

Host Defender Auto Deployment from SaaS based Prisma Cloud User Guide

Introduction

[Prisma Cloud](#) provides comprehensive visibility and threat detection for cloud workload in Google Cloud. Prisma Cloud software consists of two components: Console and Defender. Console is Prisma Cloud's management interface. It lets you define policy and monitor your environment. For the Prisma Cloud SaaS edition, the Console is hosted by Palo Alto Networks. Defender is deployed to Google Cloud environment to secure the cloud workload. Defender protects your environment according to the policies set in Console. There are a number of [Defender types](#), Host Defender utilizes Prisma Cloud's model-based approach for protecting hosts that do not run containers.

Host Defender Auto Deployment allows Prisma Cloud customers to deploy Prisma Cloud Host Defender (Security agent) from SaaS based Prisma Cloud Console to the virtual machines (VM) aka compute engine instances in your Google Cloud project automatically. Google Cloud [Guest Policy](#) manages Host Defender Auto Deployment to VMs. You can choose the target VMs based on the Guest Policy Assignment. The auto deployment use two of Google Cloud Guest Policy Assignments to allow you to target a group of VMs by using one of the following characteristics:

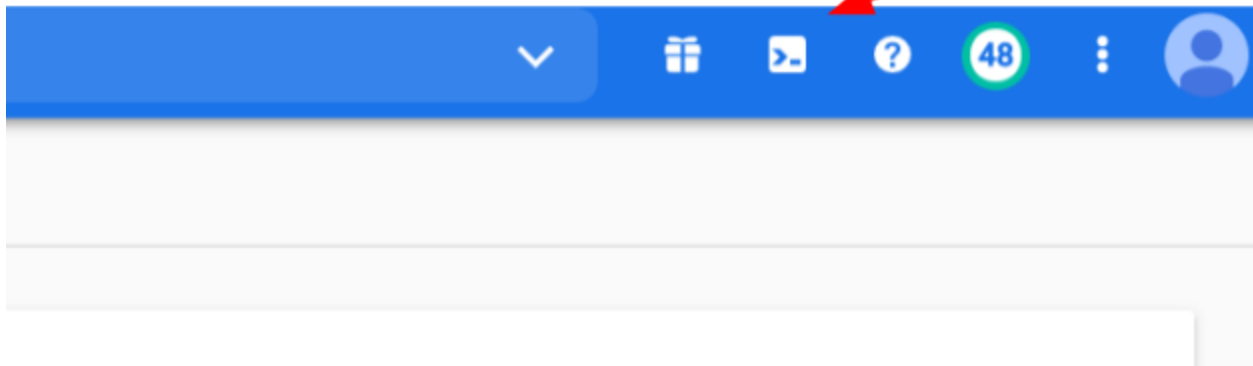
- Instance name prefix. See [example 2](#).
- Instance label. See [example 3](#).

Guest Policy supported various versions of Linux and Window OS, you can find a completed list of OS at [LINK](#),

Prepare your Google Cloud Project

You will need to prepare your Google Cloud Project with required API, Service Account and Secret before launching the Host Defender Auto Deployment.

In the upper-right corner of the Cloud Console, click the Cloud Shell icon to open the Cloud Shell.



Start by setting a variable for your project ID with this command:

```
export project_id=[YOUR PROJECT ID]
```

Verify the success of your variable creation by echoing the value:

```
echo $project_id
```

Next, create an environment variable for your username:

```
export user_id=[YOUR USER ID]
```

Verify the success of your variable creation by echoing the value:

```
echo $user_id
```

Select your account and project

```
gcloud config set account $user_id  
gcloud config set project $project_id
```

Permission for creating Guest Policy

Owners of a project have full access to create and manage policies. For all other users, you need to grant GuestPolicy Admin permission for managing the Guest Policies with the following gcloud command.

- GuestPolicy Admin (roles/osconfig.guestPolicyAdmin). Contains permissions to create, delete, update, get, and list guest policies.

```
gcloud projects add-iam-policy-binding $project_id \  
  --member user:$user_id \  
  --role roles/osconfig.guestPolicyAdmin
```

Enable the OS Config API in your project

```
gcloud services enable osconfig.googleapis.com
```

Configure the project metadata

```
gcloud compute project-info add-metadata \  
  --metadata=enable-guest-attributes=true,enable-osconfig=true,enable-os-config-debug=true,  
  osconfig-log-level=debug
```

```
Your Cloud Platform project in this session is set to paloaltonetworksgcp-public.  
Use "gcloud config set project [PROJECT_ID]" to change to a different project.  
myan@cloudshell:~ (paloaltonetworksgcp-public)$ gcloud services enable osconfig.googleapis.com  
myan@cloudshell:~ (paloaltonetworksgcp-public)$ gcloud compute project-info add-metadata \  
> --metadata=enable-osconfig=true  
Updated [https://www.googleapis.com/compute/v1/projects/paloaltonetworksgcp-public].  
myan@cloudshell:~ (paloaltonetworksgcp-public)$
```

Run the following command to confirm the project metadata is setup properly

```
gcloud compute project-info describe
```

```

use 'gcloud config set project [PROJECT_ID]' to change to a different project.
myan@cloudshell:~ (paltoaltonetworksgcp-public)$ gcloud compute project-info describe
commonInstanceMetadata:
  fingerprint: Qwx3Im4G9UI=
  items:
    - key: enable-guest-attributes
      value: 'TRUE'
    - key: enable-osconfig
      value: 'true'
    - key: gke-cluster-daniell-b25f0804-secondary-ranges
      value: services:default:default:gke-cluster-daniell-services-b25f0804,pods:default:default:gke-cluster-daniell-pods-b25f0804
    - key: gke-cluster-test-pcc-mp-12e0fb96-secondary-ranges
      value: services:default:default:gke-cluster-test-pcc-mp-services-12e0fb96,pods:default:default:gke-cluster-test-pcc-mp-pods-12e0fb96
    - key: osconfig-log-level
      value: debug

```

Deploy Token Refresher

The Prisma Cloud Compute API Token used to securely retrieve software is valid up to 60 minutes. The Host Defender Auto Deployment deploys Prisma Cloud Host Defender (Security agent) from Prisma Cloud Console to the virtual machines (VM) aka compute engine instances in your Google Cloud project per the guess policy you configure. This process requires a valid Prisma Cloud Compute API Token from time to time to ensure the automatic deployment of Host Defender when new compute instances spin up. To address this you will deploy a Token Refresher script to keep the token valid.

The Token Refresher script is a community supported script that refreshes the Prisma Cloud Console token associated with cert download for auto-install of defender agent.

Prepare for Token Refresher

Token Refresher refreshes the token through refreshing a secret stored in Secret Manager.

Enable the Following APIs using the gcloud command below:

1. Compute Engine API
2. Cloud Functions API
3. Cloud Logging API
4. Cloud Pub/Sub API
5. Cloud Build API (required by GCP for the Functions API)
6. Cloud Scheduler API
7. Cloud Secretes Manager
8. Cloud Storage API (Different than Cloud Storage)

```

gcloud services enable compute.googleapis.com cloudfunctions.googleapis.com
logging.googleapis.com pubsub.googleapis.com cloudscheduler.googleapis.com
cloudbuild.googleapis.com storage.googleapis.com secretmanager.googleapis.com

```

Create a Service Account for the Token Refresher script.

Enter the following command in Google Cloud Shell and select “Enter”. This will create a service account named “Token-Refresher-Function” and assign the service account a description of “Token Refresher function SA”

NOTE: If you get a request to authorize cloud shell to execute this script, click “Authorize.”

```
gcloud iam service-accounts create token-refresher-function \
  --description "Token Refresher function SA" \
  --display-name="Token Refresher function SA"
```

Verify the service account is created. Run the following command and locate the service account:

```
gcloud iam service-accounts list
```



Token Refresher function SA	token-refresher-function@paloalto-networks-gcp-public.iam.gserviceaccount.com	Token Refresher function SA	False
-----------------------------	---	-----------------------------	-------

From the cloudshell output, copy the full email ID of your newly created service account, and add it to an environment variable with the following command:

```
export service_account_id=[FULL SERVICE ACCOUNT EMAIL ID]
```

```
$ export service_account_id=token-refresher-function@panw-gcp-team-testing.iam.gserviceaccount.com
```

The Token Refresher script will require access to a storage bucket to be created in a later step. To grant permission for the Token Refresher script to access secretmanager admin role to the function service account use this gcloud command:

```
gcloud projects add-iam-policy-binding $project_id \
  --member serviceAccount:$service_account_id \
  --role roles/secretmanager.admin
```

Operational Steps Outline

1. Cloud Scheduler is a Cron tool that utilizes AppEngine to process Google-managed scheduled events. The Cloud Scheduler job will be used to trigger the Pub/Sub topic that will trigger a Cloud Function to refresh the API Token.
2. The Pub/Sub Topic is the trigger that activates the Cloud Function.
3. The Cloud Function runs the Python code to refresh the token by running a “requests.get” against the token endpoint on your Prisma instance.
4. The Prisma instance returns the token to the Cloud Function.

All of these steps will need to be completed by an individual with GCP organization admin or project admin rights plus token management access to the Prisma Cloud Compute Console.

Retrieve Prisma Cloud Compute Console URL and Token

1. Login to your Prisma Cloud Console and get the access information:
 - a. The URL to access Prisma Cloud Console
Prisma Cloud > Compute > Manage > System, select **Downloads** tab, locate “**Path to Console**” at the bottom of the page, click **copy** button and paste to your notepad. The URL should be in this format:

`https://us-west1.cloud.twistlock.com/us-3-159237196`

where “us-3-159237196” is the tenant ID. You will need the console address to configure the Cloud Function in a later step.

Create Secret for Prisma Cloud Token.

Navigate to the GCP Secret Manager in the GCP Console, create a secret with the name **host-defender-gcp-secure-deployment**, the value is the token you copied in the previous step.

Note: You must use the exact secret name above, or the auto deployment would fail.

Google Cloud Platform | paloaltonetworksgcp-public | Search products and resources

Security

Create secret

Name
host-defender-gcp-secure-deployment
The name should be unique and identifiable

Secret value
Input your secret value or import it directly from a file.

Upload file BROWSE
Maximum size: 64 KiB

Secret value
UpUZVhOMFpXMGdRvTJ0YVc0aWZRLjY4cHhjNnICWFdiNFMzaENIM1RGZUQ0WDJL
YkNNcWpQN0t1Z2NEeThld1kiLCJleHAiOjE2MDgyMjQ2ODIsImZcyI6InR3aXN0bG9ja
yJ9.IH3MYj-QdFxlNGp_tJw6XLeEMtuiZyOv_Nvcspg975s

CREATE SECRET CANCEL

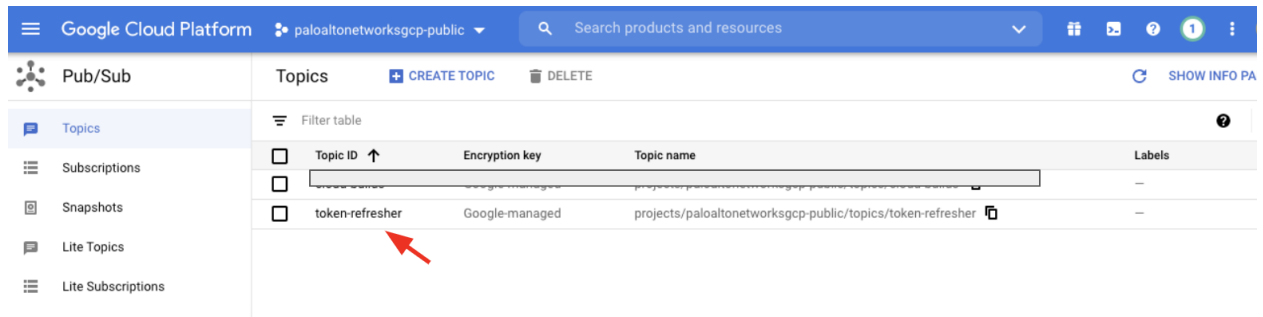
Create a Pub/Sub Topic

Now that you have the service account for your Cloud Function, you'll need to create a Pub/Sub Topic that acts as the trigger for the function.

```
gcloud pubsub topics create token-refresher
```

```
myan@cloudshell:~/host-defender-token (paloaltonetworksgcp-public)$ gcloud pubsub topics create token-refresher  
Created topic [projects/paloaltonetworksgcp-public/topics/token-refresher].
```

From the hamburger menu at the top left, select Pub/Sub > Topics, and verify that your topic is listed.



Create a GCP Function


Clone the repository containing the Token Refresher source code:

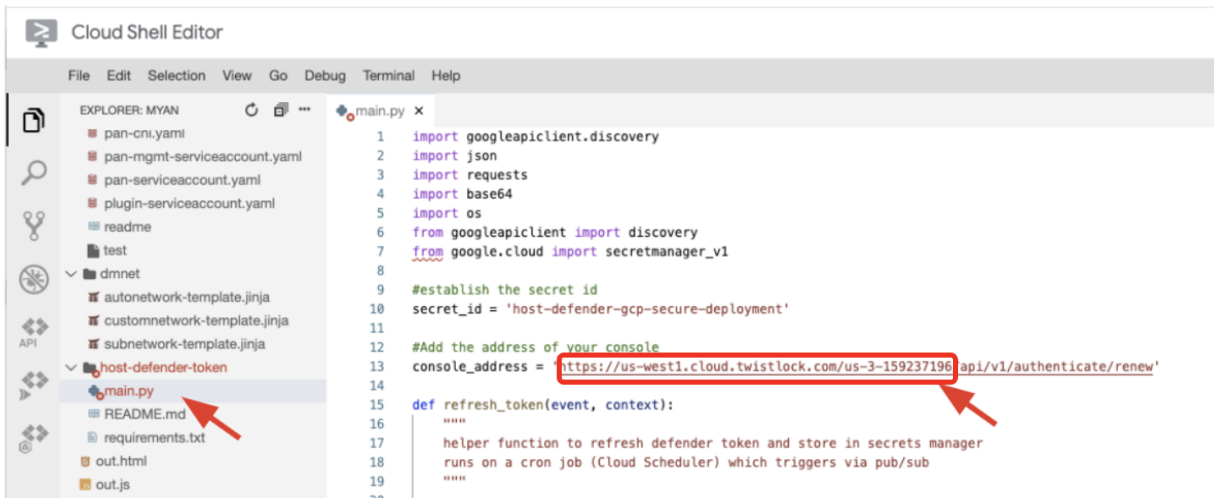
```
git clone https://github.com/PaloAltoNetworks/host-defender-token-refresher.git
cd host-defender-token-refresher
```

If you'd like, spend some time looking over the code. A Cloud Function requires two main code files - main.py, requirements.txt. Google Cloud has API-specific Python libraries to interact with their services. You use the requirements.txt to add the required libraries and APIs reference.

Configure Cloud Function

The cloud function will use a python script named main.py. We will update main.py script with your Prisma Cloud Compute Console **URL**. Make sure to replace the URL only and leave the rest in place.

Open the Editor  and navigate to main.py under host-defender-token folder



Replace the **Console Address** with the URL you copied from Prisma Cloud Console.

Run the following command to deploy the Cloud Function:

```
gcloud functions deploy pcc-token-refresher --region "us-central1"
--trigger-topic=token-refresher --entry-point=refresh_token --runtime=python37
--service-account $service_account_id
```

It may take several minutes for your Cloud Function to complete deploying. Important pieces of the command above are the entry point, which simply means the first function that should run during the operation; your service account; and trigger topic, which is the PubSub topic created previously.

```
myan@cloudshell:~/host-defender-token (palaoaltonetworksgcp-public)$ gcloud functions deploy pcc-token-refresher --region "us-central1" --trigger-topic=token-refresher
--entry-point=refresh_token --runtime=python37 --service-account $service_account_id
Created .gcloudignore file. See 'gcloud topic gcloudignore' for details.
Deploying function (may take a while - up to 2 minutes)...
For Cloud Build Stackdriver Logs, visit: https://console.cloud.google.com/logs/viewer?project=palaoaltonetworksgcp-public&advancedFilter=resource.type%3Dbuild%0Aresource.labels.build_id%3D77f039e6-258a-4f7d-807d-81759986c58c%0AlogName%3Dprojects%2Fpalaoaltonetworksgcp-public%2Flogs%2Fcloudbuild
Deploying function (may take a while - up to 2 minutes)...done.
availableMemoryMb: 256
buildId: 77f039e6-258a-4f7d-807d-81759986c58c
entryPoint: refresh_token
eventTrigger:
  eventType: google.pubsub.topic.publish
  failurePolicy: {}
  resource: projects/palaoaltonetworksgcp-public/topics/token-refresher
  service: pubsub.googleapis.com
ingressSettings: ALLOW_ALL
labels:
  deployment-tool: cli-gcloud
name: projects/palaoaltonetworksgcp-public/locations/us-central1/functions/pcc-token-refresher
runtime: python37
serviceAccountEmail: token-refresher-function@palaoaltonetworksgcp-public.iam.gserviceaccount.com
sourceUploadUrl: https://storage.googleapis.com/gcf-upload-us-central1-14b8fc50-d3f9-4389-a84d-afaf942e607c/7ee90317-30cc-4f5f-913b-d81cb8af6346.zip?GoogleAccessId=service-1026607622258gcf-admin-robot.iam.gserviceaccount.com&Expires=1608266949&Signature=EdX7ssDGvUgJlec91Bmxn521kcwByvIejKGq42f1jCgoMHGqA1eN2RK1Rp6Jduod2F3DMyXQ5Sx0TfoIq24ei6falpdzFco4z2Foi13nKObYF8ox2stctcILHuMxWmqjWv0ITeVNDx7Dbz3zuiB32f7BHn65L4B351kx7F1cU0vaJ1b3v7x42BRMo3JGwi39pnaR8tjHP42B8Hs42B7GY4yVJdJb1q1cGsQaUWg3AduxDCD2ek2TnwX9z8GVfrc8gD10Kku0hzhMcHrOrVlx145s0pGmN2BhtaytSKUV42PMrvBKy1CEvQ42B3XEJOkm5ww442FuzxUG7PhE1FFrWlUGUsisXQ43D43D
status: ACTIVE
timeout: 60s
updateTime: '2020-12-18T04:21:15.929Z'
versionId: '1'
```

Create cron job via Cloud Scheduler

Cloud Scheduler is a Cron tool that utilizes AppEngine to process Google-managed scheduled events. The Cloud Scheduler job will be used to trigger the Pub/Sub topic that acts as the trigger for the Cloud Function.

```
gcloud scheduler jobs create pubsub refresh-token-cron --schedule="*/5 * * * *"
--topic=token-refresher --message-body=foobar
```

```
myan@cloudshell:~/host-defender-token (palaltonetworksgcp-public)$ gcloud scheduler jobs create pubsub refresh-token-cron --schedule="*/5 * * * *" --topic=token-refre
sher --message-body=foobar
name: projects/palaltonetworksgcp-public/locations/us-central1/jobs/refresh-token-cron
pubsubTarget:
  data: 2m9vYmFy
  topicName: projects/palaltonetworksgcp-public/topics/token-refresher
retryConfig:
  maxBackoffDuration: 3600s
  maxDoublings: 16
  maxRetryDuration: 0s
  minBackoffDuration: 5s
schedule: '*/5 * * * *'
state: ENABLED
timeZone: Etc/UTC
userUpdateTime: '2020-12-18T04:32:55Z'
myan@cloudshell:~/host-defender-token (palaltonetworksgcp-public)$
```

Verify the token is refreshing

Navigate to **Logging > Log Explorer**, enter the following into the **Query builder** and click **Run Query**. Locate “DEBUG: <Response [200]>”, it is the indicator of a success run of “pcc-token-refresher”

```
resource.type="cloud_function"
resource.labels.function_name="pcc-token-refresher"
"DEBUG"
```

Google Cloud Platform | paloaltonetworkgcp-public | Search products and resources

Operations
Logging

Logs Explorer

Query builder | Recent (7) | Saved (0) | Suggested (2)

Resource | Log name | Severity

1 resource.type="cloud_function"
2 resource.labels.function_name="pcc-token-refresher"
3 "DEBUG"

Query results

SEVERITY	TIMESTAMP	PST	SUMMARY
> 0	2020-12-18 13:25:02.198	PST	pcc-token-refresher t0duyg91j409 Function execution took 1212 ms, finished with status: 'ok'
> 0	2020-12-18 13:30:01.013	PST	pcc-token-refresher t0durif03h01 Function execution started
> 1	2020-12-18 13:30:01.340	PST	pcc-token-refresher t0durif03h01 DEBUG: <Response [200]>
<pre>{ textPayload: "DEBUG: <Response [200]>" insertId: "000000-5c6ba2ac-c7bf-431c-9bba-5f3879549251" resource: (2) timestamp: "2020-12-18T21:30:01.340Z" severity: "INFO" labels: (1) logName: "projects/paloaltonetworkgcp-public/logs/cloudfunctions.googleapis.com%2Fcloud-functions" trace: "projects/paloaltonetworkgcp-public/traces/2c5a2d88f5d67be9a8af7755f62f8d36" receiveTimestamp: "2020-12-18T21:30:11.576289688Z" }</pre>			
> 0	2020-12-18 13:30:02.192	PST	pcc-token-refresher t0durif03h01 Function execution took 1180 ms, finished with status: 'ok'
> 0	2020-12-18 13:35:01.044	PST	pcc-token-refresher t0dulheukfwr Function execution started
> 1	2020-12-18 13:35:01.370	PST	pcc-token-refresher t0dulheukfwr DEBUG: <Response [200]>
> 0	2020-12-18 13:35:02.212	PST	pcc-token-refresher t0dulheukfwr Function execution took 1169 ms, finished with status: 'ok'
> 0	2020-12-18 13:40:00.089	PST	pcc-token-refresher t0dufhgryvau Function execution started

Debug response code 401 indicates a failed Cloud Function run.

Logs Explorer | OPTIONS | REFINE SCOPE | Project | SHARE LINK | 6:01:00 AM - 8:01:00 AM | PAGE LAYOUT | LEA

Query preview
resource.type="cloud_function" resource.labels.function_name="pcc-token-refresher" "DEBUG"

Query results

To view more results, expand the time range for this query. Extend time by: 1 hour Edit time

SEVERITY	TIMESTAMP	PST	SUMMARY
> 0	2020-12-18 06:05:03.097	PST	pcc-token-refresher 98kmh47rf9ry Function execution started
> 1	2020-12-18 06:05:07.797	PST	pcc-token-refresher 98kmh47rf9ry DEBUG: <Response [401]>
> 0	2020-12-18 06:05:07.821	PST	pcc-token-refresher 98kmh47rf9ry Function execution took 4726 ms, finished with status: 'crash'
> 0	2020-12-18 06:10:11.260	PST	pcc-token-refresher rbyybxh70kp7 Function execution started
> 1	2020-12-18 06:10:14.117	PST	pcc-token-refresher rbyybxh70kp7 DEBUG: <Response [401]>
> 0	2020-12-18 06:10:14.135	PST	pcc-token-refresher rbyybxh70kp7 Function execution took 2877 ms, finished with status: 'crash'
> 0	2020-12-18 06:15:05.510	PST	pcc-token-refresher ewbnwdp16s2m Function execution started
> 1	2020-12-18 06:15:09.153	PST	pcc-token-refresher ewbnwdp16s2m DEBUG: <Response [401]>
> 0	2020-12-18 06:15:09.161	PST	pcc-token-refresher ewbnwdp16s2m Function execution took 3654 ms, finished with status: 'crash'
> 0	2020-12-18 06:20:03.883	PST	pcc-token-refresher j4may1kcn1rs Function execution started
> 1	2020-12-18 06:20:06.776	PST	pcc-token-refresher j4may1kcn1rs DEBUG: <Response [401]>
> 0	2020-12-18 06:20:06.789	PST	pcc-token-refresher j4may1kcn1rs Function execution took 2908 ms, finished with status: 'crash'
> 0	2020-12-18 06:25:17.538	PST	pcc-token-refresher 3x15xol2bxfg Function execution started
> 1	2020-12-18 06:25:31.356	PST	pcc-token-refresher 3x15xol2bxfg DEBUG: <Response [401]>

Delete the secret **host-defender-gcp-secure-deployment** and recreate it with updated token from Prisma Cloud

Secret Manager [+ CREATE SECRET](#)

Secret Manager lets you store, manage, and secure access to your application secrets.

[Learn more](#)

Filter table

<input checked="" type="checkbox"/>	Name ↑	Location	Encryption	Labels	Created	Actions
<input checked="" type="checkbox"/>	host-defender-gcp-secure-deployment	Automatically replicated	Google-managed	None	1/6/21, 2:03 PM	<div><div></div><div>Add new version</div><div>Disable all versions</div><div>Delete</div><div>Copy Resource ID</div></div>

1 secret selected

You may also see the secrets get updated every 5 minutes at Security Manager

Google Cloud Platform [paloaltonetworksgcp-public](#)

Security [← Prisma Cloud](#)

Security Command Center [Customer Information](#)

secret

PRODUCTS & PAGES

Secret Manager
Security

Secret Manager lets you store, manage, and secure access to your application secrets.

[Learn more](#)

Filter table

<input type="checkbox"/>	Name ↑	Location	Encryption	Labels	Created
<input type="checkbox"/>	host-defender-gcp-secure-deployment	Automatically replicated	Google-managed	None	12/18/20, 1

No secrets selected

Select secret “**host-defender-gcp-secure-deployment**”, click **Action** and **View secret value**

Google Cloud Platform paloaltonetworksgcp-public Search products and resources

Security

- Security Command Center
- reCAPTCHA Enterprise
- Threat Detection
- Context-Aware Access
- Identity-Aware Proxy
- Access Context Manager
- VPC Service Controls
- Binary Authorization
- Data Loss Prevention
- Cryptographic Keys
- Certificate Authority Service

Secret details DELETE

Secret: "host-defender-gcp-secure-deployment"
projects/1026607622258/secrets/host-defender-gcp-secure-deployment

OVERVIEW **VERSIONS**

Versions + NEW VERSION ENABLE SELECTED DISABLE SELECTED DESTROY SELECTED

<input type="checkbox"/>	Version	Status	Encryption	Created on ↓	Actions
<input type="checkbox"/>	1	Enabled	Google-managed	12/17/20, 9:09 PM	<div>View secret value Disable Destroy Copy Resource ID</div>

No versions selected

Version 1 of "host-defender-gcp-secure-deployment"

Enabled

Secret value

RGd5TmnpnMk9URXNJbIZ6WlhKdVIXMWxJam9pYmisaGJrQndZV3h2WVd4MGlyNWxkSGR2
Y210ekxtTnZiU0lzSW5WelpYSiNiMnhsVG1GdFpTSTZJbE41YzNSbGJTQkJaRzFwYmlKOS5G
Vm1aX1pjQnV1SHoyQlIUczFsNFUtMGV6bmhCWE1RdDRUOVp2SzNiYUhFlwiZXhwljoxNjA4
MjcxNjkxLzJpc3MiOiJ0d2lzdGxvY2sifQ.2kXUHEky40JAgLLagwGk6JaOp0CstC3QLs_-
e0JuVQM

DONE

Confirm that the secret value has been updated with a new token.

Deploy Host Defender from Marketplace

Prepare Compute Instance

VM instances need the permission to access the secret

host-defender-gcp-secure-deployment you created in the section above. You will add the Secret Admin role to the service account you use for VM instances. The following is using the default service account, you may use another service account.

Navigate to IAM & Admin > Service accounts, locate the default service account for VM instance

Google Cloud Platform PANW-GCP-TEAM-TESTING

IAM & Admin

Service accounts + CREATE SERVICE ACCOUNT DELETE

Service accounts for project "PANW-GCP-TEAM-TESTING"

A service account represents a Google Cloud service identity, such as code running on Compute Engine VMs, App Engine apps, or systems running outside Google. [Learn more about service accounts.](#)

Organization policies can be used to secure service accounts and block risky service account features, such as automatic IAM Grants, key creation/upload, or the creation of service accounts entirely. [Learn more about service account organization policies.](#)

Filter table

<input type="checkbox"/>	Email	Status	Name ↑	Description	Key ID	Key creation date	Actions
<input type="checkbox"/>	287979962335-compute@developer.gserviceaccount.com	✓	Compute Engine default service account		No keys		⋮

Service account ID

SHOW INFO PANEL

Navigate to Secret Manager, select the secret

host-defender-gcp-secure-deployment, click show info panel at up right corner

Security

Secret Manager + CREATE SECRET

Secret Manager lets you store, manage, and secure access to your application secrets. [Learn more](#)

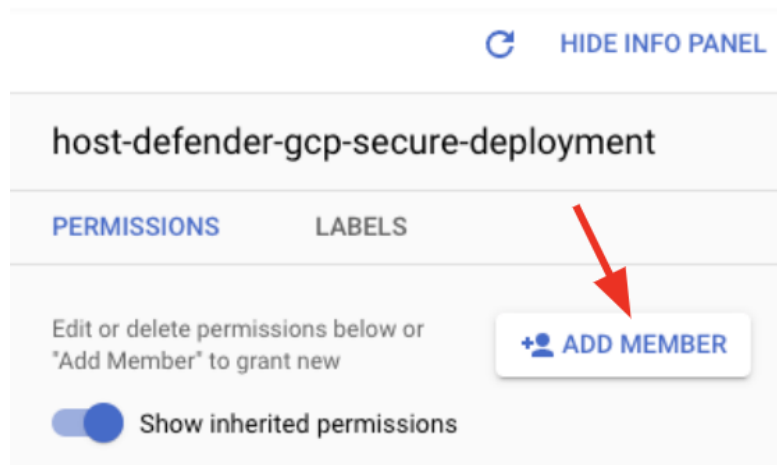
Filter table

<input checked="" type="checkbox"/>	Name ↑	Location	Encryption	Labels	Created	Actions
<input checked="" type="checkbox"/>	host-defender-gcp-secure-deployment	Automatically replicated	Google-managed	None	1/6/21, 8:50 PM	⋮

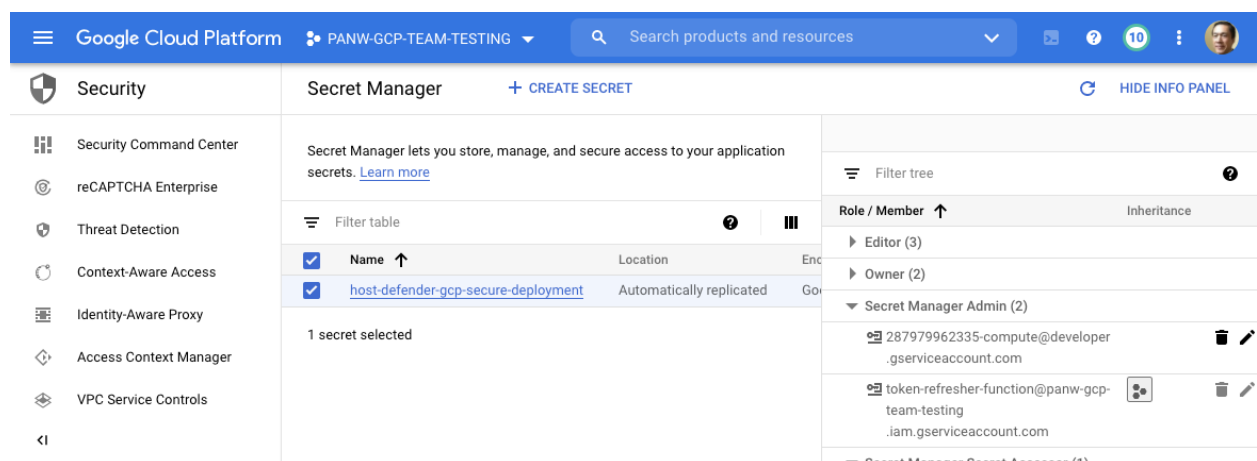
1 secret selected

SHOW INFO PANEL

Click **Add Member**




Add the Compute Engine service account under New members. Select a role of **Secret Manager Admin** and **Save**



Deploy Host Defender

Create a Guest Policy

Navigate to marketplace and search for Prisma Cloud Host Defender



Prisma Cloud Host Defender

Palo Alto Networks, Inc.

Security for Virtual Machines

VISIT PALO ALTO NETWORKS, INC. SITE TO SIGN UP ↗

Enter the name of the Guest Policy ID you would like to use. Select Prisma Console endpoint and Prisma Tenant ID (you can find the information at your **Prisma Cloud Console**, navigate to **Compute > Manage > System** you did in a previous step).

Google Cloud Platform

PANW-GCP-TEAM-TESTING

Search products and resources

Security

Prisma Cloud

Security Command Center

reCAPTCHA Enterprise

Threat Detection

Context-Aware Access

Identity-Aware Proxy

Access Context Manager

VPC Service Controls

Binary Authorization

Data Loss Prevention

Cryptographic Keys

Certificate Authority Service

Secret Manager

Customer Information

Guest Policy ID *

now-is-offical

The Guest Policy ID will be used to uniquely identify a specific policy.

This deployment will create the following Guest Policies:

agent-deploy-panw-linux-now-is-offical

agent-deploy-panw-win-now-is-offical

Select Prisma Console endpoint *

us-west1.cloud.twistlock.com

The Prisma Console URL that agents should connect to.

Prisma Tenant ID *

us-3-159237196



VM Assignment


This guest policy ensures the agent is installed on any new or existing VM instances that match the assignment. If the assignment is empty, it applies to all instances. Otherwise, the targeted instances must meet ALL constraints specified.

DEPLOY

You may choose `Add label` or `add VM name prefix` to your guest policy. Refer to the [Google Cloud link](#) for more details.

Add label



[Click here for more information](#) 

Label Key *

Label Value *

CANCEL

DONE

ADD A VM LABEL

Enter a VM instance name prefix

VM instance name prefix

Click **Deploy**, it will bring you to the Guest Policy page. You may click **VIEW DETAILS** at right to view the details of the guest policy created.

Security

OS Guest Policies

DELETE

Security Command Center

reCAPTCHA Enterprise

Threat Detection

Context-Aware Access

<input type="checkbox"/>	Policy Name	Created	Updated	
<input type="checkbox"/>	agent-deploy-prisma-linux-now-is-offical	Jan 14, 2021, 8:53:33 AM	Jan 14, 2021, 8:53:33 AM	VIEW DETAILS
<input type="checkbox"/>	agent-deploy-prisma-windows-now-is-offical	Jan 14, 2021, 8:53:33 AM	Jan 14, 2021, 8:53:33 AM	VIEW DETAILS

Review the guest policy status with gcloud command

You may use the gcloud command-line tool to inspect the guest policies associated with your project and the compute instance.

1. Use the [os-config guest-policies list](#) command to list all your guest policies.

```
gcloud beta compute os-config guest-policies list
```

To review the guest policy in Google Cloud Console

https://console.cloud.google.com/security/agent/deployment/policies?project={project_id}

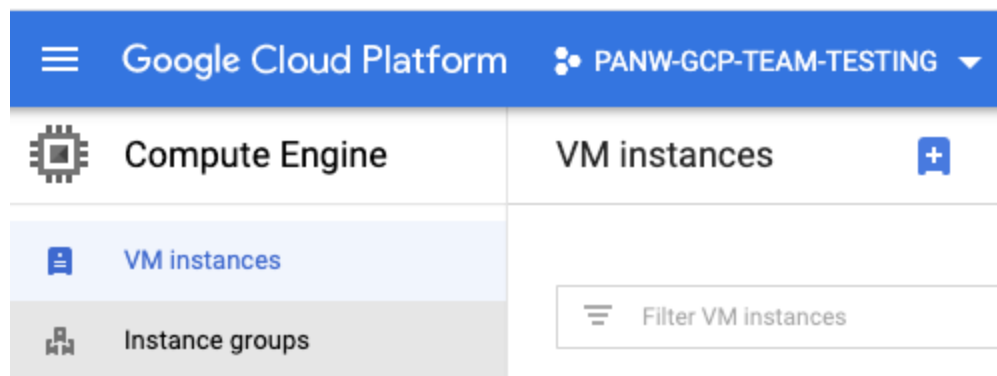
Deploying Host Defender creates OS Guest Policy. Click **VIEW DETAILS** to review the details of the Guest Policy.

2. From the list of guest policies, copy the IDs of the guest policies you would like to inspect, and then run the command to inspect each of the guest policies. Replace POLICY_ID with the policy ID that you want to review.

```
gcloud beta compute os-config guest-policies describe POLICY_ID
```

Create a VM instance

Navigate to Google Cloud Console, select **Compute Engine > VM Instance**



Select a name for your VM instance, make sure it matches the prefix if you use it when creating the guest policy in the previous step.

Name ?
Name is permanent

instance-1

Labels ? (Optional)

+ Add label

Region ? **Zone** ?

If you use a label when creating the guest policy in the previous step, make sure you add the matched label to your VM instance.

Manage labels

Labels entered here will be assigned at the time of instance creation. Labels for an existing instance may be edited on the instance details page.

Key	Value	
Key to add	empty	×
+ Add label		

If you use the Compute Engine default Service Account for your VM instance, make sure select **Allow full access to all Cloud APIs**

Identity and API access ?

Service account ?

Compute Engine default service account

Access scopes ?

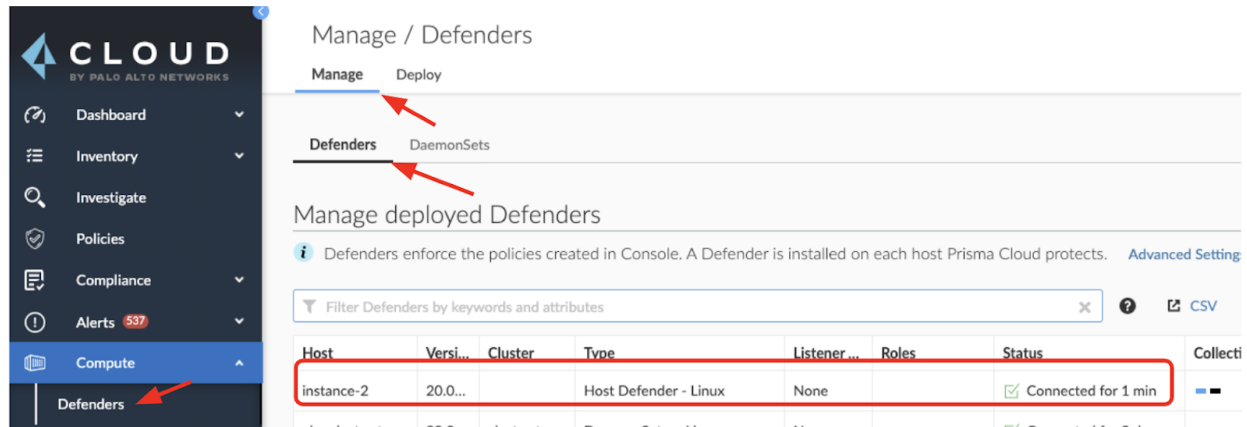
☐ Allow default access
☒ Allow full access to all Cloud APIs
☐ Set access for each API

Once the instance is created, the Host Defender should be installed automatically.

Valid the Host Defender is deployed

Login to your Prisma Cloud Console and confirm if the defender is deployed:

Prisma Cloud > Compute > Manage > Defender, select **Manage, Defenders**



Troubleshooting

If the VM doesn't show up at Prisma Cloud Compute console defender list. Go through the following steps:

Confirm the guest-policy desired state at the VM instance is "Installed"

You can also run the lookup command for a given instance to see which configurations apply to that VM instance.

```
gcloud beta compute os-config guest-policies lookup VM_NAME
```

```
myan@cloudshell:~ (palocaltonetworksgcp-public) $ gcloud beta compute os-config guest-policies lookup guest-policy
Did you mean zone [us-west1-b] for instance: [guest-policy] (Y/n)? n
No zone specified. Using zone [us-central1-a] for instance: [guest-policy].
```

SOFTWARE RECIPES			
SOURCE	NAME	VERSION	DESIRED_STATE
projects/1026607622258/guestPolicies/test	agent-deploy-prisma-linux-mypolicyid	1.0	INSTALLED

Review Guest Policy Logs

Review OSConfigAgent (Guest Policy) logs if any error.

Locate the Instance ID

Google Cloud Platform PANW-GCP-TEAM-TESTING Search products and res

Compute Engine

VM instances

Instance groups

Instance templates

Sole-tenant nodes

Machine images

Disks

Snapshots

Marketplace

VM instance details EDIT RESET CREATE M

instance-1

Details Monitoring Screenshot

Remote access

SSH

Connect to serial console

Enable connecting to serial ports

Logs

Cloud Logging

Serial port 1 (console)

More

Instance Id

4372977111290034116

Navigate to Logs Explore, run query:

```
resource.type="gce_instance"
resource.labels.instance_id="your instance ID"
logName="projects/your-project_id/logs/OSConfigAgent"
```

Here is an example of a successful run:

Google Cloud Platform PANW-GCP-TEAM-TESTING Search products and resources

Operations Logging

Logs Explorer OPTIONS REFINE SCOPE Project

SHARE LINK LAST 1 HOUR PAGE LAYOUT LEARN

Query builder Recent (26) Saved (0) Suggested (1)

Resource Log name Severity

1 resource.type="gce_instance"
2 resource.labels.instance_id="609089466373595371"
3 logName="projects/panw-gcp-team-testing/logs/OSConfigAgent"

Query results

Jump to Now Actions Configure

SEVERITY	TIMESTAMP	SUMMARY
severity: "INFO"		<p>labels: {3}</p> <p>logName: "projects/panw-gcp-team-testing/logs/OSConfigAgent"</p> <p>sourceLocation: {3}</p> <p>receiveTimestamp: "2021-01-14T05:43:47.996115446Z"</p>
>	2021-01-13 21:43:47.435 PST	"Creating working directory for recipe agent-deploy-prisma-linux-mypolicyid."
>	2021-01-13 21:43:47.436 PST	<p>"Running step 0: 'script_run:(script:\\\"AUTH_TOKEN=\$(gcloud secrets versions access latest --secret-host-defender-gcp-secure-deployment);curl -o defender.sh --sSL -k --header \\\"authorization: Bearer \$(AUTH_TOKEN)\\\" -X POST https://us-west1.cloud.twistlock.com/us-3-159237196/api/v1/scripts/defender.sh;sudo bash defender.sh -a https://us-west1.cloud.twistlock.com/us-3-159237196 -c us-west1.cloud.twistlock.com -h \\\"Bearer \$(AUTH_TOKEN)\\\" --install-host\\\" interpreter:SHHELL)'"</p> <p>{</p> <p>insertId: "e5i4ia25z4u2u"</p> <p>jsonPayload: {2}</p> <p>resource: {2}</p> <p>timestamp: "2021-01-14T05:43:47.436185594Z"</p> <p>severity: "DEBUG"</p>

Hide log summary Expand nested fields Copy to clipboard Copy link

2021-01-13 21:43:56.579 PST

"Combined output for "/bin/sh" command:
Downloading and extracting Defender image.
Downloading Twistlock scripts.
Generating certs for mason-version.us-central1-a.c.panw-gcp-team-testing.internal IP:127.0.0.1,IP:10.128.0.21.
Running twistlock.sh and installing Defender (skipping EULA).

```

-----
|_  _|_  _(-)---| | | --- ---| | --
| | \ \ / \ / / _|| _|| / _ \ / _|| / /
| | \ \ V / / \_ \ | | | (-) | (-) <
|_|  \_/\_ / | | ---/\_ \ | \_ \ / \_ \ | \_ \

Performing system checks for defender-server mode...
Installing Linux Server Defender to /opt/twistlock.
Installing systemd service twistlock-defender-server.service.

```

Possible errors

Connection timeout error

Failed downloading twistlock.cfg - curl: (7) Failed to connect to us-west1.cloud.twistlock.com port 443: Connection timed out.

Solution

There was a connection issue. May be caused by the guest policy not fully deployed. Please wait for a couple of minutes and deploy the VM again.

Permission denied error

The screenshot shows a log entry with a timestamp of 2021-01-06 21:07:37.524 PST. The log message is: "Running step 0: 'script_run:{script:\"AUTH_TOKEN=\$(gcloud secrets versions access latest --secret=host-defender-gcp-secure-deployment);curl -o defender.sh -sSL -k --header \"\"\"authorization: Bearer \${AUTH_TOKEN}\"\"\" -X POST https://us-west1.cloud.twistlock.com/us-3-159237196/api/v1/scripts/defender.sh;sudo bash defender.sh;sudo bash defender.sh -a https://us-west1.cloud.twistlock.com/us-3-159237196 -c us-west1.cloud.twistlock.com -h \"\"\"Bearer \${AUTH_TOKEN}\"\"\" --install-host\\\" interpreter:SHELL}\"". Below the log message, there is a JSON object with the following fields: insertId, jsonPayload, resource, timestamp, severity, labels, logName, sourceLocation, and receiveTimestamp. The severity is "DEBUG".

Below the log entry, there is a combined output for the "/bin/sh" command: "ERROR: (gcloud.secrets.versions.access) PERMISSION_DENIED: Request had insufficient authentication scopes."

Solution

1. Make sure the secret is refreshing, you should see code 200. If you see 401, recreate the secret with the current token. Use the following for log query:

```
resource.type="cloud_function"
resource.labels.function_name="pcc-token-refresher"
"DEBUG"
```

2. Give the full API access when deploying VM if use the default service account. You get Access Scope error if you select default

The screenshot shows the 'Identity and API access' dialog box. The 'Service account' dropdown is set to 'Compute Engine default service account'. The 'Access scopes' section has three radio buttons: 'Allow default access', 'Allow full access to all Cloud APIs' (which is selected), and 'Set access for each API'.

3. Provide VM service account the access of secret manager admin to the secret **host-defender-gcp-secure-deployment** before you deploy the VM

Appendix

Installing the Cloud Logging agent on a single VM

Use this procedure if the Cloud Logging agent is not installed automatically. The Logging agent streams logs from your VM instances and from selected third-party software packages to Cloud Logging. It is a best practice to run the Logging agent on all your VM instances. Note there is additional cost associated with the logging.

1. Open a terminal connection to your VM instance using SSH or a similar tool and ensure you have sudo access.
2. Change to a directory you have write access to, for example your home directory.
3. Add the agent's package repository:

```
curl -sO https://dl.google.com/cloudagents/add-logging-agent-repo.sh
```

```
sudo bash add-logging-agent-repo.sh
```

```
sudo apt-get update
```

4. Install the agent:

List the available versions of the agent in order to select which version to install:

```
sudo apt-cache madison google-fluentd
```

To install the latest version of the agent, run:

```
sudo apt-get install google-fluentd
```

Install the Configuration files. For unstructured logging, run:

```
sudo apt-get install -y google-fluentd-catch-all-config
```

5. Assign proper roles

You need to make sure your instance service-account has proper rights to edit/write data to StackDriver. Simply run following commands to assign proper roles:

```
gcloud projects add-iam-policy-binding PROJECT_NAME
```

```
--member="serviceAccount:SERVICE_ACCOUNT_EMAIL "
```

```
--role="roles/logging.logWriter"
```

```
gcloud projects add-iam-policy-binding PROJECT_NAME
```

```
--member="serviceAccount:SERVICE_ACCOUNT_EMAIL "
```

```
--role="roles/monitoring.metricWriter"
```

URL to Create a Guest Policy

You may use this link to create a guest policy directly

console.cloud.google.com/security/agent/deployment/prisma

URL to Guest Policy

You may use this link to access a guest policy directly

<https://console.cloud.google.com/security/agent/deployment/policies?project={project ID}>

