

## VM-Series for GCP

---



# GCP Terraform Template Deployment Guide

Deploy a two-tiered application environment secured by the VM-Series next generation firewall.

<https://www.paloaltonetworks.com>

# Table of Contents

---

<b>Version History.....</b>	<b>3</b>
<b>1. About Terraform Templates.....</b>	<b>4</b>
<b>2. Support Policy.....</b>	<b>5</b>
<b>3. Instances used .....</b>	<b>5</b>
<b>4. Prerequisites .....</b>	<b>5</b>
4.1    Create GCP account.....	5
4.2    Install the Google Cloud SDK .....	5
4.3    Accept the EULA (If Required).....	6
4.4    Create a Project.....	6
4.5    Enable the API .....	7
4.6    Create a Bootstrap Bucket.....	9
4.7    Download the Template Files.....	14
4.8    Extract the Files .....	14
4.9    Gather Information and Update the Template File.....	14
<b>5. Launch the Template .....</b>	<b>15</b>
<b>6. Review what was created.....</b>	<b>17</b>
<b>7. Access the firewall.....</b>	<b>19</b>
<b>8. Access the Webserver .....</b>	<b>22</b>
<b>9. Launch some attacks .....</b>	<b>25</b>
9.1    SSH from Web Server to DB Server.....	25
9.2    SQL Brute force attack.....	25
<b>10. Cleanup .....</b>	<b>27</b>
10.1    Delete the deployment .....	27
<b>11. Conclusion .....</b>	<b>27</b>
<b>Appendix A .....</b>	<b>27</b>
Troubleshooting tips.....	27

## Version History

Version number	Comments
1.0	Initial Draft

# 1. About Terraform Templates

GCP Terraform Templates, are files that can deploy, configure, and launch GCP resources such as VPC networks & subnets, security groups, firewall rules, route tables, and more. These templates are used for ease of deployment and are key to any cloud deployment model.

For more information on Templates refer to Google's documentation

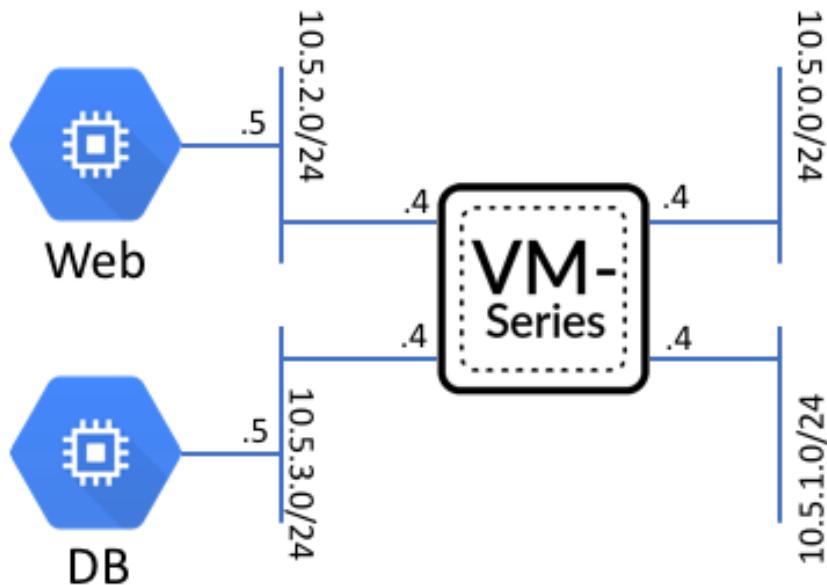
<https://cloud.google.com/community/tutorials/managing-gcp-projects-with-terraform>

There are also many Terraform template s available here:

<https://github.com/GoogleCloudPlatform/terraform-google-examples>

This document will explain how to deploy a Terraform template that launches everything that is shown below in the diagram. This includes, a WordPress server, a MySQL server, a VM-Series firewall and the subnets. In addition, the Terraform template performs a native bootstrapping feature on the VM-Series firewall that allows for additional configuration of the VM-Series firewall (such as routes, security policies, etc.) Once the Terraform template has been deployed, the network topology will align with the following diagram:

**Note: VM-Series must be Licensed to pass traffic.**



## **2. Support Policy**

This template is released under an as-is, best effort, support policy. These scripts should be seen as community supported and Palo Alto Networks will contribute our expertise as and when possible. We do not provide technical support or help in using or troubleshooting the components of the project through our normal support options such as Palo Alto Networks support teams, or ASC (Authorized Support Centers) partners and backline support options. The underlying product used (the VM-Series firewall) by the scripts or templates are still supported, but the support is only for the product functionality and not for help in deploying or using the template or script itself.

Unless explicitly tagged, all projects or work posted in our GitHub repository (at <https://github.com/PaloAltoNetworks/googlecloud>) or sites other than our official Downloads page on <https://support.paloaltonetworks.com> are provided under the best effort policy.

## **3. Instances used**

When using this Terraform template the following machine types are used:

Instance name	Machine Type
WordPress Web Server	f1-micro
WordPress DB Server	f1-micro
VM Series Firewall	n1-standard-4

**Note: There are costs associated with each machine type launched, please refer to the Google instance pricing page <https://cloud.google.com/compute/pricing>**

## **4. Prerequisites**

Here are the prerequisites required to successfully launch this template:

- Terraform installed

### **4.1 Create GCP account**

If you do not have a GCP account already, go to <https://cloud.google.com/free/> and create an account.

### **4.2 Install the Google Cloud SDK**

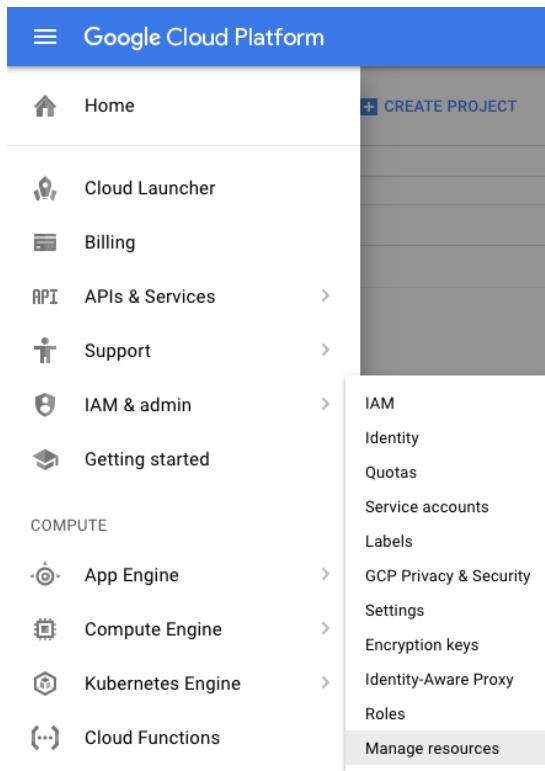
Template installations in GCP are performed from the CLI. Install the SDK/CLI by selecting the relevant platform from the following link and following the installation instructions:

<https://cloud.google.com/sdk/>

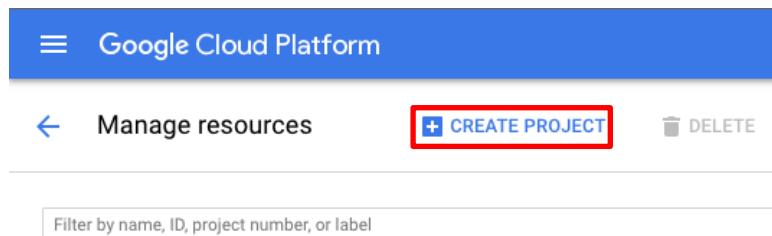
## 4.3 Accept the EULA (If Required)

## 4.4 Create a Project

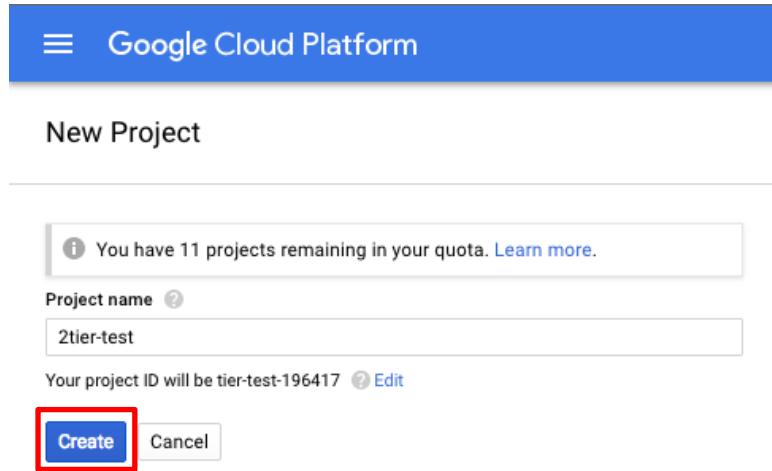
All GCP resources are deployed to a project, which is an organizational boundary that separates users, resources, billing information, etc. It is similar to an AWS VPC or an Azure Resource Group. By default, GCP will create a project upon creation of an account. If that is not the case or to create a dedicated project, use the drop-down on the left and select **IAM & admin > Manage Resources**:



Click **Create Project**:



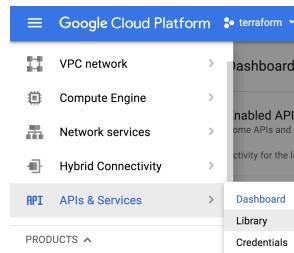
Specify a name for the project and click **Create**:



Note that project creation will take a few minutes.

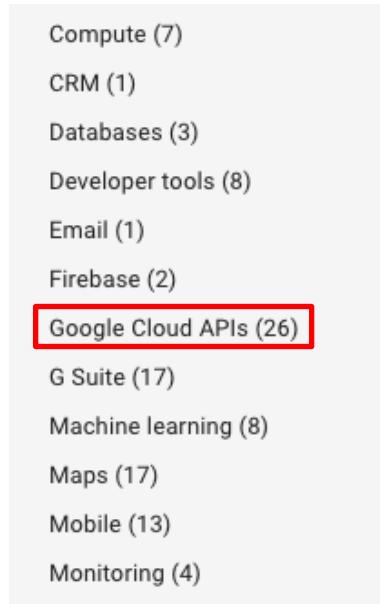
## 4.5 Enable the API

Deploying a template requires the API be enable on the project. Navigate to **APIs & Services > Library**:



Select Google Cloud APIs on the left-hand-side:

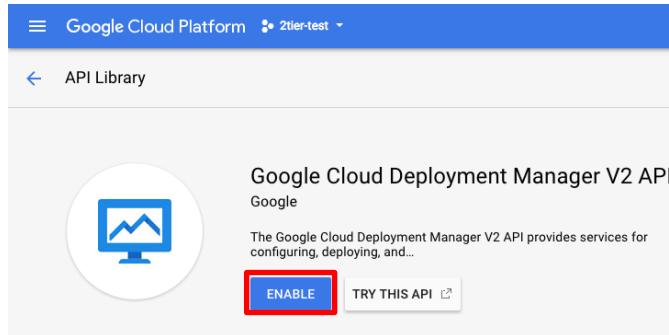
# Palo Alto Networks GCP Terraform Template Deployment Guide



## Select Google Cloud Deployment Manager V2 API:

 BigQuery Data Transfer API Google  Transfers data from partner SaaS applications to Google BigQuery on a scheduled, managed basis.	 Cloud Source Repositories API Google  Access source code repositories hosted by Google.	 Cloud Spanner API Google  Cloud Spanner is a managed, mission-critical, globally consistent and scalable relational
 Google App Engine Flexible Environment Google  This service enables App Engine's Flexible Environment, which gives you the benefits of App...	 Google Cloud Deployment Manager V2 API Google  The Google Cloud Deployment Manager V2 API provides services for configuring, deploying, and...	 Google Cloud Monitoring API Google  This API allows you to monitor resource usage and costs, detect and investigate issues, and
 Google Cloud Runtime Configuration API Google  The Runtime Configurator allows you to dynamically configure and expose variables through Google...	 Google Cloud SQL Google  Google Cloud SQL is a hosted and fully managed relational database service on Google's...	 Google Cloud Storage Google  Google Cloud Storage is a RESTful service for storing and accessing your data on Google's ...

## Select Enable:

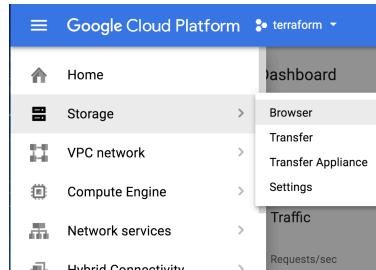


Enabling the API for the project will take a few minutes to complete.

## 4.6 Create a Bootstrap Bucket

Bootstrapping is a feature of the VM-Series firewall that allows you to load a pre-defined configuration into the firewall during boot-up. This ensures that the firewall is configured and ready at initial boot-up, thereby removing the need for manual configuration. The bootstrapping feature also enables automating deployment of the VM-Series.

In order to create a Bootstrap bucket, navigate to **Storage > Browser**:



Click **Create Bucket**:

## Palo Alto Networks GCP Terraform Template Deployment Guide

Cloud Storage  
Buckets

Cloud Storage lets you store unstructured objects in containers called buckets. You can serve static data directly from Cloud Storage, or you can use it to store data for other Google Cloud Platform services.

[Create bucket](#) or [Take the quickstart](#)

Specify a globally-unique bucket name and regional settings and click **Create**:

[←](#) Create a bucket

**Name** ⓘ  
Must be unique across Cloud Storage. Privacy: Do not include sensitive information in your bucket name. Others can discover your bucket name if it matches a name they're trying to use.

**Default storage class** ⓘ  
 **Multi-Regional**  
Use to stream videos and host hot web content.  
Best for data accessed frequently around the world.  
 **Regional**  
Use to store data and run data analytics.  
Best for data accessed frequently in one part of the world.  
 **Nearline**  
Use to store rarely accessed documents.  
Best for data accessed less than once per month.  
 **Coldline**  
Use to store very rarely accessed documents.  
Best for data accessed less than once per year.

**Multi-Regional location**  
Redundant across 2+ regions within your selected location.

[Specify labels](#)

[Create](#) [Cancel](#)

You will need to enter a globally unique bucket name. GCP will warn you if the name is not unique. Once the bucket is created, click on the newly created bucket and add four folders called **config**, **license**, **software** and **content** by clicking on **Create Folder**:

## Palo Alto Networks GCP Terraform Template Deployment Guide

The screenshot shows a cloud storage interface with the following elements:

- Top navigation bar with "Browser", "UPLOAD FILES", "UPLOAD FOLDER", "CREATE FOLDER" (which is highlighted with a red box), "REFRESH", "SHARE PUBLICLY", and "DELETE".
- Search bar with placeholder "Filter by prefix...".
- Breadcrumbs: "Buckets / 2tier-bootstrap".
- Table listing bucket contents:

Name	Size	Type	Storage class	Last modified
config/	—	Folder	—	—
content/	—	Folder	—	—
license/	—	Folder	—	—
software/	—	Folder	—	—

Download the following files and save them in a known location:

<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/bootstrap.xml>

<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/init-cfg.txt>

<https://support.paloaltonetworks.com/Updates/DynamicUpdates/245>

<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/dbserver-startup.sh>

<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/webserver-startup.sh>

Now click on the root folder in the console and click **UPLOAD FILES**:

The screenshot shows a cloud storage interface with the following elements:

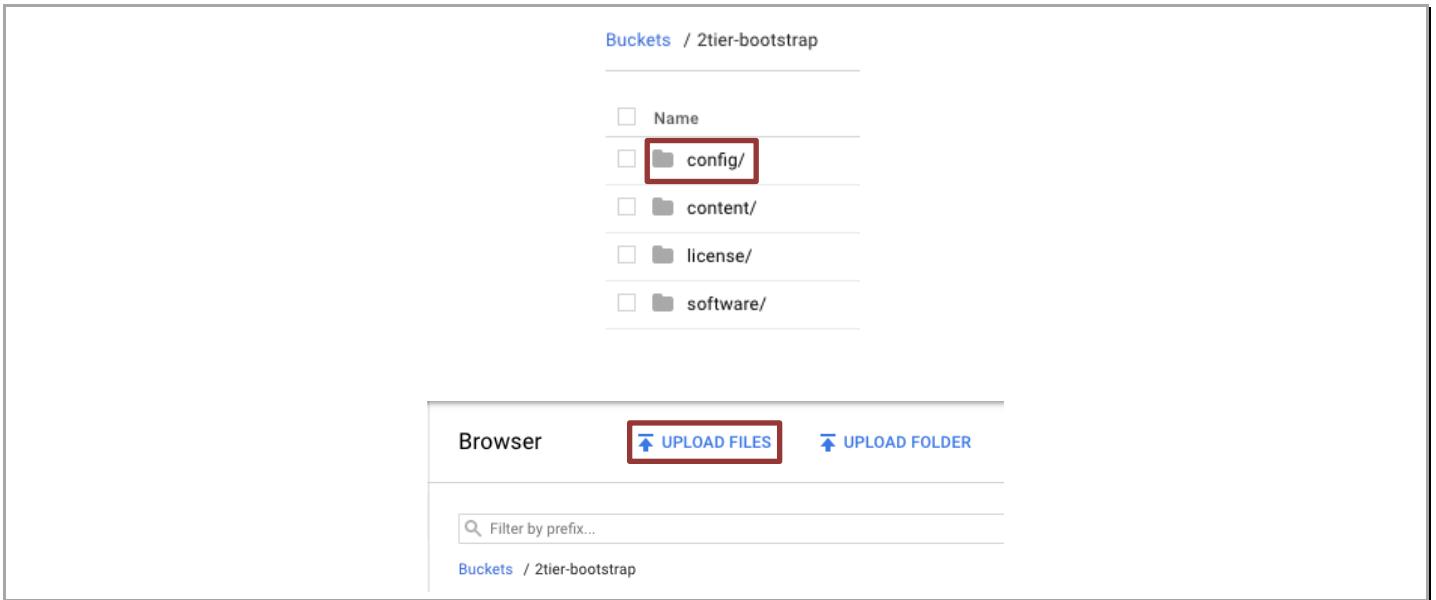
- Top navigation bar with "Buckets / 2tier-bootstrap".
- Table listing bucket contents (same as the first screenshot).
- Bottom navigation bar with "Browser", "UPLOAD FILES" (which is highlighted with a red box), and "UPLOAD FOLDER".
- Search bar with placeholder "Filter by prefix...".
- Breadcrumbs: "Buckets / 2tier-bootstrap".

Select the two files (webserver-startup.sh and dbserver-startup.sh) downloaded previously and put them in the root of your bucket.

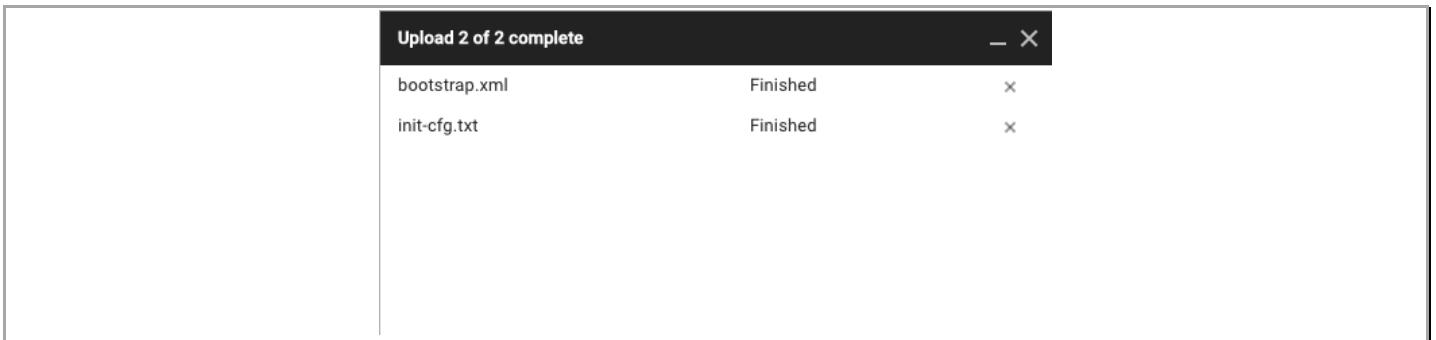
## Palo Alto Networks GCP Terraform Template Deployment Guide



Now click on the **config** folder in the console and click **UPLOAD FILES**:

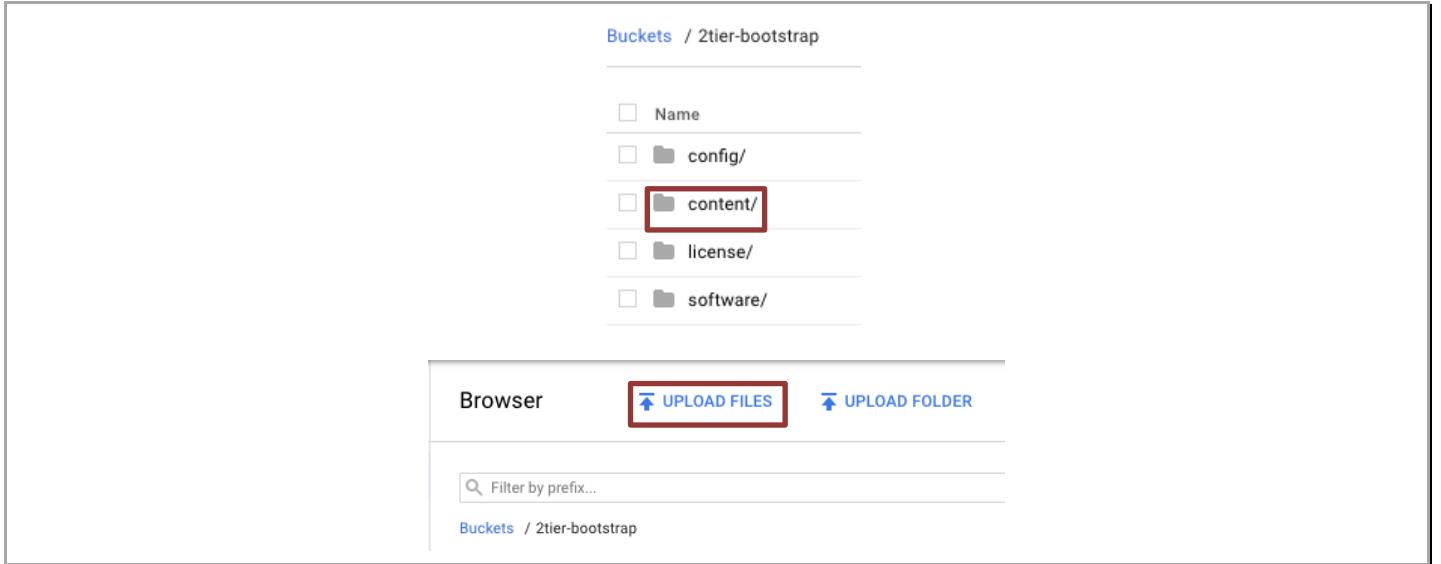


Select the two files (bootstrap.xml and init-cft.txt) downloaded previously and click **Open**:

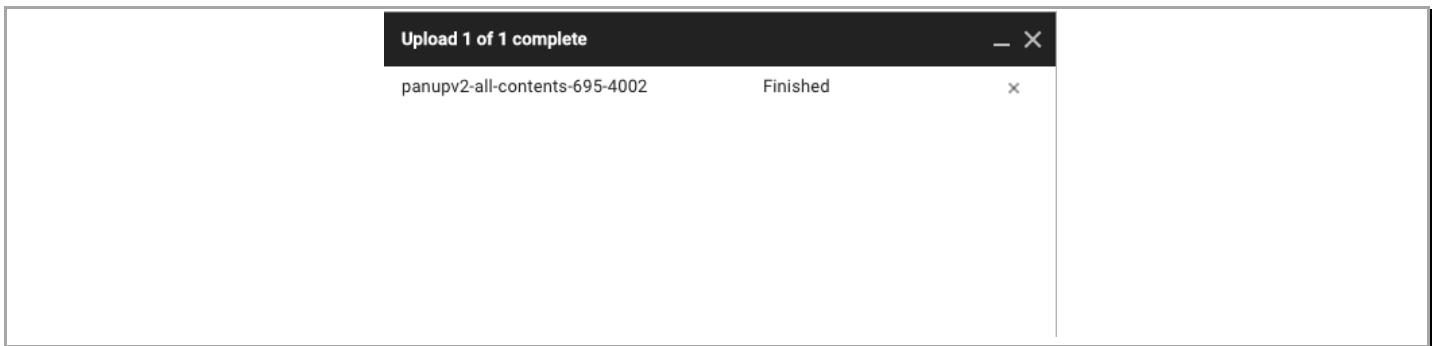


Now click on the **content** folder in the console and click **UPLOAD FILES**:

## Palo Alto Networks GCP Terraform Template Deployment Guide



Select the content file from Dynamic Update Support page (use 797-4614 or higher) **Open**:



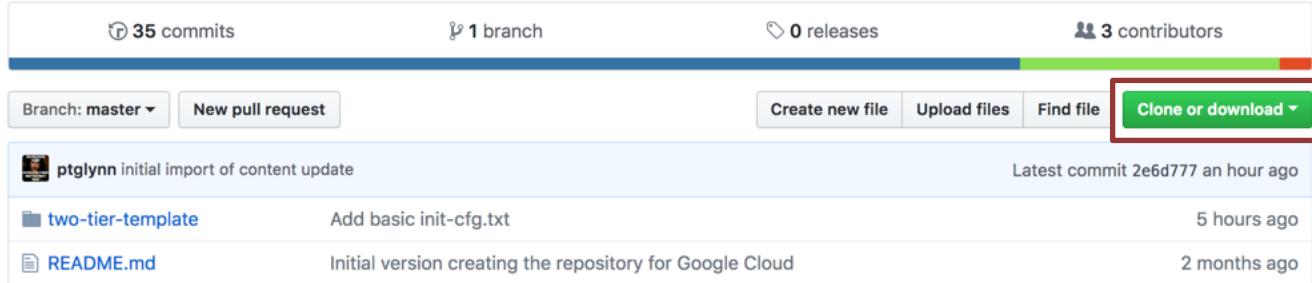
A licenses authcode can be added to the licenses folder or you can manually add licenses to the VM-Series firewall.

Note: This is txt file with the authcode list in the file but there is no extension. Make sure to remove the extension.

**NOTE: Please create the folders using the GCP GUI console or CLI. Creating folders locally on your machine and uploading them may not work as expected. All folders must exist for the VM-Series bootstrap to initiate. However, all folders do not have to have files in them.**

## 4.7 Download the Template Files

Download and save all of the template files to a known location by selecting **Clone or download**:



A screenshot of a GitHub repository page for a 'two-tier-template' repository. The page shows 35 commits, 1 branch, 0 releases, and 3 contributors. A red box highlights the 'Clone or download' button in the top right corner of the main content area. Below the button, there are tabs for 'Create new file', 'Upload files', and 'Find file'. The repository has three files: 'ptglynn initial import of content update' (commit 2e6d777, an hour ago), 'two-tier-template' (Add basic init-cfg.txt, 5 hours ago), and 'README.md' (Initial version creating the repository for Google Cloud, 2 months ago).

## 4.8 Extract the Files

Unzip the template files:

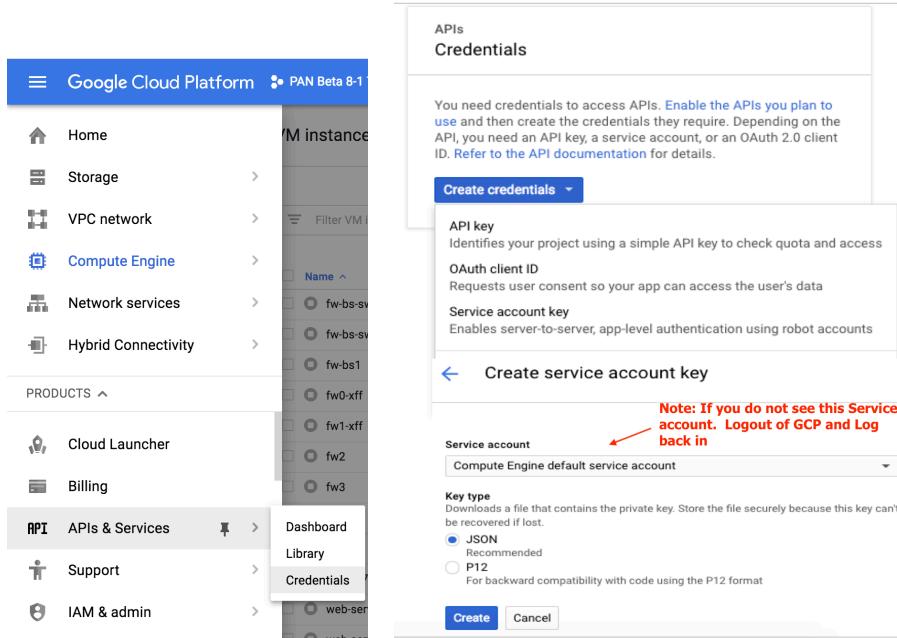
## 4.9 Gather Information and Update the Template File

Deploying the Terraform template in GCP requires modification of the template Main and Variable files to include deployment-specific information. The required information is:

```
credentials = "${file("Your_Project_Credentials.json")}"  
project    = "${Your_Project_ID}"  
region     = "${Your_Project_Region}"  
zone       = "${Your_Project_Zone}"  
sshKey     = "${Your_Public_SSHKey}"
```

## Palo Alto Networks GCP Terraform Template Deployment Guide

To create the credentials to access the APIs in JSON format. In GCP console go to (APIs & Services > Credentials > Create Credentials > Service Account Key), and download the file (client\_secrets.json). Put the .json credential file in your Terraform template folder.



Once the information has been gathered, update the Main and Variable files with the information. Save the Files.

## 5. Launch the Template

Navigate to a command shell navigate to the directory containing the downloaded template files:

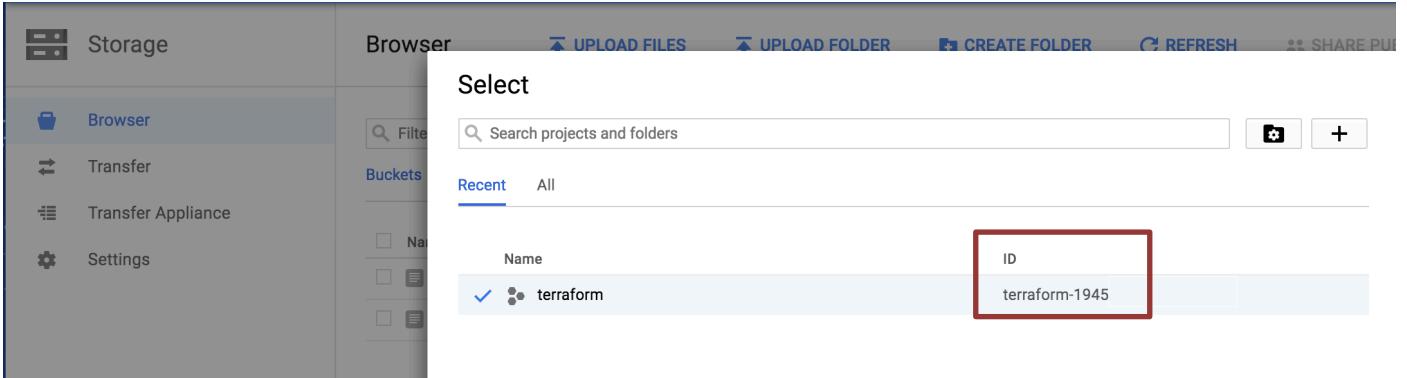
Authenticate to the GCP environment from the command line with the command:

```
$ gcloud auth login
```

- Copy/paste the link into a browser and select the account to authenticate if a browser does not automatically launch:
- Review the requested permissions and click **Allow**:
- Copy the one-time verification code:
- Paste it into the window to complete the authentication request (ignore the warning):

Note the Project ID:

# Palo Alto Networks GCP Terraform Template Deployment Guide



Set the target project for template deployment via command line:

```
$ gcloud config set project my_Project_id
```

Run Terraform Commands:

Initiate template deployment using command “**terraform init**”.

```
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Once the terraform init has completed run the command “**terraform plan**”.

```
Plan: 12 to add, 0 to change, 0 to destroy.

-----
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

You will see if there are any errors and what terraform will be deploying. Now run the “**terraform apply**” command and say **yes** when prompt.

```
Plan: 12 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

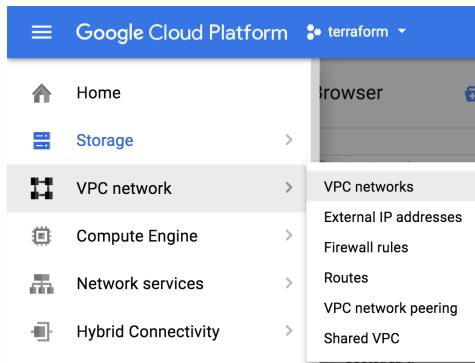
Enter a value: yes
```

If all goes well, Terraform will report success (“Apply Complete!” and no errors):

```
Apply complete! Resources: 12 added, 0 changed, 0 destroyed.
```

## 6. Review what was created

Let's review what the template has launched. The newly created networks can be viewed via **VPC Networks > VPC Network**:



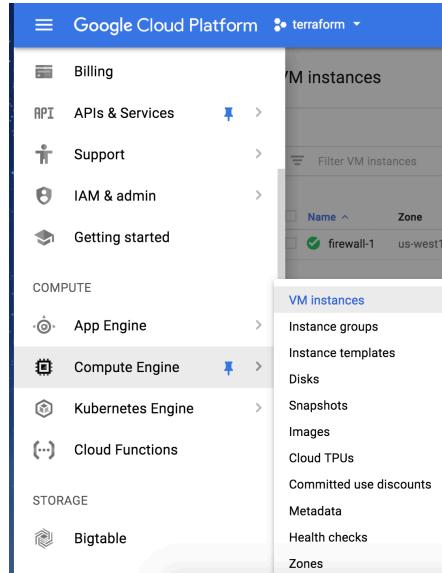
The template creates four networks: db-network, management-network, untrust-network, and web-network:

VPC network		VPC networks							
		CREATE VPC NETWORK		REFRESH					
VPC networks		Name	Region	Subnets	Mode	IP addresses ranges	Gateways	Firewall Rules	Global dynamic routing
	db	1		Custom			1	Off	
		us-west1	db-subnet		10.5.3.0/24		10.5.3.1		
	management	1		Custom			1	Off	
		us-west1	management-subnet		10.5.0.0/24		10.5.0.1		
	untrust	1		Custom			1	Off	
		us-west1	untrust-subnet		10.5.1.0/24		10.5.1.1		
	web	1		Custom			1	Off	
		us-west1	web-subnet		10.5.2.0/24		10.5.2.1		

Note: The default network was automatically created when the project was instantiated. It can be ignored or deleted.

# Palo Alto Networks GCP Terraform Template Deployment Guide

Deployed hosts can be viewed by navigating to **Compute Engine > VM Instances**:



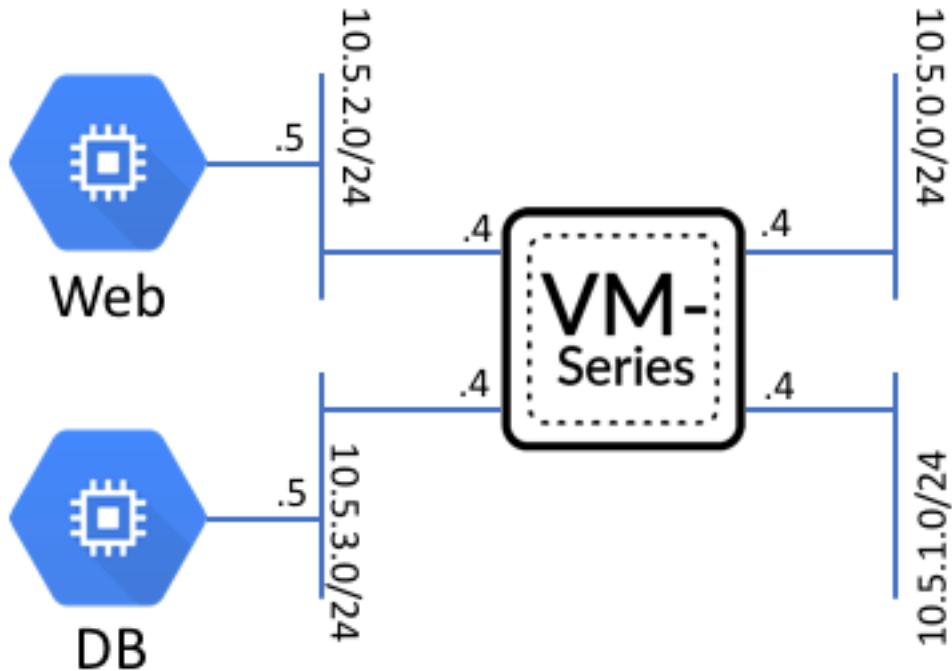
High-level information regarding the deployed instances are available with the default view:

Compute Engine		VM instances	CREATE INSTANCE	IMPORT VM	REFRESH	SHOW
<b>VM Instances</b>						
	Instance groups	Filter VM instances				Columns
	Instance templates					
	Disks					
	Snapshots					
	Images					
	Cloud TPUs					
	Committed use discounts					
	Metadata					
	Health checks					
	Zones					

The table displays three VM instances in the 'VM instances' list:

Name	Zone	Recommendation	Internal IP	External IP	Connect
db-vm	us-west1-a		10.5.3.5	None	SSH
vm-series	us-west1-a		10.5.0.2	35.203.138.227	SSH
webserver	us-west1-a		10.5.2.5	None	SSH

All of this matches the topology shown previously:



## 7. Access the firewall

**NOTE:** Bootstrapping a VM-Series firewall takes approximately 9 minutes. Be patient ☺ Once the template has been deployed successfully, it may be a while before the VM-Series firewall is up and you are able to log into the VM-Series firewall.

Once the template deployment is complete, the VM-Series firewall will show a green checkmark indicating it is running. On the right side is the management IP address of the VM-Series firewall:

# Palo Alto Networks GCP Terraform Template Deployment Guide

VM instances

CREATE INSTANCE IMPORT VM REFRESH

Instance "vm-series" is underutilized. You can save an estimated \$52 per month by switching to the machine type (memory). [Learn more](#)

Filter VM instances

Name	Zone	Recommendation	Internal IP	External IP	Connect
db-vm	us-central1-a		10.5.3.5	None	SSH
vm-series	us-central1-a	Save \$52 / mo	10.5.0.4	35.224.8.98	SSH
web-vm	us-central1-a		10.5.2.5	None	SSH

You should now be able to login to the firewall using the **username: paloalto** and password: **Pal0Alt0@123**

Dashboard ACC Monitor Policies Objects Network Device

Layout: 3 Columns Widgets Last updated: 15:11:51

Commit Config Search

5 mins Help

General Information

Device Name	sample-cft-fw
MGT IP Address	10.5.0.4 (DHCP)
MGT Netmask	255.255.255.255
MGT Default Gateway	10.5.0.1
MGT IPv6 Address	unknown
MGT IPv6 Link Local Address	fe80::4001:aff:fe05:4/64
MGT IPv6 Default Gateway	42:01:0a:05:00:04
Model	PA-VM
Serial #	007200000042435
CPU ID	GCP:D7060200FFFBBB1F
UUID	C33393B-E54A-46B2-F9E5-88BC737EDC87
VM License	VM-300
VM Mode	GCE
Software Version	8.1.0-b8
GlobalProtect Agent	0.0.0
Application Version	695-4002
Threat Version	695-4002
URL Filtering Version	0000.00.00.0000
GlobalProtect Clientless VPN Version	0
Time	Wed Feb 28 13:11:51 2018
Uptime	1 days, 0:43:45

Logged In Admins

Admin	From	Client	Session Start	Idle For
paloalto	12.206.19.5	Web	02/28 12:56:48	00:00:00s

Data Logs

No data available.

System Logs

Description	Time
Failed password for csgoserver from 192.169.155.230 port 58787 ssh2	02/28 12:59:58
failed authentication for user 'csgoserver', Reason: Authentication profile not found for the user. From: ip-192-169-155-230.ip.secureserver.net.	02/28 12:59:42
User paloalto logged in via Web from 12.206.19.5 using https	02/28 12:56:48
authenticated for user 'paloalto'. From: 12.206.19.5.	02/28 12:56:48
Session for user paloalto via Web from 47.183.71.197 timed out	02/28 12:56:37
Session for user paloalto via Web from 10.5.2.5 timed out	02/28 12:56:37
Failed password for user01 from 192.169.155.230 port 37049 ssh2	02/28 12:56:28
failed authentication for user 'user01', Reason: Authentication profile not found for the user. From: ip-192-169-155-230.ip.secureserver.net.	02/28 12:56:20
Failed password for applmgr from 192.169.155.230 port 43818 ssh2	02/28 12:52:58
failed authentication for user 'applmgr', Reason: Authentication profile not found for the user. From: ip-192-169-155-230.ip.secureserver.net.	02/28 12:52:55

Config Logs

No data available.

Locks

No locks found.

ACC Risk Factor (Last 60 minutes)

4.0

Here are the interfaces to zone mappings:

# Palo Alto Networks GCP Terraform Template Deployment Guide

The screenshot shows the Palo Alto Networks Device interface. On the left, there's a navigation tree with categories like Interfaces, Zones, Virtual Routers, IPsec Tunnels, DHCP, DNS Proxy, GlobalProtect (Portals, Gateways, MDM, Device Block List, Clientless Apps, Clientless App Groups), QoS, LLDP, and Network Profiles (GlobalProtect IPsec Crypto, IKE Gateways, IPsec Crypto, IKE Crypto, Monitor, Interface Mgmt, Zone Protection, QoS Profile, LLDP Profile, BFD Profile). The main pane displays a table of network interfaces:

Interface	Interface Type	Management Profile	Link State	IP Address	Virtual Router	Tag	VLAN / Virtual-Wire	Security Zone	Features	Comment
ethernet1/1	Layer3		Up	Dynamic-DHCP Client	default	Untagged	none	Untrust		
ethernet1/2	Layer3	Allow-HTTPS	Up	Dynamic-DHCP Client	default	Untagged	none	Web		
ethernet1/3	Layer3	Allow-HTTPS	Up	Dynamic-DHCP Client	default	Untagged	none	Db		
ethernet1/4			Up	none	none	Untagged	none	none		
ethernet1/5			Up	none	none	Untagged	none	none		
ethernet1/6			Up	none	none	Untagged	none	none		
ethernet1/7			Up	none	none	Untagged	none	none		

At the bottom, there are buttons for Delete, PDF/CSV, Logout, Last Login Time (02/27/2018 12:31:46), Tasks, and Language.

In the policies tab you can review the security policies:

The screenshot shows the Palo Alto Networks Policies interface. On the left, there's a navigation tree with Security, NAT, QoS, Policy Based Forwarding (Decryption, Tunnel Inspection, Application Override, Authentication, DoS Protection), and a Tag Browser (1 item) showing 'none (6) 1-6'. The main pane displays a table of security rules:

Name	Tags	Type	Source				Destination				Rule Usage		
			Zone	Address	User	HIP Profile	Zone	Address	Hit Count	Last Hit	First Hit		
SSH inbound	none	universal	Untrust	any	any	any	Web	any	-	-	-		
SSH 221-222 inbound	none	universal	Untrust	any	any	any	Db	any	0	-	-		
Allow all ping	none	universal	any	any	any	any	any	any	0	-	-		
Web browsing	none	universal	Untrust	any	any	any	Web	any	60	2018-02-28 13:03:26	2018-02-28 13:03:26		
Allow all outbound	none	universal	Db	any	any	any	Untrust	any	35239	2018-02-28 13:12:35	2018-02-28 13:12:35		
Web to DB	none	universal	any	web-object	any	any	any	db-object	0	-	-		
intrazone-default	none	intrazone	any	any	any	any	(intrazone)	any	4	2018-02-27 12:31:51	2018-02-27 12:31:51		
interzone-default	none	interzone	any	any	any	any	any	any	0	-	-		

At the bottom, there are buttons for Add, Delete, Clone, Override, Revert, Enable, Disable, Move, PDF/CSV, Highlight Unused Rules, Reset Rule Hit Counter, Object: Addresses, and a URL field with https://35.224.8.98/#. There are also buttons for Tasks and Language.

## Palo Alto Networks GCP Terraform Template Deployment Guide

These policies are defined to allow ssh access on ports 221 and 222 to the web and db server respectively (for troubleshooting purposes), secures N/S traffic and E/W traffic between zones.

And the NAT policies allow for ssh access to the web and db servers as well as directing web traffic to the web server only. There is also a rule for source NAT from web and db servers to the outside world.

The screenshot shows the Palo Alto Networks Firewall UI. The top navigation bar includes tabs for Dashboard, ACC, Monitor, Policies (which is selected), Objects, Network, and Device. On the right side, there are buttons for Commit, Config, and Search, along with a Help link. The main content area displays a table titled 'Original Packet' and 'Translated Packet' for four rules:

Name	Tags	Source Zone	Destination Zone	Destination Interface	Source Address	Destination Address	Service	Source Translation	Destination Translation
1 Web SSH	none	Untrust	Untrust	any	any	10.5.1.4	service-tcp-2...	dynamic-ip-and-port ethernet1/2	destination-translation address: web-object port: 22
2 DB SSH	none	Untrust	Untrust	any	any	10.5.1.4	service-tcp-2...	dynamic-ip-and-port ethernet1/3	destination-translation address: db-object port: 22
3 WordPress NAT	none	Untrust	Untrust	any	any	10.5.1.4	service-http	dynamic-ip-and-port ethernet1/2	destination-translation address: web-object port: 80
4 Outbound nat	none	any	Untrust	any	any	any	any	dynamic-ip-and-port ethernet1/1	none

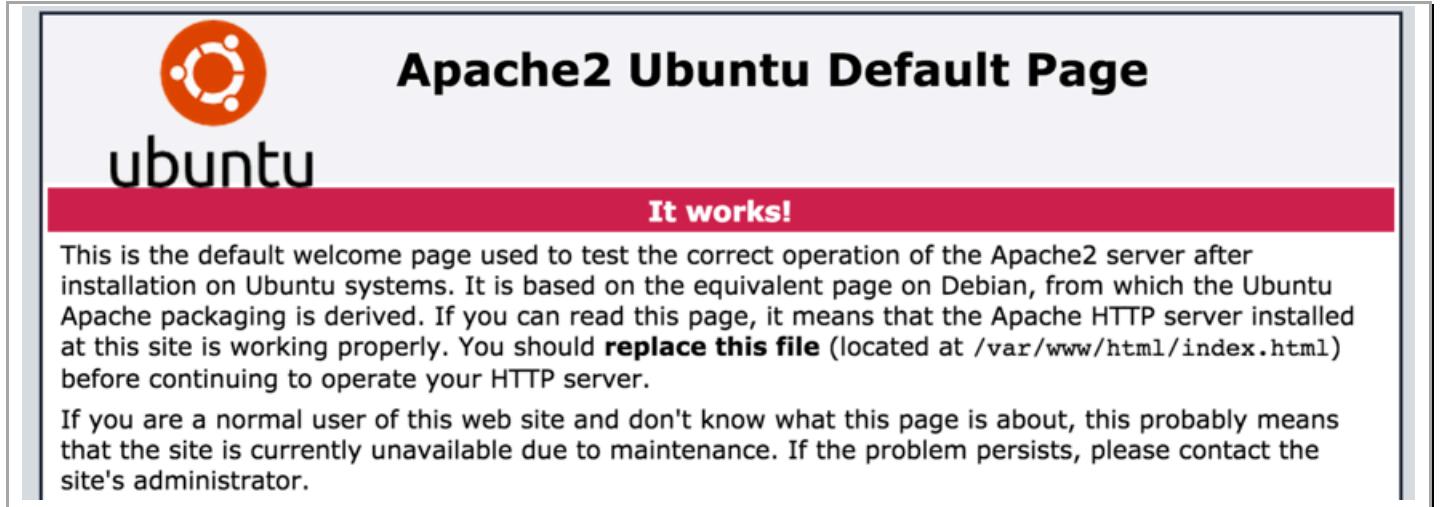
Below the table, a 'Tag Browser' window is open, showing one item: 'Tag(1) Rule 1-4'. It includes filter options: 'Filter by first tag in rule' (checked), 'Rule Order' (radio button), and 'Alphabetical' (radio button). At the bottom of the screen, there are buttons for Object: Addresses, Add, Delete, Clone, Enable, Disable, Move, PDF/CSV, Highlight Unused Rules, and Reset Rule Hit Counter. The footer shows the user is logged in as 'paloalto' and the last login time was '02/27/2018 12:31:46'.

## 8. Access the Webserver

Navigate to the list of VM instances and click on the firewall. Locate the public IP address of the untrust-network interface:

Creation time						
Feb 27, 2018, 2:17:57 PM						
Network interfaces						
Network	Subnetwork	Primary internal IP	Alias IP ranges	External IP	IP forwarding	
mgmt-network	mgmt-subnet	10.5.0.4	—	35.224.8.98 (ephemeral)	On	
untrust-network	untrust-subnet	10.5.1.4	—	35.193.28.231 (ephemeral)		
web-network	web-subnet	10.5.2.4	—	None		
db-network	db-subnet	10.5.3.4	—	None		

Open a new browser tab and type `http://<untrust-network-IP>/` and you should see:

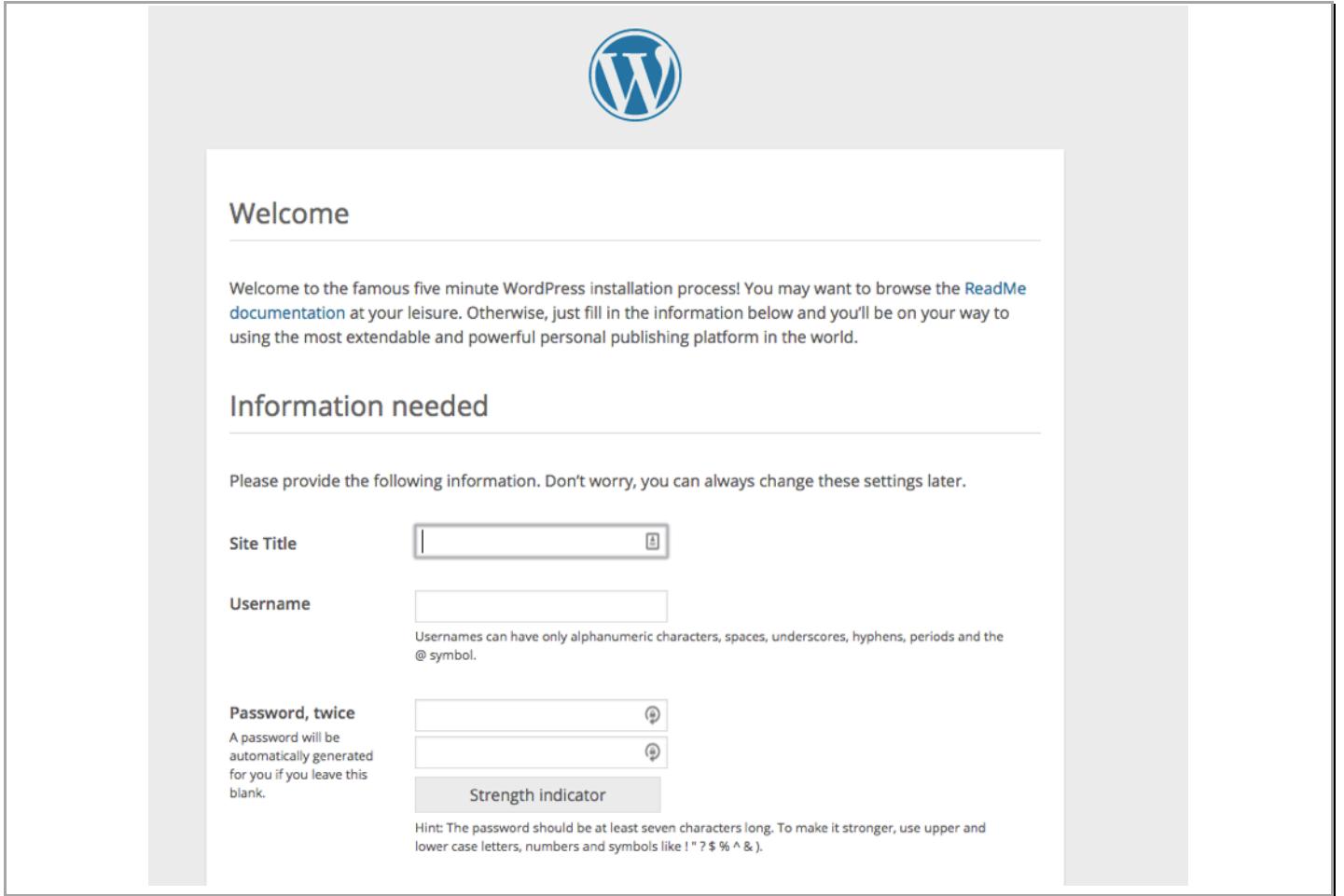


Check firewall logs to verify that the traffic is passing through the firewall:

Manual Help													
( zone.src eq Untrust )													
	Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action	Rule	Session End Reason	Bytes
02/28 15:29:37	start		Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	n/a	913
02/28 15:28:57	start		Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	n/a	795

Now let us verify we pass east-West traffic through the firewall. In the browser, head to the wordpress server (<http://<untrust-network-IP>/wordpress>). You should see the WordPress welcome screen:

## Palo Alto Networks GCP Terraform Template Deployment Guide



**Note: You don't need to actually configure the new WordPress server for the purpose of the test drive. In its initial, un-configured state, it will generate the traffic we need to test the VM-Series firewall.**

Now, head back to the firewall and verify that the traffic did indeed go through the firewall from web to db:

Traffic Log													
	Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action	Rule	Session End Reason	Bytes
🔗	02/28 15:32:03	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	2.2k
🔗	02/28 15:32:03	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	7.3k
🔗	02/28 15:32:03	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	364.3k
🔗	02/28 15:32:02	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	37.5k
🔗	02/28 15:32:02	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	14.9k
🔗	02/28 15:32:02	end	Untrust	Web	12.206.19.5		10.5.1.4	80	web-browsing	allow	Web browsing	tcp-fin	5.1k
🔗	02/28 15:31:04	end	Web	Db	10.5.2.5		10.5.3.5	3306	mysql	allow	Web to DB	tcp-fin	27.6k
🔗	02/28 15:31:03	end	Web	Db	10.5.2.5		10.5.3.5	3306	mysql	allow	Web to DB	tcp-fin	4.5k

You have now successfully deployed a GCP Terraform template with a VM-Series firewall in GCP.

## 9. Launch some attacks

### 9.1 SSH from Web Server to DB Server

Let's simulate a compromised web server that is being used to attack the database. This is a common attack strategy of getting a foothold on the web front-end server and then expanding to the other application tiers with the ultimate goal of accessing all data in the database.

Go to <http://<untrust-network-IP>/sql-attack.html> and simulate a web to db ssh attempt by clicking on the **LAUNCH WEB TO DB SSH ATTEMPT**.

**LAUNCH WEB TO DB SSH ATTEMPT**

This launches a CGI script that attempts to ssh as root to the db server from the web server. Now return to the firewall's monitor tab to note the failed traffic:

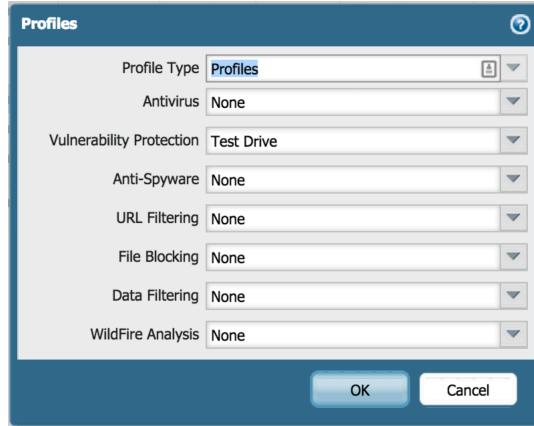
	Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action	Rule	Session End Reason	Bytes
02/28 15:33:51	drop		Web	Db	10.5.2.5		10.5.3.5	22	not-applicable	deny	interzone-default	policy-deny	74
02/28 15:33:50	drop		Web	Db	10.5.2.5		10.5.3.5	22	not-applicable	deny	interzone-default	policy-deny	74

### 9.2 SQL Brute force attack

On the firewall's security policies tab, under Security, Rule 6, you will notice that the web to db traffic is protected further by a vulnerability profile:

Tags	Type	Source				Destination				Rule Usage				Application	Service	Action	Profile	Options
		Zone	Address	User	HIP Profile	Zone	Address	Hit Count	Last Hit	First Hit								
none	universal	Untrust	any	any	any	Web	any	-	-	-	ping	ssh	application-default	Allow	none			
none	universal	Untrust	any	any	any	Db	any	0	-	-	ping	ssh	service-tcp-221	Allow	none			
none	universal	any	any	any	any	Web	any	1	2018-02-28 15:28:21	2018-02-28 15:28:21	ping	ssh	service-tcp-222	Allow	none			
none	universal	Untrust	any	any	any	Web	any	20	2018-02-28 15:33:51	2018-02-28 15:29:37	ping	application-default	service-http	Allow	none			
none	universal	Db	any	any	any	Untrust	any	87	2018-02-28 15:28:21	2018-02-28 15:28:21	any	mysql	application-default	Allow	none			
<hr/>																		
none	universal	any	web-object	any	any	any	db-object	8	2018-02-28 15:30:49	2018-02-28 15:28:21	mysql	application-default	allow	block				
none	intrazone	any	any	any	any	(intrazone)	any	4	2018-02-28 15:27:42	2018-02-28 15:27:42	any	any	any	Allow	none	none		
none	interzone	any	any	any	any	any	any	2	2018-02-28 15:33:52	2018-02-28 15:33:52	any	any	any	Deny	none			

Now click on the icon in the Profile column and you will see all the threat protection profiles



Note the Vulnerability Protection profile. This is a custom profile created just for this lab. It is part of the default vulnerability protection profile but is called out separately for the purpose of this demo environment.

Let's finally trigger the attack. Head back to the sql-attack.html page at <http://<untrust-network-IP>/sql-attack.html>

Click on Launch Brute Force Attack to start a script that will generate multiple failed MySQL authentication attempts.

### LAUNCH BRUTE FORCE SQL ROOT PASSWORD GUESSING

This will launch some scripted attacks on the SQL server and use the pre-configured threat protection to show and block those attacks on the VM-Series firewall. Now return to the firewall and click the Monitor tab and then click on Threats in the left-hand pane under Logs and notice the new vulnerability log message regarding the failed MySQL events:

Logs	Traffic	Threat	URL Filtering	WildFire Submissions	Receive Time	Type	Name	From Zone	To Zone	Source address	Source User	Destination address	To Port	Application	Action	Severity
					02/28 15:37:48	vulnerability	MySQL Login Authentication Failed	Web	Db	10.5.2.5		10.5.3.5	3306	mysql	reset-client	informational

The CGI script you launched above attempted to login to the MySQL database multiple times with an incorrect password. The VM-Series firewall saw this activity and using the vulnerability profile, reset the connection and logged the activity.

# 10. Cleanup

## 10.1 Delete the deployment

Once done, cleanup as follows:

- If you licensed the VM-Series firewall perform the De-License function.
  - [https://www.paloaltonetworks.com/documentation/71/virtualization/virtualization/license-the-vm-series-firewall/deactivate-vm#\\_87329](https://www.paloaltonetworks.com/documentation/71/virtualization/virtualization/license-the-vm-series-firewall/deactivate-vm#_87329)
- From the CLI, issue the command “**terraform destroy**”
  - This will delete all the resources created via the Terraform template.

# 11. Conclusion

You have successfully deployed a Terraform template in GCP and demonstrated how the Palo Alto Next Generation VM-Series firewall can be deployed via Terraform automation to not only secure traffic throughout your GCP Project, but throughout your Enterprise Google Cloud Infrastructure.

# Appendix A

## Troubleshooting tips

### 1. Unable to access the webserver or web page not visible

If the VM-Series firewall is up and accessible but you are unable to access the webserver (or the web page is not visible), then chances are that the startup scripts did not get downloaded from the bootstrap bucket or were corrupted during (or prior to) the upload. Ensure that the files webserver-startup.sh and dbserver-startup.sh are in the bootstrap bucket. If they are extant, replace them with new copies downloaded from the GitHub repository.

### 2. Bootstrapping not working

If the VM-Series firewall is up and you are able to access the login page, but unable to login using the username/password: paloalto/Pal0Alt0@123, then chances are bootstrapping has failed.

There could be several reasons:

#### *a. Corrupt configuration files*

Please ensure that the bootstrap.xml and init-cft.txt files mentioned in [Section 4.6](#) are not corrupted.

*b. Incorrect bootstrap bucket-name*

Another reason for bootstrapping to fail is that the bootstrap bucket name (Parameter: bootstrapbucket) was incorrectly entered in the template file. Please make sure the bucket name created in [Section 4.6](#) is mentioned when launching the template.