



Cloud Secure Web Proxy (SWP) Evaluation Guide

Cloud SWP Overview

Google Cloud SWP is a cloud-first service that provides a secure web gateway to help you secure egress web traffic (HTTP/S). You can configure your clients to explicitly use Cloud SWP as a proxy.

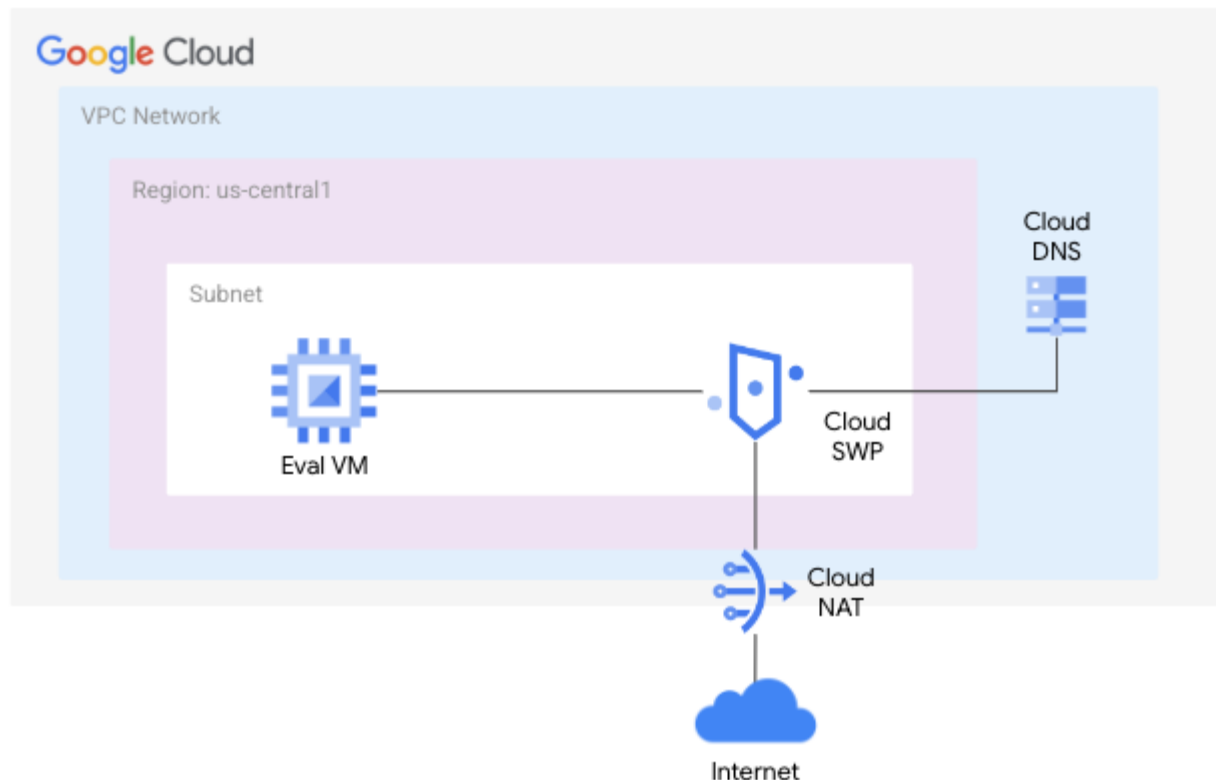


Purpose

This guide walks through the key steps on how to set up and evaluate the core capabilities of Cloud SWP in your Google Cloud environment. For a complete set of Cloud SWP how-to guides and API reference, please visit the [Cloud SWP documentation](https://cloud.google.com/secure-web-proxy/docs).

The guide will cover the following steps:

1. Initial Cloud SWP set up
2. Creating a Linux virtual machine (VM) instance for evaluation
3. Creating a Cloud SWP policy
4. Creating a Cloud SWP rule
5. Creating a Cloud SWP instance
6. Enable TLS Inspection (Optional)
7. Testing connectivity



Before you begin

You will need the following:

- Cloud SWP [Identity and Access Management \(IAM\) roles](#):
 - roles/compute.networkAdmin - This role enables you to configure policies and provision a Cloud SWP proxy instance
 - roles/certificatemanager.editor - This role enables you to explicitly upload Cloud SWP TLS certificates
 - roles/compute.orgSecurityPolicyAdmin (optional) - If you have a set of users responsible for ongoing policy management, this role will allow them to manage the security policies
 - (Optional - only if applying TLS inspection) roles/privateca.caManager - This role enables you to create CA pools and CAs which can then be used to generate certificates to perform TLS inspection
- Create or select a project



- Create a project

```
gcloud projects create <PROJECT_ID>
```

- Select a project

```
gcloud config set project <PROJECT_ID>
```

Replace **<PROJECT_ID>** with the project ID that you want.

- Enable billing and APIs
 - Ensure billing is enabled for your Google Cloud project. Learn how to [verify the billing status of your projects](#)
 - Enable APIs using: `gcloud services <SERVICE>.googleapis.com`
 - Enable the Compute Engine API ([compute.googleapis.com](#))
 - Enable the Certificate Manager API ([certificatemanager.googleapis.com](#))
 - Enable the Network Security API ([networksecurity.googleapis.com](#))
 - Enable the Network Services API ([networkservices.googleapis.com](#))
 - (Optional - only if applying TLS inspection) Enable the Certificate Authority Service API ([privateca.googleapis.com](#))
- Create or identify a VPC network in the selected project

```
gcloud compute networks create <NETWORK> \  
  --subnet-mode=custom \  
  --bgp-routing-mode=global \  
  --mtu=1460
```

Replace **<NETWORK>** with the VPC network name that you want.

Step 1: Initial set up Cloud SWP

Cloud SWP will need a proxy subnet in your selected VPC network and an SSL certificate uploaded to Certificate Manager.

1. Create a proxy subnet for each region where you want to deploy Cloud SWP:
 - The subnet size should be at least /26 (or 64 proxy-only addresses)



- A subnet size of /23 (or 512 proxy-only addresses) is recommended since Cloud SWP connectivity is provided by a pool of IP addresses reserved for Cloud SWP. The pool is used to allocate unique IP addresses on the egress side of each proxy for interaction with Cloud NAT and destinations in the VPC network

```
gcloud compute networks subnets create <PROXY_SUBNET> \  
  --purpose=REGIONAL_MANAGED_PROXY \  
  --role=ACTIVE \  
  --region=<REGION> \  
  --network=<NETWORK> \  
  --range=<IP_RANGE>
```

Replace the following:

- **<PROXY_SUBNET>** with the name you want for your proxy subnet
- **<REGION>** with your preferred region. This guide uses the **us-central1** region, however, you may replace this with your desired region that is supported by Cloud SWP.
- **<NETWORK>** with the VPC network you selected
- **<IP_RANGE>** with a subnet range, such as 192.168.0.0/23

2. Create an SSL certificate:

```
openssl req -x509 -newkey rsa:2048 \  
  -keyout <KEY_PATH> \  
  -out <CERTIFICATE_PATH> -days 365 \  
  -subj '/CN=<SWP_HOSTNAME>' -nodes -addext \  
  "subjectAltName = DNS:<SWP_HOSTNAME>"
```

Replace the following:

- **<KEY_PATH>** with path to save the key, such as ~/key.pem
- **<CERTIFICATE_PATH>** with the path to save the certificate, such as ~/cert.pem
- **<SWP_HOSTNAME>** with the hostname for your Cloud SWP instance, such as myswp.example.com

3. Upload the SSL certificate to Certificate Manager:

```
gcloud certificate-manager certificates create <CERTIFICATE_NAME> \  

```



```
--certificate-file=<CERTIFICATE_FILE_PATH> \  
--private-key-file=<KEY_FILE_PATH> \  
--location=<REGION>
```

Replace the following:

- **<CERTIFICATE_NAME>** with the preferred name of your certificate.
- **<CERTIFICATE_FILE_PATH>** with the path to the certificate file.
- **<KEY_FILE_PATH>** with the path to the key file.
- **<REGION>** with your preferred region. This guide uses the **us-central1** region, however, you may replace this with your desired region that is supported by Cloud SWP.

For more information about SSL certificates, see [SSL certificates overview](#).

4. Set environment variables

```
export PROJECT_ID="<PROJECT_ID>"  
export NETWORK="<NETWORK>"  
export SUBNET="<SUBNET>"  
export REGION="us-central1"  
export ZONE="us-central1-a"  
export CERTIFICATE="<CERTIFICATE_NAME>"  
export SWP_HOSTNAME="<SWP_HOSTNAME>"
```

Replace the following:

- Replace **<PROJECT_ID>** with your project ID
- Replace **<NETWORK>** with your VPC network
- Replace **<SUBNET>** with the subnet you will use to deploy the SWP instance and a test VM
- This example uses the **us-central1** region and **us-central1-a** zone, however, you may replace this with your desired zone that is supported by SWP
- Replace **<CERTIFICATE_NAME>** with your SSL certificate name in Certificate Manager
- Replace **<SWP_HOSTNAME>** with the hostname for your Cloud SWP instance

See [here](#) more information on setting up Cloud SWP.



Step 2: Creating a Linux Virtual Machine (VM) Instance

This Linux VM instance will be used to test connectivity and functionality to Cloud SWP. First, you will need to create a firewall rule to allow SSH access to the VM instance through Identity-Aware Proxy (IAP).

1. Run the following command to create a firewall rule allowing SSH through IAP to VM instances with the 'ssh' tag:

```
gcloud compute firewall-rules create allow-iap-ssh \  
  --direction=INGRESS \  
  --action=ALLOW \  
  --rules=tcp:22 \  
  --source-ranges=35.235.240.0/20 \  
  --target-tags=ssh \  
  --network=${NETWORK}
```

Note: The network flag is optional. If you do not specify the network, the default network for the current project will be used.

2. Run the following command to create a Linux VM instance and apply the 'ssh' tag:

```
gcloud compute instances create swp-test-vm \  
  --subnet ${SUBNET} \  
  --zone ${ZONE} \  
  --image-project debian-cloud \  
  --image-family debian-11 \  
  --no-address \  
  --tags=ssh
```



Step 3: Creating a Cloud SWP Policy

A Secure Web Proxy (SWP) policy is a set of rules, destination lists, and protections defined and ultimately applied to the SWP instance. Here you will create the policy before adding a rule in the subsequent step.

3. Use your preferred text editor to create the file *policy.yaml* with the following details:

```
description: basic Cloud SWP policy
name: projects/${PROJECT_ID}/locations/${REGION}/gatewaySecurityPolicies/policy1
```

4. Run the following command to create the Cloud SWP policy:

```
gcloud alpha network-security gateway-security-policies import policy1 \
  --source=policy.yaml --location=${REGION}
```



Step 4: Creating a SWP Rule

A Secure Web Proxy (SWP) rule contains the traffic matching parameters and actions to apply. Traffic matching parameters include session matcher, TLS decryption, application matcher, and action. Additionally, you may reference URL lists that consist of supported domains, wildcard domains, URLs or URL patterns. Rules are created within a policy.

Let's create a simple SWP rule that allows access to *example.com* by matching on the domain. This rule will be added to the previously created SWP policy.

1. Use your preferred text editor to create the file *rule.yaml* with the following details:

```
name:
projects/${PROJECT_ID}/locations/${REGION}/gatewaySecurityPolicies/policy1/rules/allow-exam
ple-com
description: Allow example.com
enabled: true
priority: 1
basicProfile: ALLOW
sessionMatcher: host() == 'example.com'
```

- **priority:** Lower integers indicate higher priorities
- **basicProfile:** Determines if traffic is allowed or denied
- **sessionMatcher:** Matches against the following session-level [attributes](#):
 - IP address of the source machine
 - Service account of the source machine
 - Secure Tag of the source machine
 - Hostname of the target machine
- **applicationMatcher (optional):** Matches against the following HTTP request [attributes](#):
 - URL
 - Path
 - Request headers



-
2. Run the following command to create the Cloud SWP rule:

```
gcloud alpha network-security gateway-security-policies rules import allow-example-com \  
  --source=rule.yaml \  
  --location=${REGION} \  
  --gateway-security-policy=policy1
```



Step 5: Creating a SWP Instance

Let's create the SWP instance and apply the previously created policy.

1. Use your preferred text editor to create the file *proxy.yaml* with the following details:

```
name: projects/<PROJECT_ID>/locations/<REGION>/gateways/swp1
type: SECURE_WEB_GATEWAY
addresses: ["10.128.0.99"]
ports: [443]
certificateUrls: ["projects/<PROJECT_ID>/locations/<REGION>/certificates/<CERTIFICATE>"]
gatewaySecurityPolicy:
  projects/<PROJECT_ID>/locations/<REGION>/gatewaySecurityPolicies/policy1
network: projects/<PROJECT_ID>/global/networks/<NETWORK>
subnetwork: projects/<PROJECT_ID>/regions/<REGION>/subnetworks/<SUBNET>
scope: samplescope
```

Note: While this example assigns 10.128.0.99 to the SWP instance, it may be necessary to allocate an IP address from your specified `${SUBNET}`

This *proxy.yaml* file requires hardcoding the constant values. Replace the following variables:

- `<PROJECT_ID>` with your previously used project ID
- `<REGION>` with your previously used region
- `<CERTIFICATE>` with your previously used certificate
- `<NETWORK>` with your previously used VPC network
- `<SUBNET>` with your previously used subnet

Note: Before running the following command, ensure Cloud NAT, Cloud Routers, and Regional Internal Load Balancers are not deployed in your VPC. Cloud SWP will deploy Cloud NAT and a Cloud Router as part of the Cloud SWP instance being deployed. These limitations will be removed at GA.

2. Run the following command to create the Cloud SWP instance:

```
gcloud alpha network-services gateways import swp1 \
  --source=proxy.yaml \
  --location=${REGION}
```



Step 6 (Optional): Enable TLS Inspection

1. Create a certificate authority (CA) pool:

```
gcloud privateca pools create swp-tls-pool \  
  --tier=devops \  
  --project=${PROJECT_ID} \  
  --location=${REGION}
```

2. Create a restrictive certificate issuance policy for the CA pool

```
cat << EOF > ca-policy.yaml  
maximumLifetime: 604800s  
baselineValues:  
  caOptions:  
    isCa: false  
  keyUsage:  
    extendedKeyUsage:  
      serverAuth: true  
identityConstraints:  
  allowSubjectPassthrough: false  
  allowSubjectAltNamesPassthrough: true  
  celExpression:  
    expression: 'subject_alt_names.all(san, san.type == DNS && san.value ==  
"${SWP_HOSTNAME}")'  
EOF
```

3. Apply the CA policy

```
gcloud privateca pools update swp-tls-pool \  
  --location=${REGION} \  
  --issuance-policy ca-policy.yaml
```

4. Create a root CA



```
gcloud privateca roots create swp-root-ca --pool=swp-tls-pool \
  --location=${REGION} \
  --auto-enable \
  --subject="CN=my-ca, O=Test LLC"
```

5. Grant the Secure Web Proxy access to the CA pool by first creating a service account:

```
gcloud beta services identity create \
  --service=networksecurity.googleapis.com \
  --project=${PROJECT_ID}
```

In response, the gcloud CLI creates a service account called `service-${PROJECT_ID}@gcp-sa-networksecurity.iam.gserviceaccount.com`.

6. Next, grant permissions for the service account to generate certificates from your CA pool:

```
gcloud privateca pools add-iam-policy-binding swp-tls-pool \
  --member='serviceAccount:<SERVICE_ACCOUNT>' \
  --role='roles/privateca.certificateRequester'
```

Replace , `<SERVICE_ACCOUNT>` with the service account created in the previous step.

7. Use your preferred text editor to create the file `tls-policy1.yaml` with the following details:

```
name: projects/${PROJECT_ID}/locations/${REGION}/tlsInspectionPolicies/tls-policy1
caPool: projects/${PROJECT_ID}/locations/${REGION}/caPools/swp-tls-pool
```

Note: This file references the previously created CA pool, `swp-tls-pool`.

Depending on the text editor used, the `tls-policy1.yaml` file might require hardcoding of the constant values. If needed, replace the following variables:

- `${PROJECT_ID}` with your previously used project ID
- `${REGION}` with your previously used region



8. Create the TLS inspection policy

```
gcloud alpha network-security tls-inspection-policies import tls-policy1 \  
  --source=tls-policy1.yaml \  
  --location=${REGION}
```

9. Use your preferred text editor to modify the existing file *policy.yaml* by adding the following line to the end:

```
...  
tlsInspectionPolicy:  
  projects/${PROJECT_ID}/locations/${REGION}/tlsInspectionPolicies/tls-policy1
```

10. Run the following command to overwrite the original Cloud SWP policy with the new Cloud SWP policy that includes TLS inspection:

```
gcloud alpha network-security gateway-security-policies import policy1 \  
  --source=policy.yaml --location=${REGION}
```

11. Use your preferred text editor to modify the existing file *rule.yaml* by adding the following lines to the end:

```
...  
applicationMatcher: request.path.contains('index.html')  
tlsInspectionEnabled: true
```

12. Run the following command to apply the Cloud SWP rule:

```
gcloud alpha network-security gateway-security-policies rules import allow-example-com \  
  --source=rule.yaml \  
  --location=${REGION} \  
  --gateway-security-policy=policy1
```



Step 7: Testing Connectivity

1. Connect to the VM that you previously provisioned:

```
gcloud compute ssh swp-test-vm --zone us-central1-a --tunnel-through-iap
```

2. Test the Cloud SWP instance:

```
curl -i -x https://10.128.0.99:443 -k https://example.com/index.html --proxy-insecure
```

Note: This example uses an IP address of **10.128.0.99** assigned to the SWP instance, however, you may need to change this to an IP address within your specified **\$(SUBNET)**.

You should receive an *HTTP/2 200* indicating that the connection was successful and traffic was proxied through Cloud SWP.

Press CTRL+C to exit, if needed.

You may try connecting to another website. All other websites should be blocked with an *HTTP 403* status code since Cloud SWP only has one rule allowing traffic to the *example.com* domain (specifically to example.com/index.html if you have added TLS inspection).

3. Verify connectivity via SWP logs:
 - a. In the Google Cloud console, go to the [Logging](#) page.
 - b. When in the Logs Explorer, enter the following filter to see logs for Cloud SWP and traffic going to **93.184.216.34** (<https://example.com>).
Note: You may need to perform `dig example.com` in the Cloud Shell to verify the IP address of *example.com*.

```
resource.labels.gateway_type="SECURE_WEB_GATEWAY"  
httpRequest.serverIp="93.184.216.34:443"
```

You should see a log with an HTTP request sourced from the IP address of *swp-test-vm* and destined to the IP address of *https://example.com*. This log message has an HTTP status code of 200 indicating that the connection was allowed and successful.



Logs Explorer

REFINE SCOPE Project

SHARE LINK LEARN

Query Recent (3) Saved (0) Suggested (1) Library

Clear query Save Stream logs Run query

Last 1 hour Search all fields

Resource Log name Severity +2 filters Show query

1 resource.labels.gateway_type="SECURE_WEB_GATEWAY"

2 httpRequest.serverIp="93.184.216.34:443"

Log fields Histogram

Create metric Create alert Jump to now More actions

Query results 2 log entries

Find in results Correlate by Download

SEVERITY TIMESTAMP CST SUMMARY EDIT

Showing logs for last 1 hour from 2/27/23, 3:17 PM to 2/27/23, 4:17 PM. Extend time by: 1 hour Edit time

>	2023-02-27 16:06:05.946 CST	{"@type":"type.googleapis.com/google.cloud.loadbalancing.type.LoadBalancerLogEntry"}
▼	2023-02-27 16:07:26.037 CST	<div><div>{"@type":"type.googleapis.com/google.cloud.loadbalancing.type.LoadBalancerLogEntry"}</div><div><div><div>httpRequest: { latency: "180.263833s" protocol: "HTTP/1.1" remoteIp: "10.200.1.4:47774" requestMethod: "CONNECT" requestSize: "1448" responseSize: "6073" serverIp: "93.184.216.34:443" status: 200 userAgent: "curl/7.74.0" }</div><div>insertId: "141ea39d2xaw"</div><div>jsonPayload: {</div></div></div></div>

Hide log summary

Expand nested fields

Copy to clipboard

Copy link

Similar entries



Clean up

To avoid incurring additional charges to your Google Cloud account for the resources used throughout this guide, follow these steps to delete the resources that were created:

1. Delete the Cloud SWP instance 'swp1':

```
gcloud alpha network-services gateways delete swp1 \
  --location=${REGION}
```

2. Locate and delete the Cloud NAT gateway associated with Cloud SWP.

List the Cloud Routers associated with Cloud SWP:

```
gcloud compute routers list | grep 'swg-autogen-router'
```

Delete the Cloud NAT gateway associated with the Cloud Router shown in the previous output:

```
gcloud compute routers nats delete swg-autogen-nat --router=<swg-autogen-router-####> \
  --region=${REGION}
```

Replace `<swg-autogen-router-####>` with the name of your Cloud Router

3. Delete the Cloud Router associated with Cloud SWP:

```
gcloud compute routers delete <swg-autogen-router-####> --region=us-central1
```

Replace `<swg-autogen-router-####>` with the name of your Cloud Router

4. Delete the rule *allow-example*:

```
gcloud alpha network-security gateway-security-policies rules delete allow-example-com \
  --location=${REGION} \
```




```
--project=${PROJECT_ID} \  
--gateway-security-policy=policy1
```

5. Delete the Cloud SWP policy *policy1*:

```
gcloud alpha network-security gateway-security-policies delete policy1 \  
--location=${REGION}
```

6. Delete the Linux VM instance *swp-test-vm*:

```
gcloud compute instances delete swp-test-vm
```

7. Delete the **<CERTIFICATE>** in Certificate Manager:

```
gcloud certificate-manager certificates delete <CERTIFICATE> --location=${REGION}
```

Replace **<CERTIFICATE>** with the name of the certificate.

8. Delete the **<PROXY_SUBNET>**:

```
gcloud compute networks subnets delete <PROXY_SUBNET> --region=${REGION}
```

Replace **<PROXY_SUBNET>** with the name of the proxy subnet created in the beginning.

TLS Clean up

9. Delete the **<SERVICE_ACCOUNT>** created for the Secure Web Proxy to have access to the CA Pool:

```
gcloud iam service-accounts delete <SERVICE_ACCOUNT> --project=${PROJECT_ID}
```

Replace **<SERVICE_ACCOUNT>** with the name of the email address of the service account created in Step 6, Enable TLS Inspection, sub-step 5. The service account should follow this format: `service-{PROJECT_ID}@gcp-sa-networksecurity.iam.gserviceaccount.com`.



10. Delete the Cloud SWP TLS inspection policy, *tls-policy1*:

```
gcloud alpha network-security tls-inspection-policies delete tls-policy1 \
  --location=${REGION}
```

11. Disable and delete the root CA:

```
gcloud privateca roots disable swp-root-ca --pool=swp-tls-pool \
  --location=${REGION}

gcloud privateca roots delete swp-root-ca --pool=swp-tls-pool \
  --location=${REGION}
```

12. Delete the CA pool:

```
gcloud privateca pools delete swp-tls-pool \
  --project=${PROJECT_ID} \
  --location=${REGION}
```

Note: You must wait for the expiration of the recovery period (7 days) of the CA before you may delete the CA pool.

That completes the guide on how to deploy and evaluate a Cloud Secure Web Proxy (SWP) instance in your Google Cloud environment.
