

Auto Scaling with the Panorama Plugin for GCP

Contact Information

Corporate Headquarters:

Palo Alto Networks

3000 Tannery Way

Santa Clara, CA 95054

www.paloaltonetworks.com/company/contact-support

About the Documentation

- To ensure you are viewing the most current version of this document, or to access related documentation, visit the Technical Documentation portal: docs.paloaltonetworks.com.
- To search for a specific topic, go to our search page: docs.paloaltonetworks.com/search.html.
- Have feedback or questions for us? Leave a comment on any page in the portal, or write to us at documentation@paloaltonetworks.com.

Copyright

Palo Alto Networks, Inc.

www.paloaltonetworks.com

© 2019-2020 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Last Revised

March 26, 2020

Table of Contents

Auto Scaling the VM-Series Firewall on Google Cloud Platform.....	4
Auto Scaling Components for Google Cloud Platform.....	5
Auto Scaling Requirements.....	5
Prepare to Deploy the Auto Scaling Templates.....	5
Deploy GCP Auto Scaling Templates.....	15
About the Auto Scaling Templates.....	15
Deploy the Firewall Template.....	16
Prepare a Service Project.....	17
Configure the Shared VPC.....	18
Configure a Peered VPC.....	19
Deploy the Application Template.....	19
Onboard a New Application.....	20
Parameters in the Auto Scaling Templates for GCP.....	30
Sample GKE Service Templates.....	36

Auto Scaling the VM-Series Firewall on Google Cloud Platform

The Panorama plugin for Google Cloud Platform (GCP) version 2.0.0 assists you in deploying the VM-Series firewall in GCP and enables Panorama to manage VM-Series firewalls securing VM monitoring or auto scaling deployments in GCP. Using Panorama for centralized policy and firewall management increases operational efficiency in managing and maintaining a distributed network of firewalls.

With Panorama maintaining your GCP [managed instance groups](#) you can create application enablement policies that protect and control the network.

The auto scaling deployment supports using a [shared VPC](#) network configuration or [VPC network peering](#) to create a common [VPC](#) network in which a host project contains shared VPC networks and the VM-Series firewalls, and a service project contains a vm-based or container-based application deployment (a Kubernetes cluster). Palo Alto networks supplies templates to help you deploy the VM-Series firewalls in the host project and deploy an optional sample application in the service project.

[BYOL and PAYG](#) licenses can be used for the VM-Series firewalls. During licensing, VM-Series firewall instances talk directly to the Palo Alto Networks license server.

If you choose BYOL your deployment can deactivate license instances in response to a scale-down event. If a VM-Series firewall's deployment information is configured in the Panorama plugin for GCP and the firewall is automatically removed, Panorama detects the firewall status and automatically deregisters the firewall.

- [Auto Scaling Components for Google Cloud Platform](#)
- [Deploy GCP Auto Scaling Templates](#)

Auto Scaling Components for Google Cloud Platform

Typical GCP auto scaling deployments use a host project and a service project and form a common [VPC](#) network between the two. The Panorama plugin for GCP can secure an auto scaling deployment in a single project with host and service VPCs, or host and service projects in a [shared VPC](#) or [peered VPC](#) network configuration, where the host project contains the VM-Series firewalls and the shared VPC networks, and the service project contains your application deployment. If your application is deployed in a Kubernetes cluster, a [peered VPC](#) is required.

- [Auto Scaling Requirements](#)
- [Prepare to Deploy the Auto Scaling Templates](#)

Auto Scaling Requirements

- ❑ **General Requirements**—Ensure your environment meets the basic [Requirements](#).
- ❑ **Panorama Plugin for GCP**—If you have not done so, [Install the Panorama Plugin for GCP](#).



If you previously installed the Panorama plugin for GCP version 1.0.0, remove it before you install 2.0.0. You cannot upgrade.

- ❑ **Palo Alto Networks Auto Scale templates version 1.0**—Palo Alto Networks provides the templates to deploy VM-Series firewall instances in the host project and configure and deploy a sample application in a service project. See [About the Auto Scaling Templates](#) for more about the templates.

Download the templates from [GitHub](#). The zip file contains separate zip files for the firewall and application templates.

Prepare to Deploy the Auto Scaling Templates

Complete the following tasks before you deploy the auto scaling templates.

- [Prepare a Host Project and Required Service Accounts](#)
- [Obtain a Licensing API Key](#)
- [Configure the Panorama Plugin for GCP to Secure an Auto Scaling Deployment](#)
- [Prepare a VM-Series Firewall Bootstrap Package for Auto Scaling](#)

Prepare a Host Project and Required Service Accounts

You need a host project and a service project to form the shared VPC topology that supports the firewall and application templates. You can create a new host project or prepare an existing project to act as your host.

To [set up the Shared VPC](#) an organization administrator must grant the host project administrator the Shared VPC Admin role. The Shared VPC Admin can [enable](#) a project to act as a host, and grant the Service Project Admin role to the service project administrator. Review the GCP documentation on [Administrators and IAM](#) roles.

STEP 1 | In the GCP console, create a GCP project to act as the host. If you want to use an existing project, skip to the next step.

To create a new project, select your organization or **No organization**, click **New Project** and fill in your project information. Note, this is your only chance to **EDIT** the project ID.



The Google Cloud SDK must be [installed](#) and configured so that you can authenticate with your host project from the CLI. You will use the command line interface to deploy the firewall template and the application template, and to attach the service project to the host project.

STEP 2 | Enable APIs and services required for auto scaling. The required APIs are:

- ☐ Cloud Pub/Sub API
- ☐ Cloud Deployment Manager API
- ☐ Cloud Storage API
- ☐ Compute Engine API
- ☐ Google Compute Engine Instance Group Manager API
- ☐ Google Compute Engine Instance Group Updater API
- ☐ Google Compute Engine Instance Groups API
- ☐ Kubernetes Engine API
- ☐ Stackdriver API
- ☐ Stackdriver Logging API
- ☐ Stackdriver Monitoring API

You can enable APIs from the [GCP console](#) or the [GCP CLI](#), as shown below.

Enable APIs from the GCP console

1. Select the host project, and from the Navigation menu, select **APIs & Services**.
2. Search for and view each required API.
3. **ENABLE** any APIs that do not display the “API enabled” status.

Enable APIs from the CLI

1. In the CLI, view your configuration to ensure that you are in the correct project.

```
gcloud config list
```

If not, set the project as follows:

```
gcloud config set project <project-name>
```

2. Issue the following commands to enable the required APIs.

```
gcloud services enable pubsub.googleapis.com
gcloud services enable deploymentmanager.googleapis.com
gcloud services enable storage-component.googleapis.com
gcloud services enable compute.googleapis.com
gcloud services enable replicapool.googleapis.com
gcloud services enable replicapoolupdater.googleapis.com
gcloud services enable resourceviews.googleapis.com
gcloud services enable container.googleapis.com
gcloud services enable stackdriver.googleapis.com
gcloud services enable logging.googleapis.com
gcloud services enable monitoring.googleapis.com
```

3. Confirm that the required APIs are enabled.

```
gcloud services list --enabled
```

STEP 3 | Create a service account for deploying the VM-Series firewall, and assign the IAM roles required for auto scaling a service or a Kubernetes cluster.

When you configure the firewall templates you add the email address for this service account to the VM-Series firewall `.yaml` file. Within the host project, the template uses credentials from this service account to create a host VPC with subnets, deploy VM-Series firewalls in the VPC, configure Stackdriver custom metrics, create a Pub/Sub topic, and more.

1. In the GCP console select **IAM & Admin** > **Service accounts** and select **+CREATE SERVICE ACCOUNT**.

Fill in the service account details and click **CREATE**.

2. Give the service account permission to auto-scale resources in this project.

Select a role type from the drop menu, and on the right, select an appropriate access level. For example, select Project > Editor. You can select multiple roles for a service account.

- ☐ Compute Engine > Compute Admin
- ☐ Compute Engine > Compute Network User
- ☐ Pub/Sub > Admin
- ☐ Monitoring > Monitoring Metric Writer
- ☐ Stackdriver > Stackdriver Accounts Editor
- ☐ Storage > Storage Admin
- ☐ (GKE only) Kubernetes > Kubernetes Engine Cluster Admin
- ☐ (GKE only) Kubernetes > Kubernetes Engine Viewer

Service account permissions (optional)

Grant this service account access to GCP-AutoScale-KK so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role	
Compute Admin	Full control of all Compute Engine resources.
Compute Network User	Access to use Compute Engine networking resources.
Editor	Edit access to all resources.
Pub/Sub Admin	Full access to topics, subscriptions, and snapshots.

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#) [CANCEL](#)

Continue when you are finished adding roles.

3. Click **+CREATE KEY** to create a key for the host service account.
 - (Optional) Add email addresses to grant other users or administrators access to this service account.
 - Click JSON to download the private key in JSON form.
 - Store the key in a safe location. You will need this key when you [Deploy GCP Auto Scaling Templates](#).
4. Click **DONE**.

STEP 4 | Create a service account that a Panorama administrator can use to interact with this host project.

1. In the GCP console select **IAM & Admin** > **Service accounts** and select **+CREATE SERVICE ACCOUNT**.
2. Fill in the service account details and click **CREATE**.
3. Grant service account access.

Select a role type from the drop menu, and on the right, select an appropriate access level. For example, select Project > Editor. You can select multiple roles for a service account.

- ☐ Compute Engine > Compute Viewer
- ☐ Deployment Manager > Viewer
- ☐ Pub/Sub > Admin

Click **CONTINUE**.

4. Click **+CREATE KEY** to create a key for the host service account.
 - (Optional) Add email addresses to grant other users or administrators access to this service account.
 - Select JSON to download the private key in JSON form.
 - Store the key in a safe location. You will need this key when you [Configure the Panorama Plugin for GCP to Secure an Auto Scaling Deployment](#).

STEP 5 | (optional) In the CLI, ensure you can communicate with your new host project.

1. Set your project to the host project you just created.

```
gcloud set project <your-autoscale-host-project-name>
```

2. Create a configuration for auto scaling. Your new configuration is automatically activated unless you disable activation.

```
gcloud config configurations create <CONFIGURATION_NAME> gcloud config list
```

Obtain a Licensing API Key

You need a Licensing API key so Panorama can license and de-license managed assets in GCP.

STEP 1 | On support.paloaltonetworks.com select **Assets** > **Licensing API** and click **Enable**. The key is displayed.



Only a Super User can view the Enable link to generate this key. See [How to Enable, Regenerate, Extend the Licensing API Key](#).

Licensing API Key

This license API key provides user
license API calls. To enable this

Key : 986a2d53dcf

STEP 2 | Select the key and copy it.

STEP 3 | From the CLI, SSH in to Panorama and issue the following command, replacing <key> with the API key you copied from the support portal:

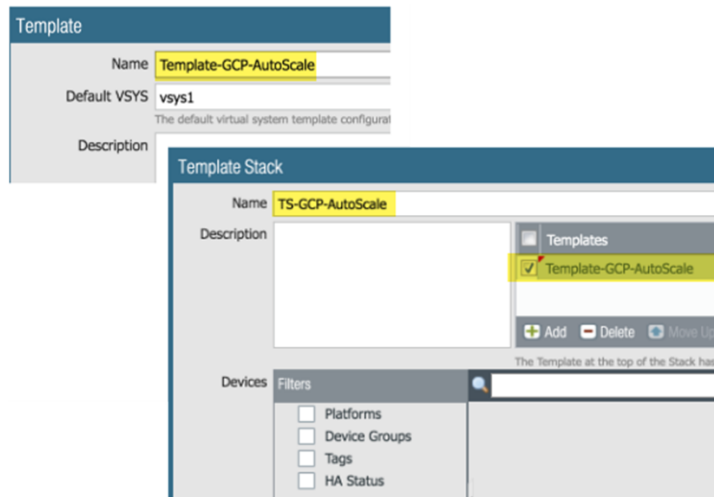
```
request license api-key set key <key>
```


API Key is successfully set

Configure the Panorama Plugin for GCP to Secure an Auto Scaling Deployment

In Panorama, create assets to support the auto scaling firewall deployment.

STEP 1 | Create a [template](#), and a [template stack](#) that includes the template, and **Commit** the changes.



STEP 2 | In the **Network** context, select either the template or the template stack. Select **Virtual Routers** and **Add** a virtual router.

When the firewall template creates static routes, they are added to this virtual router.



Define only one router for the auto scale deployment.



STEP 3 | In the **Network** context, select the template you created, select **Interfaces** and **Add Interface**.

- On the Config tab, select a slot, select the **Interface name** and select the Layer3 **Interface Type**. From the **Security Zone** menu, select **New Zone**, name the zone Untrust and click **OK**.
- On the IPv4 tab enable **DHCP Client** and **Automatically create default route pointing to default gateway provided by server** (enabled by default) and click **OK**.

Ethernet Interface		Ethernet Interface	
Slot	Slot 1	Slot	Slot 1
Interface Name	ethernet1/1	Interface Name	ethernet1/1
Comment		Comment	
Interface Type	Layer3	Interface Type	Layer3
Netflow Profile	None	Netflow Profile	None
Config IPv4 IPv6 A		Config IPv4 IPv6 Advanced	
Assign Interface To Virtual Router: None Virtual System: vsys1 Security Zone: Untrust		Type: <input type="radio"/> Static <input type="radio"/> PPPoE <input checked="" type="radio"/> DHCP Client <input checked="" type="checkbox"/> Enable <input checked="" type="checkbox"/> Automatically create default route pointing to default gateway provided by server <input type="checkbox"/> Send Hostname: system-hostname Default Route Metric: 10	

STEP 4 | Add the ethernet1/2 (Trust) Layer 3 interface.

- On the Config tab, chose the same slot as the previous step, select the **Interface name** (ethernet1/2), and select the Layer3 **Interface Type**. From the **Security Zone** menu, select **New Zone** name the zone Trust and click **OK**.
- On the IPv4 tab enable **DHCP Client**, disable **Automatically create default route pointing to default gateway provided by server** and click **OK**.

Ethernet Interface		Ethernet Interface	
Slot	Slot 1	Slot	Slot 1
Interface Name	ethernet1/2	Interface Name	ethernet1/2
Comment		Comment	
Interface Type	Layer3	Interface Type	Layer3
Netflow Profile	None	Netflow Profile	None
Config IPv4 IPv6 A		Config IPv4 IPv6 Advanced	
Assign Interface To Virtual Router: None Virtual System: vsys1 Security Zone: Trust		Type: <input type="radio"/> Static <input type="radio"/> PPPoE <input checked="" type="radio"/> DHCP Client <input checked="" type="checkbox"/> Enable <input type="checkbox"/> Automatically create default route pointing to default gateway provided by server <input type="checkbox"/> Send Hostname: system-hostname Default Route Metric: 10	

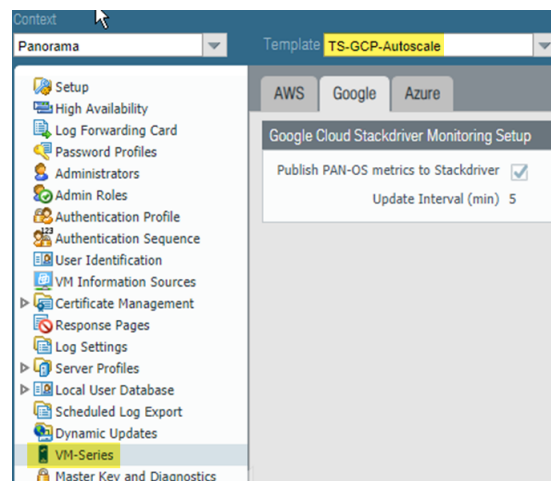
STEP 5 | Return to your template stack and the virtual router you created earlier. Place the untrust and trust interfaces (ethernet1/1 and ethernet1/2) in the virtual router, and click **OK**.



STEP 6 | Configure Stackdriver for your auto scaling deployment.

You must have the [VM-Series plugin on Panorama](#) to configure Stackdriver.

1. In the **Device** context, select the template stack you created earlier from the Template drop menu.
2. Select **Device > VM-Series > Google** and click the Edit cog (⚙️). Enable **Publish PAN-OS metrics to Stackdriver**.



3. Commit your changes.

STEP 7 | Create a Device Group that references the template or template stack you created in step 1.

This Device Group will contain the VM-Series firewalls you create with the firewall template.

1. Add a security policy that allows web-browsing traffic from Untrust to Trust.

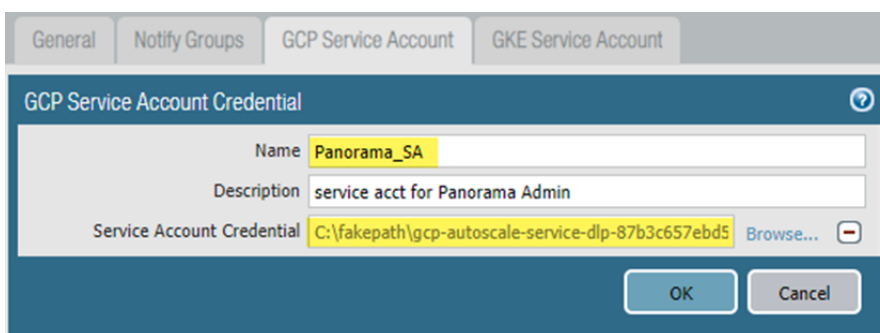
In the Policies context, select the Device Group you just created. Select **Security > Pre Rules** and **Add** the following security policy.

Panorama		Device Group DG-GCP-Autoscale-Firewalls											
<div>▼ Security</div> <div>Pre Rules</div> <div>Post Rules</div> <div>Default</div> <div>▼ NAT</div> <div>Pre Rules</div> <div>Post Rules</div>													
	Name	Location	Tags	Type	Source				Destination				
					Zone	Address	User	HIP Profile	Zone	Address	Application	Service	Action
1	allow-untrust-trust	DG-GCP-Autoscale-Firewalls	none	universal	Untrust	any	any	any	Trust	any	web-browsing	application-default	Allow

STEP 8 | Set up the GCP service account for the host project.

1. In the Panorama context, expand Google Cloud Platform, select **Setup**, and click **Add**.

2. Supply a name and description for the host service account you created in Step 4.
3. Upload the JSON credentials file you created in Step 4.4.

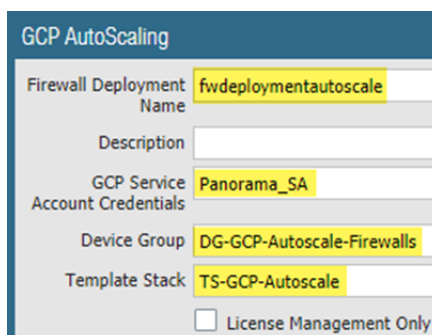


After you add a service account credential, you can validate the credential from your Panorama command line (you cannot validate from the web interface):

```
request plugins gcp validate-service-account gcp_service_account <svc-  
acct-credential-name>
```

STEP 9 | Set up auto scaling on the Panorama plugin for GCP.

1. In the Panorama context, expand Google Cloud Platform, select AutoScaling, and click **Add**.
2. Supply the Firewall Deployment Name and an optional description for the deployment.
3. For the GCP Service Account Credential, supply the GCP service account name from Step 8.



4. Chose the Device Group you created in Step 7, and the Template Stack you created in Step 1.
5. Disable **License Management Only** to ensure traffic is secured.

STEP 10 | Commit your changes.

Prepare a VM-Series Firewall Bootstrap Package for Auto Scaling

During bootstrap, the initial request from the firewall provides the host IP address and serial number, and the VM auth key so Panorama can validate the VM auth key and add the firewall as a managed device. Panorama can then assign the firewall to the appropriate device group and template so that you can centrally configure and administer the firewall using Panorama.

In this case, you must generate a VM auth key on Panorama and include the key in the init-cfg.txt file that you use for bootstrapping. The VM auth key allows Panorama to authenticate the newly bootstrapped VM-Series firewall. The bootstrap package must include.

- In the /config directory, an init-cfg.txt file that includes the Panorama IP address
- In the /license directory, the VM authentication key in a file named authcodes.

The lifetime of the key can vary between 1 hour and 8760 hours (1 year). After the specified time, the key expires and Panorama will not register VM-Series firewalls without a valid auth-key in this connection request.

STEP 1 | Set up a [Google storage bucket](#) with the folders required to [Bootstrap the VM-Series Firewall on Google Cloud Platform](#). You can use an existing bootstrap package or create a new bootstrap package, for these folders.

STEP 2 | Edit the values in the sample `init-cfg.txt` file to customize the file for your environment. The firewall templates include a sample `init-cfg.txt` file.

Parameter	Value	Comment
type	dhcp-client	
hostname	<pa-vm>	Optional name you assigned when you prepared the host project . Only required if a specific host is necessary, and dhcp-send-hostname is no.
vm-auth-key	<vmauthkey>	A key that Panorama must validate before adding a firewall as a managed device. See Generate the VM Auth Key On Panorama .
panorama-server	<panorama-ip>	The IP address of the Panorama management device you configured in Configure the Panorama Plugin for GCP to Secure an Auto Scaling Deployment
tplname	<template-stack-name>	The template stack you created in Configure the Panorama Plugin for GCP to Secure an Auto Scaling Deployment .
dgname	<dg-name>	The name of the Device Group you created in the Panorama Plugin for GCP.
dns-primary		Your primary DNS server.
dns-secondary		Your secondary DNS server.
dhcp-send-hostname	yes	Leave as is.
dhcp-send-client-id	yes	Leave as is.
dhcp-accept-server-hostname	yes	Leave as is.

Parameter	Value	Comment
dhcp-accept-server-domain	yes	Leave as is.

STEP 3 | Upload your edited `init-cfg.txt` file to the `/config` folder in your bootstrap package.

STEP 4 | If you are using BYOL, create a text file named `authcodes` (no extension), add your auth code, and upload the file to the `/license` folder.

Deploy GCP Auto Scaling Templates

- [About the Auto Scaling Templates](#)
- [Deploy the Firewall Template](#)
- [Prepare a Service Project](#)
- [Configure the Shared VPC](#)
- [Deploy the Application Template](#)
- [Onboard a New Application](#)
- [Sample GKE Service Templates](#)

About the Auto Scaling Templates

Download the Palo Alto Networks auto scaling templates from <https://github.com/PaloAltoNetworks/GCP-AutoScaling>. The zip file contains separate zips for firewall and application templates. Each zip is a template directory containing several files, but you only need to edit the YAML files.

- [Firewall Templates](#)
- [Application Template](#)

Firewall Templates

The firewall directory files create VM-Series firewalls and other deployment resources. They create new networks and the familiar subnetworks for the VM-Series firewall: management, untrust, and trust. They also deploy a Cloud Pub/Sub messaging service to relay information from GCP to the Panorama plugin for GCP. With this infrastructure in place, the plugin can leverage dynamic address groups to apply security policy on inbound traffic routed to services running on GCP, and use auto scale metrics to deploy VM-Series firewalls to meet increased demand for application workload resources or to eliminate firewalls that are no longer needed.

To configure your load balancer, edit the `.yaml` file for an external application load balancer (ALB) or network load balancer (NLB).

- **ALB** (HTTP External Load Balancer)

To customize an ALB, edit `vm-series-fw-alb.yaml`.

HTTP external load balancer is a proxy-based load balancer that performs SNAT and DNAT on the inbound traffic from Internet. The HTTP load balancer is designed to support only the 80 and 8080 TCP ports.

To support multiple applications using HTTP load balancer in load balancer sandwich architecture, we can use the GCP HTTP load balancer `urlMap` and `namedPort` to map different URLs to different ports in the load balancer. In turn, the VM-Series firewall can translate the ports to different applications, each represented by one internal load balancer per application.

- **NLB** (TCP Load Balancer)

To customize an NLB, edit `vm-series-fw-nlb.yaml`.

TCP load balancer is a non-proxy based load balancer, which means it doesn't perform NATing on inbound traffic from the Internet.

TCP load balancer in GCP allows adding multiple frontend IP addresses with an arbitrary port, making it possible to support multiple applications.

Another advantage of TCP load balancer is that the original client IP address is preserved, which is desirable for some applications.

Application Template

The application directory provides a sample application. You configure and deploy an internal load balancer (ILB) to enable your application servers to subscribe to the Pub/Sub service and communicate with your VM-Series firewalls and the GCP plugin on Panorama.

To customize the application template, edit `apps.yaml` as described in [Deploy the Firewall Template and Application Template](#).

Deploy the Firewall Template

Edit the [Firewall Templates](#) from the host project.

STEP 1 | Edit the `vm-series-fw-nlb.yaml` or `vm-series-fw-alb.yaml` environment variables to reflect your cloud environment.

The sample in this workflow is for the NLB. See [vm-series-fw-nlb.yaml](#) and [vm-series-fw-alb.yaml](#) for further explanation of the template parameters.

```
properties:
  region: us-east1
  zones:
  -us-east1-b
  # Do not modify the lb-type field.
  lb-type: nlb
  cloud-nat: yes
  forwarding-rule-port: 80
```

```
# Only one app is allowed
urlPath-namedPort-maps:
- appName: app1
```

```
# ssh key PUBLIC:
- optional
```

The autoscaling firewall template requires you to enter the value in single quotes and prepend the key with **admin:** followed by a space. This is the same convention used for the Google Marketplace template, as detailed in [SSH Key Pair](#). For example:

```
sshkey: 'admin: ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQDe0gJHd8okxPGWXsmdTdcZBJNI6ONT/NSz6INs2CNtKW
[REDACTED]
oTKXL8t0SRnOKaKV73NR5KnfpsNfwGxG8aQtEkeMCZIxX+6WOnRf/N4K3
yourname@.paloaltonetworks.local'
```

```
bootstrap-bucket: bootstrap-autoscale
```

```
image: vmseries-byol-814
machine-type: n1-standard-4
```

For the service-account, supply the email address for the [host project](#) service account you created earlier ([step 3](#)).


```
service-account: sa-pan@gcp-autoscale-kk.iam.gserviceaccount.com
```

The fw-instance-tag value will be the managed instance group name in the deployment.

```
fw-instance-tag: vm-series-fw
```

Choose one metric for auto scaling. Possible values are: panSessionActive, panSessionUtilization, DataPlaneCPUUtilizationPct, DataPlanePacketBufferUtilization, or panSessionUtilization.

```
metric: custom.googleapis.com/VMSeries/panSessionActive
```

```
max-size: 2
min-size: 1
target-type: GAUGE
util-target: 100
```

```
# Greenfield deployment
mgmt-network-cidr: 172.22.2.0/24
untrust-network-cidr: 172.22.1.0/24
trust-network-cidr: 172.22.3.0/24
mgmt-network-access-source-range:
- 199.167.54.229/32
- 199.167.52.5/32
mgmt-network-access-ports:
- 22
- 443
```

STEP 2 | Deploy the firewall template.

```
gcloud deployment-manager deployments create <your-template>
--config apps.yaml
--automatic-rollback-on-error
```

Take note of the outputs the CLI prints after the deployment—the subnet names, the deployment name, and the Panorama Pub/Sub topic name. You need these values to configure the Shared VPC and for the application template deployment.

The firewall deployment name must be configured in the Panorama plugin for GCP auto scaling definition.

Prepare a Service Project

Create a separate service project, or choose an existing project, for your application.

To learn more about host and service projects in a shared VPC, see the [Shared VPC Overview](#), and review the [Administrators and IAM](#) roles. A host project administrator must have the proper role to [set up the Shared VPC](#) and make the application project a service project for the host project. See the instructions in [Provisioning Shared VPC](#).

STEP 1 | Enable the service project APIs from the GCP console or the CLI.

The required APIs are:

- ❑ Cloud Deployment Manager API
- ❑ Cloud Pub/Sub API
- ❑ Compute Engine API

Enable APIs from the GCP console

1. Select the service project, and from the Navigation menu, select **APIs & Services**.
2. Search for and view each required API.
3. **ENABLE** any APIs that do not display the “API enabled” status.

Enable APIs from the CLI

1. In the CLI, view your configuration to ensure that you are in the correct project.

```
gcloud config list
```

If not, set the project as follows:

```
gcloud config set project <project-name>
```

2. Issue the following commands to enable the required APIs.

```
gcloud services enable deploymentmanager.googleapis.com
gcloud services enable pubsub.googleapis.com
gcloud services enable compute.googleapis.com
```

3. Confirm that the required APIs are enabled.

```
gcloud services list --enabled
```

STEP 2 | Make the application project a service project for the host project.

Add the service account from Service/application project administrator as a member in host project with following roles:

- Compute Network User
- Pub/Sub Admin

STEP 3 | Choose a VPC configuration.

- If the Service project will share the networks in the host project, continue to [Configure the Shared VPC](#).
- If the Service project has its own VPC network for the application deployment, continue to [Configure a Peered VPC](#).

Configure the Shared VPC

After the firewall template is deployed in the host project, configure the service project that supports your applications. An administrator with shared VPC credentials performs these tasks from the host project. To understand more about the host project and service projects in the context of shared VPC, see the [Shared VPC Overview](#).

STEP 1 | Create a shared VPC using the Trust VPC created when you deployed the firewall template.

Set up a shared VPC for the host (firewall) project:

```
gcloud compute shared-vpc enable HOST_PROJECT_ID
```

STEP 2 | Attach the service/application project to the host project.

```
gcloud compute shared-vpc associated-projects add [SERVICE_PROJECT_ID] --  
host-project [HOST_PROJECT_ID]
```

Additional options are available to share only specific subnets, rather than all subnets in the host project.

STEP 3 | If you want to use the sample application template to deploy an application, continue to [Deploy the Application Template](#).

If you have already deployed an application and you want to secure it in your auto scaling deployment, go to [Manually Onboard an Application to an Existing Auto Scaling Deployment](#).

If you have deployed a service in a GKE cluster, continue to [Onboard a GKE Cluster in a Shared VPC](#).

Configure a Peered VPC

A [VPC network peering](#) connection must be made between two VPCs. If the VPCs are in two different projects, a connection must be created in **both** projects.

STEP 1 | In the host project, peer the Trust VPC network of the Firewall deployment with the Application VPC.

```
gcloud beta compute networks peerings create [PEERING-NAME] \  
--network=[MY-LOCAL-NETWORK] \  
--peer-project [SERVICE-PROJECT-ID] \  
--peer-network [PEER-NETWORK-NAME] \  
[--import-custom-routes] \  
[--export-custom-routes]
```

STEP 2 | In the service project, peer the Trust VPC network of the application deployment with the Trust VPC network of the Firewall deployment.

```
gcloud beta compute networks peerings create [PEERING-NAME] \  
--network=[MY-LOCAL-NETWORK] \  
--peer-project [HOST-PROJECT-ID] \  
--peer-network [PEER-NETWORK-NAME] \  
[--import-custom-routes] \  
[--export-custom-routes]
```

STEP 3 | If you want to use the sample application template to deploy an application, continue to [Deploy the Application Template](#).

If you have already deployed an application and you want to secure it in your auto scaling deployment, go to [Manually Onboard an Application to an Existing Auto Scaling Deployment](#).

If you have deployed a service in a GKE cluster, continue to [Onboard a GKE Cluster in a Peered VPC](#).

Deploy the Application Template

The Service project administrator deploys the [Application Template](#) from the service project.

STEP 1 | Create a separate application project (service project) to deploy the application (see [Prepare a Service Project](#)).

STEP 2 | Prepare the `apps.yaml` file as outlined in [apps.yaml](#).

STEP 3 | Deploy a new application with the application template and define a label for the named port.

```
gcloud deployment-manager deployments create <your-template>
--config apps.yaml
--automatic-rollback-on-error
```

Continue to [View the Onboarded Application in the Panorama Plugin for GCP](#).

Onboard a New Application

When you use the [Application Template](#) to deploy an application, it takes care of the connection to the host project. You can secure applications you did not deploy with the application template, provided they are deployed in a service project with the capabilities described in [Prepare a Service Project](#).

- [Manually Onboard an Application to an Existing Auto Scaling Deployment](#)
- [Onboard a GKE Cluster](#)

Manually Onboard an Application to an Existing Auto Scaling Deployment

To secure an application you have deployed using an external load balancer and an auto-scaled VM-Series firewall deployment, follow these steps. For each application you onboard, you must supply the application name, the named ports, and the path.

STEP 1 | Prepare to add a new named port and URL path to the HTTP external load balancer created when you [deployed the firewall template](#).

STEP 2 | Update all instance groups named-ports with an additional service name and port values. The following sample onboards the applications `app2` and `app3`.

```
gcloud compute instance-groups set-named-ports
fw-template2-fw-igm-us-east1-b
--zone us-east1-b
--named-ports=app1:80,app2:81,app3:82

gcloud compute instance-groups set-named-ports
fw-template2-fw-igm-us-east1-c
--zone us-east1-c
--named-ports=app1:80,app2:81,app3:82
```

STEP 3 | Create a new http-health-check.

```
gcloud compute backend-services create fw-template2-backend-app3
--protocol="HTTP"
--port-name=app3
--http-health-checks=fw-template2-healthcheck-app3
--load-balancing-scheme="EXTERNAL"
--global
```

STEP 4 | Create a new backend service with the port-name created earlier on the HTTP external load balancer.

```
gcloud compute backend-services create fw-template2-backend-app3
--protocol="HTTP" --port-name=app3
--http-health-checks=fw-template2-healthcheck-app3 --load-balancing-
scheme="EXTERNAL"
--global
```

Check to see if the new backend service is visible.

```
gcloud compute backend-services list
```

STEP 5 | Edit url-maps and add new path rule. For example:

```
- paths:
  - /app3
  - /app3/*service:
    https://www.googleapis.com/compute/v1/projects/<project-name>/global/
    backendServices/fw-template2-backend-app3
```

```
gcloud compute url-maps edit fw-template2-ext-loadbalancer
```

STEP 6 | To secure this application with the VM-Series firewall, manually trigger the pub/sub message through the gcloud CLI. This sends a message to the topic created in the firewall template.

```
gcloud pubsub topics publish
projects/topics/hj-asg-891ca3-gcp-pavmqa-panorama-apps-deployment
--attribute ilb-ip=172.22.9.34,
  app-deployment-name=hj-asg-891ca3-app1,
  ilb-port=80,
  named-port=81,
  network-cidr=172.22.9.0/24,
  fw-deployment-name=hj-asg-891ca3,
  host-project=gcp-pavmqa,
  type=ADD-APP
--message "ADD-APP"
```

STEP 7 | [View the Onboarded Application in the Panorama Plugin for GCP.](#)

STEP 8 | (Optional) To update application attributes, such as ilb-ip, ilb-port, or named-port, issue the pubsub command:

```
gcloud pubsub topics publish projects/gcp-pavmqa/topics/hj-asg-891ca3-gcp-
pavmqa-panorama-apps-deployment
--attribute ilb-ip=172.22.9.34,
  app-deployment-name=hj-asg-891ca3-app1,
  ilb-port=80,
  named-port=81,
  network-cidr=172.22.9.0/24,
  fw-deployment-name=hj-asg-891ca3,
  host-project=gcp-pavmqa,
```

```
type=UPDATE-APP
--message "UPDATE-APP"
```

STEP 9 | (Optional) To stop securing the application, issue the following command:

```
gcloud pubsub topics publish projects/gcp-pavmqa/topics/hj-asg-891ca3-gcp-
pavmqa-panorama-apps-deployment
--attribute ilb-ip=172.22.3.20,app-deployment-name=fw-templ-3-app-1,
  ilb-port=80,
  named-port=80,
  fw-deployment-name=hj-asg-891ca3,
  type=DEL-APP
--message "DEL-APP"
```

Onboard a GKE Cluster

To onboard a private GKE cluster, the GCP plugin for Panorama requires the following information.

- In GCP, expose the ELB frontend for the cluster to the GKE service so the VM-Series firewall can get the named port information for the service.
- The cluster API server address.
- The service account credential for the service in which the cluster is deployed, in JSON format.



The GKE cluster name must not exceed 24 characters. This ensures that if you deploy auto scaling in a peered VPC configuration the static route name does not exceed 31 characters.

- [Onboard a GKE Cluster in a Shared VPC](#)
- [Onboard a GKE Cluster in a Peered VPC](#)
- [View the Onboarded Application in the Panorama Plugin for GCP](#)
- [View the Deployment Status from the CLI](#)

Onboard a GKE Cluster in a Shared VPC

To onboard the GKE cluster you must share the Host project Trust network VPC with the Service project. See [Configure the Shared VPC](#).



For security reasons, only private clusters should be used in an auto scaling deployment.
See [Creating a private cluster](#).

STEP 1 | Set the Host project ID.

```
gcloud config set project [HOST_PROJECT_ID]
```

STEP 2 | (optional) Set compute zone or region for clusters.

If the cluster is zonal, enter the following:

```
gcloud config set compute/zone [COMPUTE_ZONE]
```

If the cluster is regional, enter the following:

```
gcloud config set compute/region [COMPUTE_REGION]
```

STEP 3 | In the Host project, update secondary ranges in the Trust VPC subnet.

```
gcloud compute networks subnets update [TRUST_SUBNETWORK_NAME]
--add-secondary-ranges
[PODS_IP_RANGE_NAME] = [POD_RANGE_CIDR] ,
[SERVICE_IP_RANGE_NAME]=[SERVICE_RANGE_CIDR]
```



Pods and service IP ranges must be within: 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16, and cannot collide with existing IP ranges in the subnetwork.

STEP 4 | In the Service project, create a private cluster in the shared VPC.

1. Set the Service project ID.

```
gcloud config set project [SERVICE_PROJECT_ID]
```

2. Create a private cluster in the shared VPC.

```
gcloud container clusters create [CLUSTER_NAME]
--project [SERVICE_PROJECT_ID]
--zone=[ZONE_NAME]
--enable-ip-alias
--enable-private-nodes
--network projects/[HOST_PROJECT_ID]/global/networks/[NETWORK_NAME]
--subnetwork projects/[HOST_PROJECT_ID]/regions/[REGION_NAME]
/subnetworks/[TRUST_SUBNETWORK_NAME]
--cluster-secondary-range-name=[PODS_IP_RANGE_NAME]
--services-secondary-range-name=[SERVICE_IP_RANGE_NAME]
--master-ipv4-cidr=[MASTER_IPV4_CIDR]
--enable-master-authorized-networks
--master-authorized-networks=[PANORAMA_MANAGEMENT_IP/32] ,
[MY_MANAGEMENT_IP/32]
```

STEP 5 | Check your current cluster context:

```
kubectl config current-context
```

STEP 6 | Check all cluster contexts.

```
kubectl config get-context
```

STEP 7 | Change to another cluster.

```
kubectl config use-context [CONTEXT_NAME]
```

If you created your cluster in the GCP console, generate a kubeconfig entry:

```
gcloud container clusters get-credentials [CLUSTER_NAME]
```

STEP 8 | Create a cluster role in a `.yaml` file—for example, `gke_cluster_role.yaml`.

```

apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: gke-plugin-role
rules:
  - apiGroups:
    - ""
    resources:
    - services
    verbs:
    - list

```

STEP 9 | Apply the cluster role.

```
kubectl apply -f gke_cluster_role.yaml
```

STEP 10 | Create a cluster role binding in a `.yaml` file—for example, `gke_cluster_role_binding.yaml`.

```

kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: gke-plugin-role-binding
subjects:
  - kind: ServiceAccount
    name: [SERVICEACCOUNT_NAME]
    namespace: default
roleRef:
  kind: ClusterRole
  name: gke-plugin-role
  apiGroup: rbac.authorization.k8s.io

```

STEP 11 | Apply the cluster role binding.

```
kubectl apply -f gke_cluster_role_binding.yaml
```

STEP 12 | Create a service account.

```
kubectl create serviceaccount [SERVICEACCOUNT_NAME]
```

STEP 13 | Export the service account secret token in JSON format.

```

MY_TOKEN=`kubectl get serviceaccounts [SERVICEACCOUNT_NAME]
-o jsonpath='{.secrets[0].name}'`

kubectl get secret $MY_TOKEN -o json > [FILE_NAME].json

```

STEP 14 | Get the API server address.

```
kubectl config view --minify | grep server | cut -f 2- -d ":" | tr -d " "
```

STEP 15 | In the Panorama plugin for GCP, add the service account information.

Select **Panorama > Google Cloud Platform > Setup**.

Name the credential, enter a description, and enter the **API server address** from step 14, and for **GKE Service Account Credential** upload the JSON file from step 13.

After you add a service account credential, you can validate the credential from your Panorama command line (you cannot validate from the web interface):

```
request plugins gcp validate-service-account gke_service_account <svc-acct-credential-name>
```

STEP 16 | Set up auto scaling on the Panorama plugin for GCP.

1. In the Panorama context, expand Google Cloud Platform, select AutoScaling, and click **Add**.
2. Supply the **Firewall Deployment Name** and an optional description for the deployment.
3. For the **GCP Service Account Credential**, supply the GCP service account name from Step 15.
4. Chose the Device Group and the Template Stack you created in when you [configured the Panorama plugin](#).
5. Disable **License Management Only** to ensure traffic is secured.
6. Enter the exact **GKE Cluster Name**.
7. (Optional) Enter a **Description** of the GKE cluster.
8. Enter the **Network CIDR** for the GKE cluster.
9. Select the **GKE Service Account** corresponding to the GKE cluster.

STEP 17 | Commit your changes.

STEP 18 | (Optional) Create and deploy a service template according to [Using the Sample GKE Service Templates](#), or deploy a GKE service in the GCP console. .

Onboard a GKE Cluster in a Peered VPC

To onboard the GKE cluster you must create and peer the Service VPC with the firewall Trust network in the Host project, as described in [Configure a Peered VPC](#).



For security reasons, only private clusters should be used in an auto scaling deployment.
See [Creating a private cluster](#).

STEP 1 | Set the project ID.

```
gcloud config set project [PROJECT_ID]
```

STEP 2 | Set compute zone or region for clusters.

If the cluster is zonal, enter the following:

```
gcloud config set compute/zone [COMPUTE_ZONE]
```

If the cluster is regional, enter the following:

```
gcloud config set compute/region [COMPUTE_REGION]
```

STEP 3 | Update the service project VPC network with the secondary IP ranges for the pods and services.

```
gcloud compute networks subnets update [GKE_PEERED_VPC_SUBNETWORK]
--region=[REGION]
--add-secondary-ranges PODS_IP_RANGE_NAME=[ip cidr],
SERVICE_IP_RANGE_NAME=[ip cidr]
```

STEP 4 | Enable cloud NAT.



Cloud NAT is required to deploy a private cluster.

```
gcloud compute routers create [ROUTER_NAME]
--network [NETWORK_NAME]
--region [REGION_NAME]
```

```
gcloud compute routers nats create [NAT_CONFIG_NAME]
--router-region [REGION_NAME]
--router [ROUTER_NAME]
--nat-all-subnet-ip-ranges
--auto-allocate-nat-external-ip
```

STEP 5 | Create a new private cluster in the Service VPC.

```
gcloud container clusters create [CLUSTER_NAME]
--project [SERVICE_PROJECT_ID]
--zone=[ZONE_NAME]
--enable-ip-alias
--network [NETWORK_NAME]
--subnetwork [SUBNETWORK_NAME]
--enable-private-nodes
--cluster-secondary-range-name=[PODS_IP_RANGE_NAME]
--services-secondary-range-name=[SERVICE_IP_RANGE_NAME]
--master-ipv4-cidr=[MASTER_IPV4_CIDR]
--enable-master-authorized-networks
--master-authorized-networks=[PANORAMA_MANAGEMENT_IP/32],
[MY_MANAGEMENT_IP/32]
```

STEP 6 | Check your current cluster context:

```
kubectl config current-context
```

STEP 7 | Check all cluster contexts.

```
kubectl config get-context
```

STEP 8 | Change to another cluster.

```
kubectl config use-context [CONTEXT_NAME]
```

If you created your cluster in the GCP console, generate a kubeconfig entry:

```
gcloud container clusters get-credentials [CLUSTER_NAME]
```

STEP 9 | Create a cluster role in a `.yaml` file—for example, `gke_cluster_role.yaml`.

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: gke-plugin-role
rules:
  - apiGroups:
      - ""
    resources:
      - services
    verbs:
      - list
```

STEP 10 | Apply the cluster role.

```
kubectl apply -f gke_cluster_role.yaml
```

STEP 11 | Create a cluster role binding in a `.yaml` file—for example, `gke_cluster_role_binding.yaml`.

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: gke-plugin-role-binding
subjects:
  - kind: ServiceAccount
    name: [SERVICEACCOUNT_NAME]
    namespace: default
roleRef:
  kind: ClusterRole
  name: gke-plugin-role
  apiGroup: rbac.authorization.k8s.io
```

STEP 12 | Apply the cluster role binding.

```
kubectl apply -f gke_cluster_role_binding.yaml
```

STEP 13 | Create a service account.

```
kubectl create serviceaccount [SERVICEACCOUNT_NAME]
```

STEP 14 | Export the service account secret token in JSON format.

```
MY_TOKEN=`kubectl get serviceaccounts [SERVICEACCOUNT_NAME]  
-o jsonpath='{.secrets[0].name}'`
```

```
kubectl get secret $MY_TOKEN -o json >[FILE_NAME].json
```

STEP 15 | Get the API server address.

```
kubectl config view --minify | grep server | cut -f 2- -d ":" | tr -d " "
```

STEP 16 | In the Panorama plugin for GCP, add the service account information.

Select **Panorama > Google Cloud Platform > Setup**.

Name the credential and enter the **API server address** from Step 15, and upload the JSON file you exported in Step 14.

After you add a service account credential, you can validate the credential from your Panorama command line:

```
request plugins gcp validate-service-account <svc-acct-credential-name>
```

STEP 17 | Set up auto scaling on the Panorama plugin for GCP.

1. In the Panorama context, expand Google Cloud Platform, select AutoScaling, and click **Add**.
2. Supply the **Firewall Deployment Name** and an optional description for the deployment.
3. For the **GCP Service Account Credential**, supply the GCP service account name from Step 16.
4. Chose the Device Group and the Template Stack you created in when you [configured the Panorama plugin](#).
5. Disable **License Management Only** to ensure traffic is secured.
6. Enter the exact **GKE Cluster Name**.
7. (Optional) Enter a **Description** of the GKE cluster.
8. Enter the **Network CIDR** for the GKE cluster.
9. Select the **GKE Service Account** corresponding to the GKE cluster.

STEP 18 | (Optional) In your [service project](#), create and deploy a GKE template according to [Using the Sample GKE Service Templates](#), or deploy a GKE service use the GCP console. [Onboard a GKE Cluster](#)

View the Onboarded Application in the Panorama Plugin for GCP

Select **Panorama > Google Cloud Platform > Autoscaling** to view your onboarded application. The **Details** column is only visible if you have an onboarded application.

<input type="checkbox"/> Firewall Deployment Name	Project ID	Device Group	Template Stack	Details
<input type="checkbox"/> gcp-asg-fw-peerbrown0	gcp-pavmqa	GCP_ASG_DG_peerbrown0	GCP_ASG_TS_peerbrown0	Show Status Delicense Inactive VMs Trigger GKE Services Sync
<input type="checkbox"/> hj-nlb-n642wb	gcp-autoscale-host-250622	gcp-autoscale-dg2	gcp-autoscale-ts2	Show Status Delicense Inactive VMs Trigger GKE Services Sync
<input type="checkbox"/> hj-asg-891ca3	gcp-pavmqa	gcp-autoscale-dg-891ca3	gcp-autoscale-ts-891ca3	Show Status Delicense Inactive VMs Trigger GKE Services Sync
<input type="checkbox"/> hj-asg-y892bl	gcp-pavmqa	gcp-autoscale-dg-y892bl	gcp-autoscale-ts-y892bl	Show Status Delicense Inactive VMs Trigger GKE Services Sync

Each link in the Details column triggers an action.

- **Show Status**— [View](#) the details for applications onboarded to a GCP VM-Series firewall deployment.

Show Status Details - hj-asg-891ca3								
Application/GKE Service Name	Host Project	Cluster/Namespace	Named Port	ILB IP	ILB Port	Configuration Programmed	Protected	Not Protected Reason
hj-asg-891ca3-app1	gcp-pavmq	N/A	80	172.22.9.6/32	80	True	True	
web_port1	gcp-pavmq	hj-gke-891ca3-cluster1/ns1	81	172.22.9.11/32	80	True	True	
web2_port2	gcp-pavmq	hj-gke-891ca3-cluster1/ns1	82	172.22.9.12/32	81	True	True	

The following fields display information obtained from the selected deployment. You specified these values in the pub/sub message or through GKE cluster service polling.

- **Application/GKE Service Name**—An application deployment name, or the name of a GKE service.
- **Host Project**—The name of the host project.
- **Cluster/Namespace**—A GKE cluster name followed by the namespace for example, **mycluster/namespace9**.
- **Named Port**—The port assigned to the named port for the service.
- **ILB IP**—The ILB IP address.
- **ILB Port**—The ILB port number.

For autoscaling an application, this property is **ilb-port** in `apps.yaml`.

For securing a GKE cluster, this value is the port number of the GKE cluster, as specified in the `.yaml` file you used to deploy the service in your cluster.

- **Configuration Programmed**— True if a NAT Rule exists, False if not.
- **Protected**— True when an application is onboarded successfully, or False if onboarding failed. If False, see the **Not Protected Reason** column for an explanation.
- **Not Protected Reason**— If **Protected** is False, displays the reason the application is not protected. Some common reasons are:
 - **Configuration Programmed**—True if a NAT Rule exists, False if not.
 - **Protected**—True when an application is onboarded successfully, or False if onboarding failed. If False, see the **Not Protected Reason** column for an explanation.
 - **Not Protected Reason**—If **Protected** is False, displays the reason the application is not protected. Some common reasons are:
 - You deployed a UDP service in the GKE cluster.
 - You specified a named port that is already in use. Only one application can listen on a specific named port.
 - You chose the **License management only** option, so we do not program the configuration.
 - No matching label found for GKE services.

- **Delicense Inactive VMs**—Answer **Yes** to trigger the delicensing function for inactive VMs.
- **Trigger GKE Services Sync**—Answer **Yes** to poll the services running in the clusters, and program the NAT, address, and service objects, and static routes if necessary. By default, Panorama automatically polls 10 minutes after the completion of the previous poll.

View the Deployment Status from the CLI

You can use the Panorama CLI to manage deployed applications. The command line actions parallel those described in [View the Onboarded Application in the Panorama Plugin for GCP](#). In the following commands, the **autoscaling_name** is the Firewall Deployment Name you entered in the auto scaling configuration.

- List the onboarded (protected) applications.

```
show plugins gcp show-protected-apps autoscaling_name <fw-deployment-name>
```

- Trigger the delicensing function for firewalls in the specified deployment.

```
request plugins gcp force-delicensing autoscaling_name <fw-deployment-name>
```

- For a GKE deployment, force the plugin to read the pub-sub messages, and sync NAT rules that are programmed based on the pub-sub messages.

```
request plugins gcp gke-service-discovery autoscaling_name <fw-deployment-name>
```

Parameters in the Auto Scaling Templates for GCP

You can download the template .zip file from <https://github.com/PaloAltoNetworks/GCP-AutoScaling>. The .zip file contains directories to support firewall templates for network load balancer and application load balancer configurations, and the application template.

The template YAML files have the following general format:

```
#Copyright and license information
:
:
imports:                                <do not change>
:
:
resources:
  -name: vm-series-fw                  <do not change>
  -type:vm-series-fw.py               <do not change>
  -properties:
    :
    :
outputs:                                <do not change>
:
:
```

In all .yaml files, you customize the `resources` properties for your deployment. Do not change anything in the `imports` or `outputs` sections.

- [Firewall Templates](#)
- [Application Template](#)

Firewall Templates

The following sections detail the parameters for the NLB and ALB .yaml files.

- [vm-series-fw-nlb.yaml](#)
- [vm-series-fw-alb.yaml](#)

vm-series-fw-nlb.yaml

In the `vm-series-fw-nlb.yaml` template, edit the `-properties`.

Parameter	Sample Value	Comment
region	us-central1	https://cloud.google.com/compute/docs/regions-zones
zones	zones- us-central1-a	If applicable, list multiple zones as follows:

Parameter	Sample Value	Comment
- <list of zones>		zones- us-central1-a- us-central1-b- us-central1-c- us-central1-f
lb-type	nlb	Do not change.
cloud-nat	yes	Do not change.
forwarding-rule-port	80	80 or 8080
urlPath-namedPort-maps-appname	urlPath-namedPort-maps-MyApplication	Enter your application name.
sshkey	'admin:ssh-rsa <PASTE KEY>'	Review SSH Key Pair . In single quotes, type admin: followed by a space, and paste in your key. This is the same convention used for the Google Marketplace template.
bootstrap-bucket	bootstrap-autoscale	The name of the GCP bucket that contains your bootstrap file.
image	vm-series-byol-814	The BYOL image currently available from the Google marketplace. If you are using PAYG or another license model, the image might be different.
machine-type	n1-standard-4	n1-standard-4 is default for BYOL. If your license permits it, you can use any machine type in Minimum System Requirements for the VM-Series Firewall on GCP .
service-account		The unique service account name for the host project.
fw-instance-tag	vm-series-fw	The instance tag you provided in GCP.
metric	custom.googleapis.com/VMSeries/panSessionActive	The custom API path for VM-Series, and your chosen auto scaling metric. Supply only one of the following metrics.

Parameter	Sample Value	Comment
		panSessionActive panSessionUtilization DataPlaneCPUUtilizationPct DataPlanePacketBufferUtilization panSessionUtilization
max-size	2	
min-size	1	
target-type	GAUGE	Currently GAUGE is the only valid type.
util-target	100	

To deploy the VM-Series firewall you need a dedicated network and subnetwork for the firewall's management, untrust, and trust interfaces. Fill out the information for either a greenfield deployment (configure the template to create new networks) or brownfield deployment (use existing networks). Be sure to remove or comment out the network deployment parameters you are not using.

Greenfield Deployment: Enter values to create management, untrust, and trust networks and subnetworks for the firewall.

mgmt-network-cidr	172.22.2.0/24	
untrust-network-cidr	172.22.1.0/24	
trust-network-cidr	172.22.3.0/24	
mgmt-network-access-source-range- <permitted-ip-range>	mgmt-network-access-source-range - <permitted-ip-range-1> - <permitted-ip-range-2>	
mgmt-network-access-ports-<port-number>	mgmt-network-access-ports - 22 - 443	

Brownfield Deployment: Enter the name of each existing network or subnetwork

mgmt-network	my-mgmt-network	
mgmt-subnet	my-mgmt-subnet	
trust-network	my-trust-network	
trust-subnet	my-trust-subnet	
untrust-network	my-untrust-network	

Parameter	Sample Value	Comment
untrust-subnet	my-untrust-subnet	

vm-series-fw-alb.yaml

In the `vm-series-fw-alb.yaml` template, edit the `-properties`.

Parameter	Sample Value	Comment
region	us-central1	https://cloud.google.com/compute/docs/regions-zones
zones - <list of zones>	zones- us-central1-a	If applicable, list multiple zones as follows: zones- us-central1-a- us-central1-b- us-central1-c- us-central1-f
lb-type	alb	Do not change.
cloud-nat	yes	Do not change.
forwarding-rule-port	80	80
connection-draining-timeout	300	The timeout value in seconds.
urlPath-namedPort-maps: - appname: namedPort: urlMapPaths: - '/app1' - '/app1/*'	urlPath-namedPort-maps: - appName: app1 namedPort: 80 urlMapPaths: - '/app1' - '/app1/*' - appName: app2 namedPort: 81 urlMapPaths: - '/app2' - '/app2/*'	List your apps and the corresponding named port
sshkey	'admin:ssh-rsa <PASTE KEY>'	Review SSH Key Pair . In single quotes, type admin : followed by a space, and paste in your key. This is the same convention used for the Google Marketplace template.
bootstrap-bucket	bootstrap-bucket-name	The name of the GCP bucket that contains your bootstrap file.
image	vm-series-byol-814	The BYOL image currently available from the Google marketplace.

Parameter	Sample Value	Comment
		If you are using PAYG or another license model, the image might be different
machine-type	n1-standard-4	n1-standard-4 is default for BYOL. If your license permits it, you can use any machine type in Minimum System Requirements for the VM-Series Firewall on GCP .
service-account	The unique service account name for the service project.	
fw-instance-tag	vm-series-fw	The instance tag you provided in GCP.
metric	custom.googleapis.com/ VMSeries/panSessionActive	The custom API path for VM-Series, and your chosen auto scaling metric. Supply only one of the following metrics. panSessionActive panSessionUtilization DataPlaneCPUUtilizationPct DataPlanePacketBufferUtilization panSessionUtilization
max-size	2	
min-size	1	
target-type	GAUGE	Currently GAUGE is the only valid type.
util-target	100	Enter the goal utilization target value for the auto scaling.

Greenfield Deployment: Enter values to create management, untrust, and trust networks and subnetworks for the firewall.

mgmt-network-cidr	192.168.12.0/24	
untrust-network-cidr	192.168.11.0/24	
trust-network-cidr	192.168.11.0/24	
mgmt-network-access-source-range- <permitted-ip-range>	mgmt-network-access-source-range-1- <permitted-ip-range-2>	
mgmt-network-access-ports- <port-number>	mgmt-network-access-ports- 22- 443	

Parameter	Sample Value	Comment
Brownfield Deployment: Enter the name of each existing network or subnetwork		
mgmt-network	existing-vpc-mgmt	
mgmt-subnet	existing-subnet-mgmt	
trust-network	existing-vpc-trust	
trust-subnet	existing-subnet-trust	
untrust-network	existing-vpc-untrust	
untrust-subnet	existing-subnet-untrust	

Application Template

apps.yaml

The application template creates the connection between the host project (which contains the VM-Series firewalls) and the service project, which contains the application or services that the firewall deployment secures.

Parameter	Sample Value	Comment
host-project	your-host-project-name	The name of the project containing the VM-Series firewall deployment.
fw-deployment-name	my-vm-series-firewall-name	
region	us-central1	https://cloud.google.com/compute/docs/regions-zones
zones - <list of zones>	zones- us-central1-a	If applicable, list multiple zones as follows: zones- us-central1-a- us-central1-b- us-central1-c- us-central1-f
app-machine-type	n1-standard-2	The machine type for the VM running your application or service. If your license permits it, you can use any machine type in Minimum System Requirements for the VM-Series Firewall on GCP .
app-instance-tag	web-app-vm	You applied this tag (label) in GCP.

Parameter	Sample Value	Comment
sshkey	'admin:ssh-rsa <PASTE KEY>'	Review SSH Key Pair . In single quotes, type admin: followed by a space, and paste in your key. This is the same convention used for the Google Marketplace template.
trust-network	<project-name>/<vpc-network-name>	For a shared VPC, the <project-name> is the host project name. For peered VPCs the <project-name> is the Service project name.
trust-subnet	<project-name>/<subnet-name>	For a shared VPC, the <project-name> is the host project name. For peered VPCs the <project-name> is the Service project name.
trust-subnet-cidr	10.2.0.0/24	For a greenfield deployment, the Host project Trust subnet CIDR (the trust-network-cidr parameter in the firewall template). For a brownfield deployment, the CIDR for the Trust network.
vm-series-fw-template-topic	<pubsub-topic>	Enter the topic name created by the firewall deployment. The application template posts a message to the topic to program the firewall configuration to forward traffic.
ilb-port	80	Enter the port number for your application's internal-load-balancer-port. output.
urlPath-namedPort	83	Enter the port number for the urlPath-namedPort output.

Sample GKE Service Templates

These sample templates demonstrate how to configure a GKE service so it is secured by the VM-Series firewall. For the basics on creating your own cluster services, see [Creating a private cluster](#).

- [Using the Sample GKE Service Templates](#)
- [gke_cluster_role.yaml](#)
- [gke_cluster_role_binding.yaml](#)
- [web-deployment.yaml](#)

- [web-service.yaml](#)
- [web-deployment-v2.yaml](#)
- [web-service-v2.yaml](#)
- [Multiple Ports in a Service](#)

Using the Sample GKE Service Templates

You can create a service template based on the sample content in the `.yaml` files that follow. Typically you create a single `.yaml` file.

To be secured by the VM-Series firewall, services in the cluster must be labeled "pavm-named-port=<named_port>" as shown in [web-service.yaml](#) or [web-service-v2.yaml](#).

1. Deploy a `.yaml` file as follows:

```
kubectl apply -f [FILE_NAME].yaml
```

2. Configure the VPC deployment.

- In a shared VPC deployment, launch the GKE cluster in the shared VPC as described in [Configure the Shared VPC](#).
- In a peered VPC deployment, peer the GKE cluster VPC to the host project Trust network. See [Configure a Peered VPC](#).



After a deployment, you can delete all services deployed in the service template `.yaml` file as follows:

```
kubectl delete -f [FILE_NAME].yaml
```

gke_cluster_role.yaml

```
apiVersion: rbac.authorization.k8s.io/v1beta1
kind: ClusterRole
metadata:
  name: gke-plugin-role
rules:
  - apiGroups:
    - ""
    resources:
    - services
    verbs:
    - list
```

gke_cluster_role_binding.yaml

```
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
  name: gke-plugin-role-binding
subjects:
  - kind: ServiceAccount
    name: hj-gke-891ca3-cluster1-sa
    namespace: default
roleRef:
  kind: ClusterRole
```

```
name: gke-plugin-role
apiGroup: rbac.authorization.k8s.io
```

web-deployment.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: web
  namespace: default
spec:
  selector:
    matchLabels:
      run: web
  template:
    metadata:
      labels:
        run: web
    spec:
      containers:
        - image: gcr.io/google-samples/hello-app:1.0
          imagePullPolicy: IfNotPresent
          name: web
          ports:
            - containerPort: 8080
              protocol: TCP
```

web-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: web
  namespace: default
  annotations:
    cloud.google.com/load-balancer-type: "Internal"
  labels:
    pavm-named-port-port1: "80"
spec:
  ports:
    # the port that this service should serve on
    - name: port1
      port: 80
      protocol: TCP
      targetPort: 8080
  selector:
    run: web
  type: LoadBalancer
```

web-deployment-v2.yaml

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: web2
  namespace: default
```

```
spec:
  selector:
    matchLabels:
      run: web2
  template:
    metadata:
      labels:
        run: web2
    spec:
      containers:
        - image: gcr.io/google-samples/hello-app:2.0
          imagePullPolicy: IfNotPresent
          name: web2
          ports:
            - containerPort: 8080
              protocol: TCP
```

web-service-v2.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: web2
  namespace: default
  annotations:
    cloud.google.com/load-balancer-type: "Internal"
  labels:
    pavm-named-port-port2: "81"
spec:
  ports:
    # the port that this service should serve on
    - name: port2
      port: 81
      protocol: TCP
      targetPort: 8080
  selector:
    run: web2
  type: LoadBalancer
```

Multiple Ports in a Service

For multiple ports in one service, edit labels and map the target port name and number in the format `panw-named-port-<service-spec-port-name>`, as shown in the sample below.

```
apiVersion: v1
kind: Service
metadata:
  name: carts
  annotations:
    cloud.google.com/load-balancer-type: "Internal"
  labels:
    panw-named-port-carts-http: "6082"
    panw-named-port-carts-https: "6083"
  namespace: default
spec:
  type: LoadBalancer
  ports:
    # the port that this service should serve on
    - name: carts-http
```

```
protocol: TCP
port: 80
targetPort: 80
- name: carts-https
  protocol: TCP
  port: 443
  targetPort: 443
selector:
  name: carts
```