, ''Cloud computing virtualization of resources allocation for distributed systems, ''J. Appl. Sci. Technol. Trends, vol. 1, no. 3, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.*
*[6] N. Zanoon, ''Toward cloud computing: Security and performance, ''Int. J. Cloud Comput.: Services Archit., vol. 5, no. vol. 5, nos. 5–6, pp. 17–26, Dec. 2015, doi: 10.5121/ijccsa.2015.5602.*
*[7] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, ''Proposing a load balanc- ing algorithm for the optimization of cloud computing applications, ''in Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS), Dec. 2019, pp. 1–6, doi: 10.1109/MACS48846.2019.9024785.*
*[8] B. Singh and G. Singh, ''A study on virtualization and hypervisor in cloud computing, ''Int. J. Comput. Sci. Mobile Appl., vol. 6, no. 1, pp. 17–22, 2018.*
*[9] P. Kumar and R. Kumar, ''Issues and challenges of load balancing tech- niques in cloud computing: A survey, ''ACM Comput. Surv., vol. 51, no. 6, pp. 1–35, Feb. 2019, doi: 10.1145/3281010.*
*[10] Z. A. Almusaylim and NZ. Jhanjhi, ''Comprehensive review: Privacy protection of user in location-aware services of mobile cloud computing, ''Wireless Pers. Commun., vol. 111, pp. 541–564, Oct. 2019.*