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

Last Update: May 2016

Table of Contents

Version History **3**
1. About Terraform Templates **4**
2. Support Policy **5**
3. Instances used **5**
4. Prerequisites **5**
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
5. Launch the Template **15**
6. Review what was created **17**
7. Access the firewall **19**
8. Access the Webserver **22**
9. Launch some attacks **25**
9.1 SSH from Web Server to DB Server 25
9.2 SQL Brute force attack 25
10. Cleanup **27**
10.1 Delete the deployment 27
11. Conclusion **27**
Appendix A **27**
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

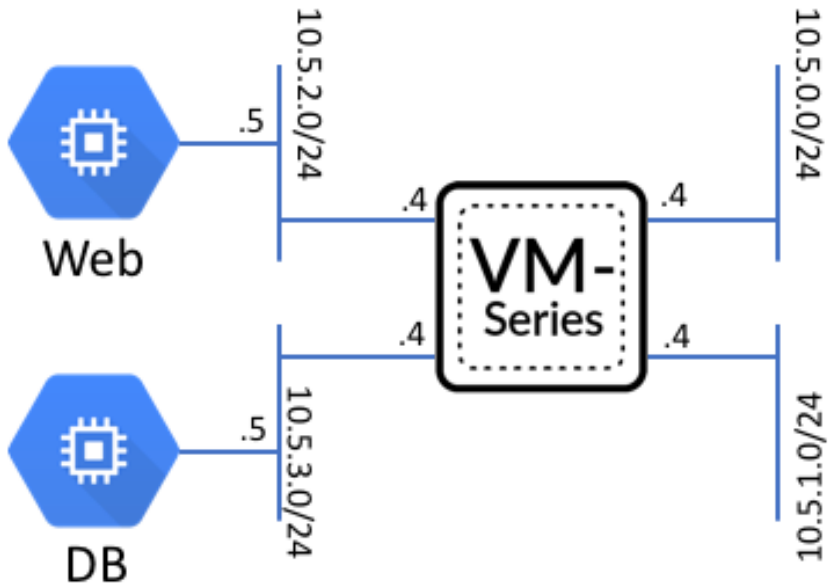
https://cloud.google.com/community/tutorials/managing-gcp-projects-with-terraform

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

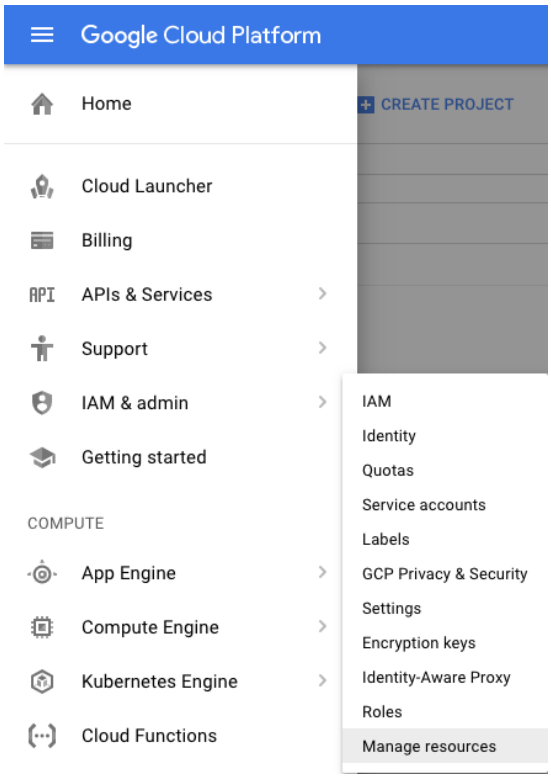
Template installations in GCP are performed from the CLI. Install the SDK/CLI by selecting the relevant platform from the following link and following the installation instructions:

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

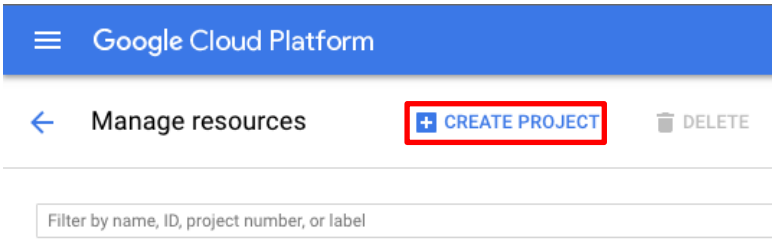
4.3 Accept the EULA (If Required)

4.4 Create a Project

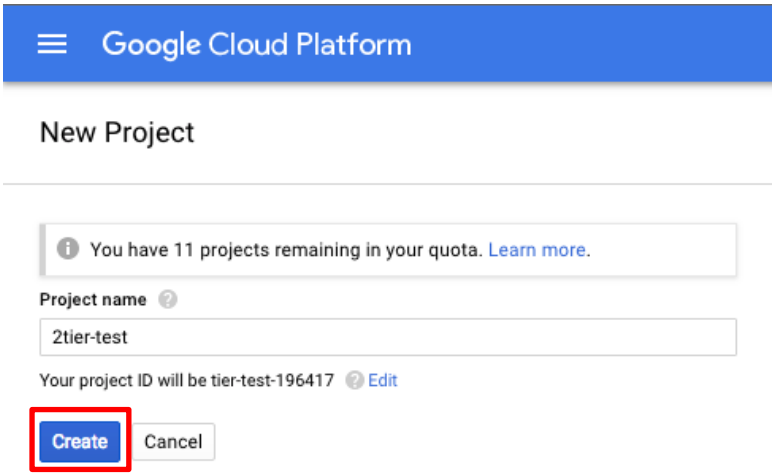
All GCP resources are deployed to a project, which is an organizational boundary that separates users, resources, billing information, etc. It is similar to an AWS VPC or an Azure Resource Group. By default, GCP will create a project upon creation of an account. If that is not the case or to create a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



Click **Create Project**:



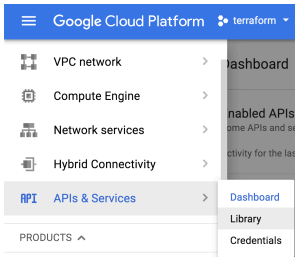
Specify a name for the project and click **Create**:



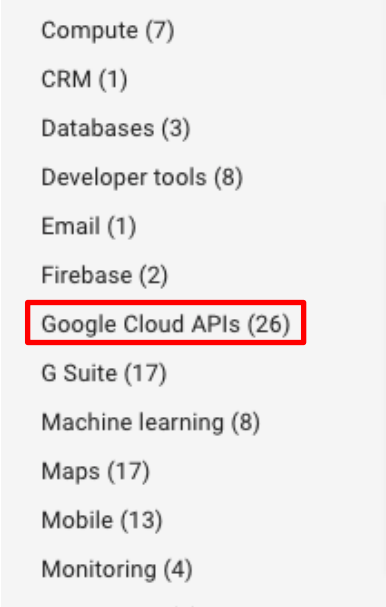
Note that project creation will take a few minutes.

4.5 Enable the API

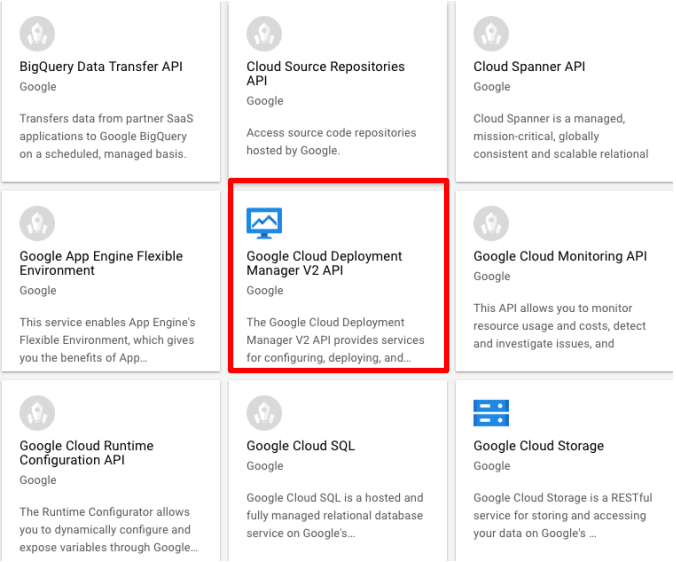
Deploying a template requires the API be enable on the project. Navigate to **APIs & Services > Library**:



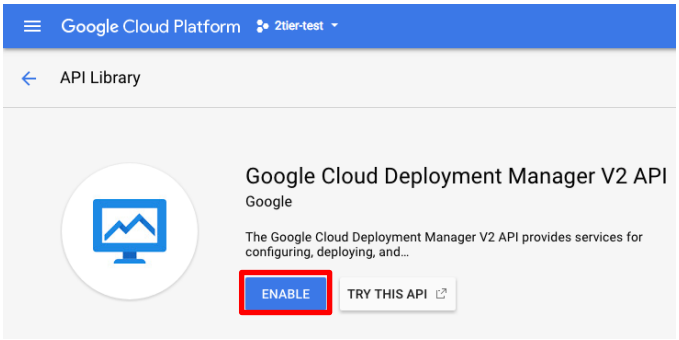
Select Google Cloud APIs on the left-hand-side:



Select Google **Cloud Deployment Manager V2 API**:



Select **Enable**:

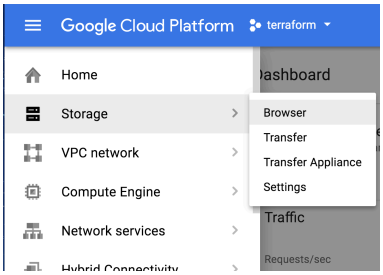


Enabling the API for the project will take a few minutes to complete.

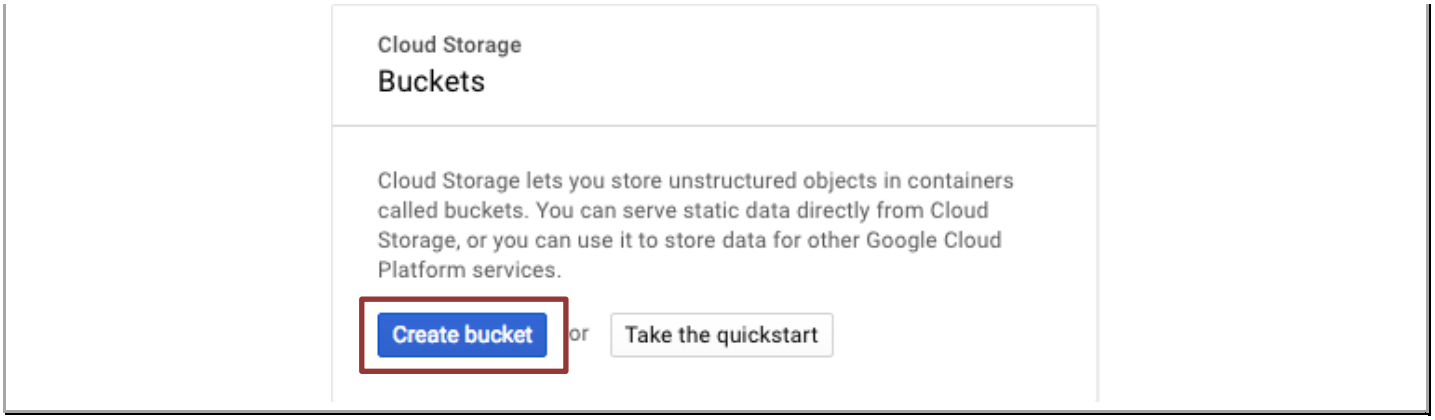
4.6 Create a Bootstrap Bucket

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.

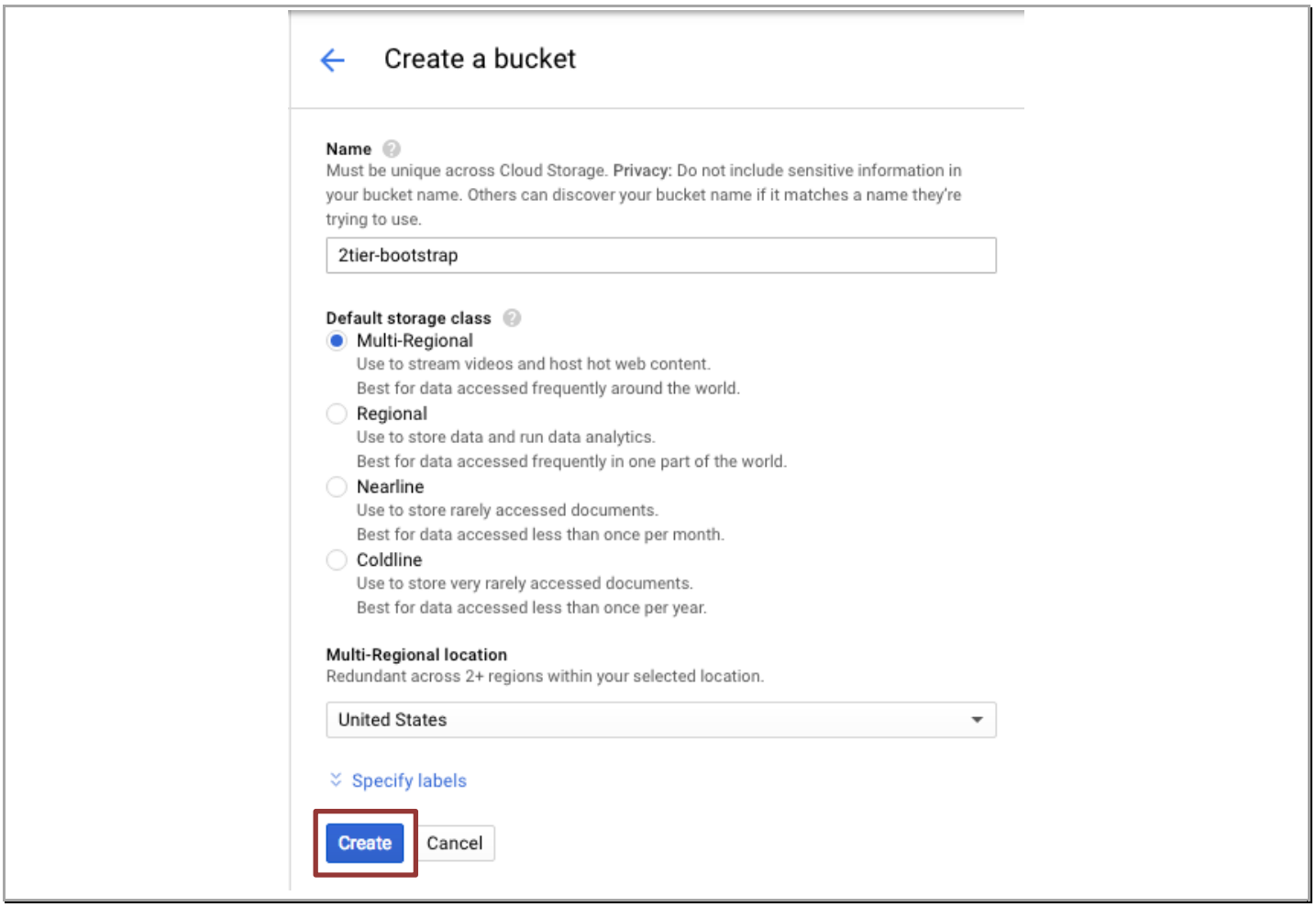
In order to create a Bootstrap bucket, navigate to **Storage > Browser**:



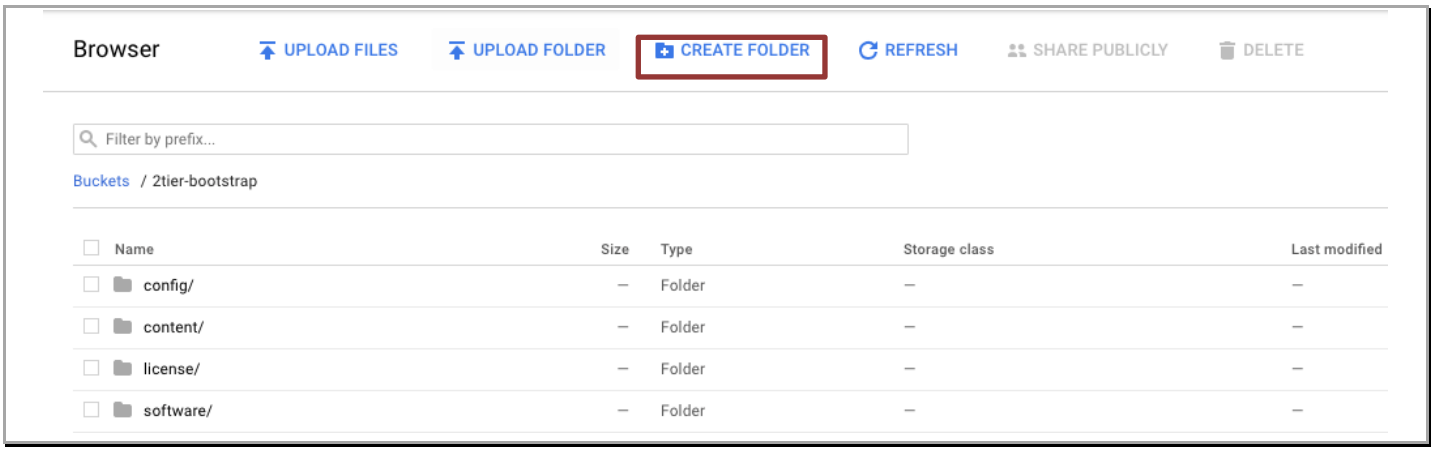
Click **Create Bucket**:



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



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**:



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

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

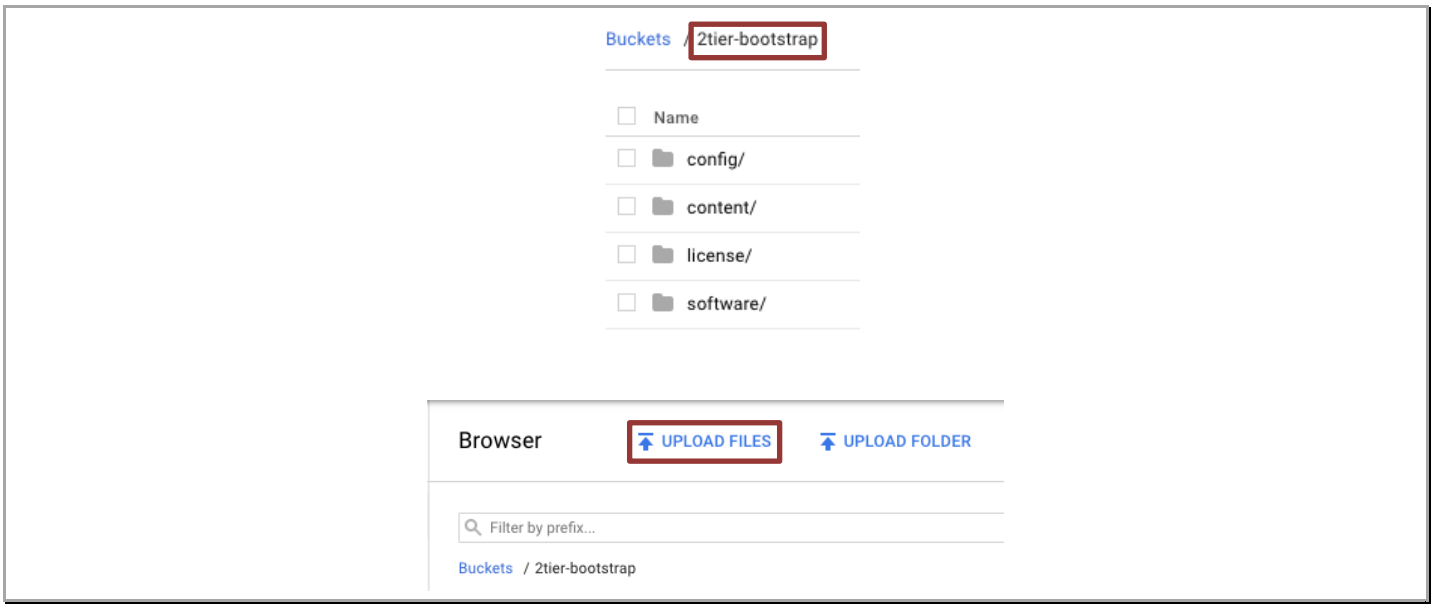
https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/init-cfg.txt

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

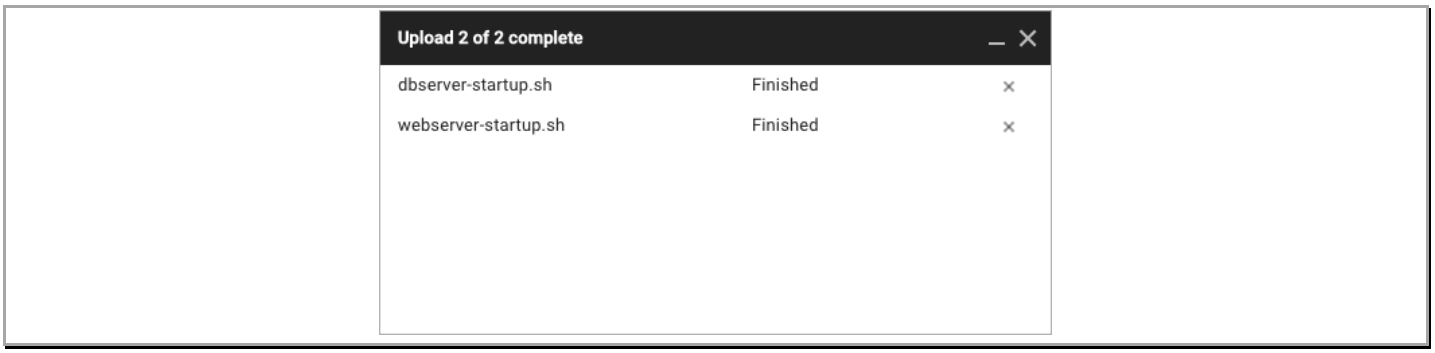
https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/dbserver-startup.sh

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

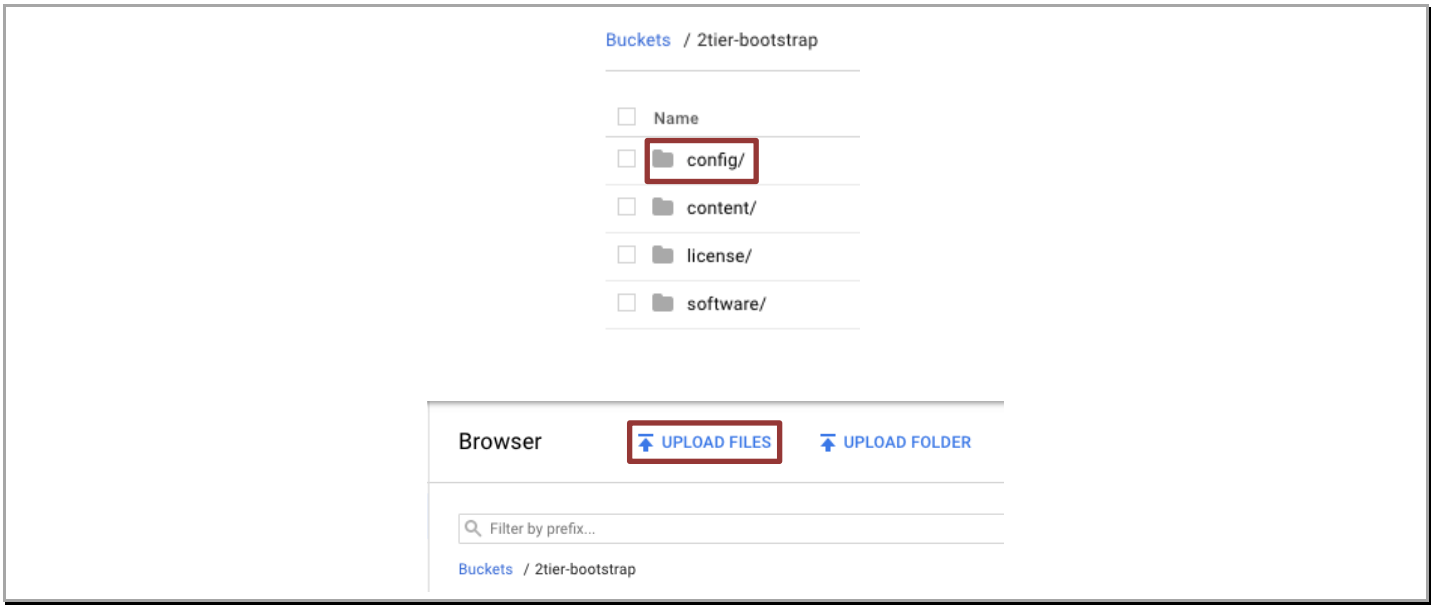
Now click on the root folder in the console and click **UPLOAD FILES:**



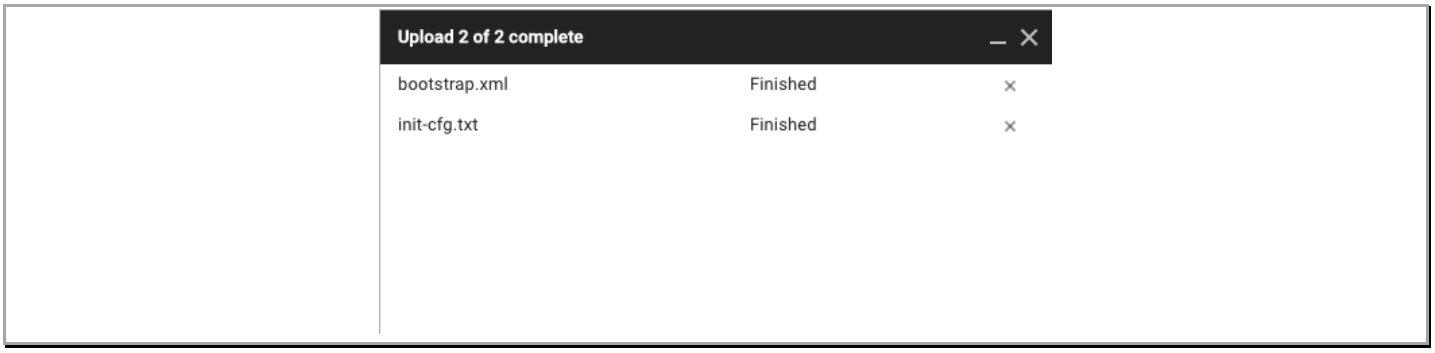
Select the two files (webserver-startup.sh and dbserver-startup.sh) downloaded previously and put them in the root of your bucket.



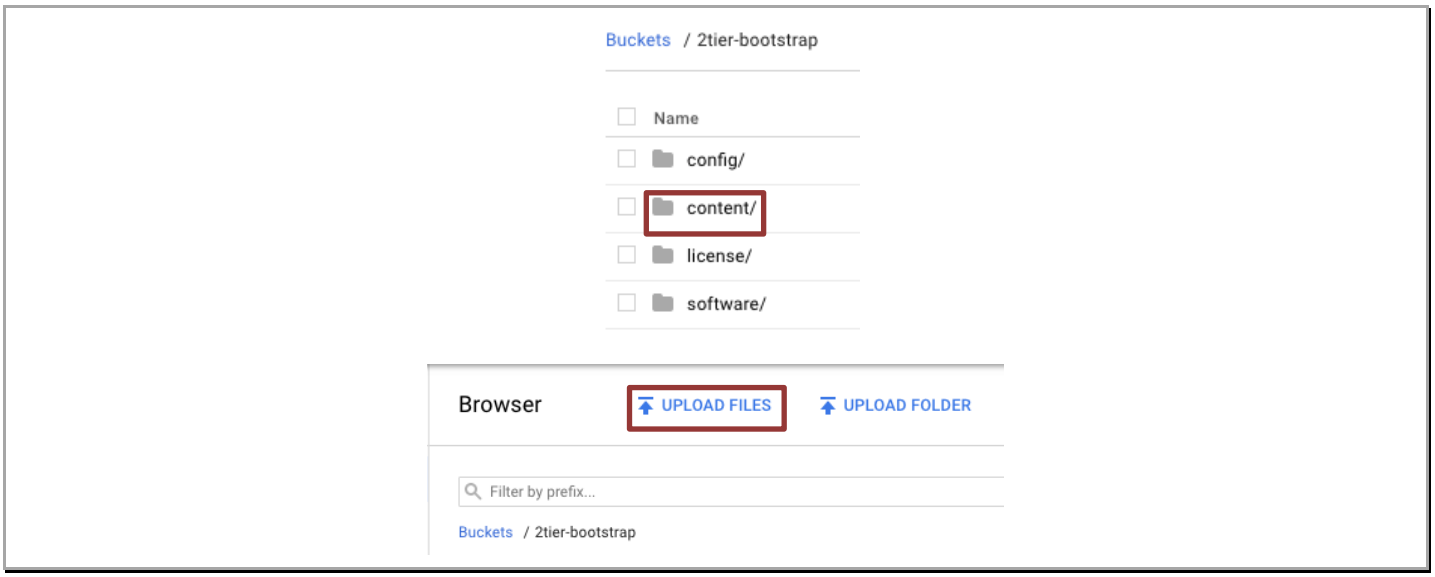
Now click on the **config** folder in the console and click **UPLOAD FILES:**



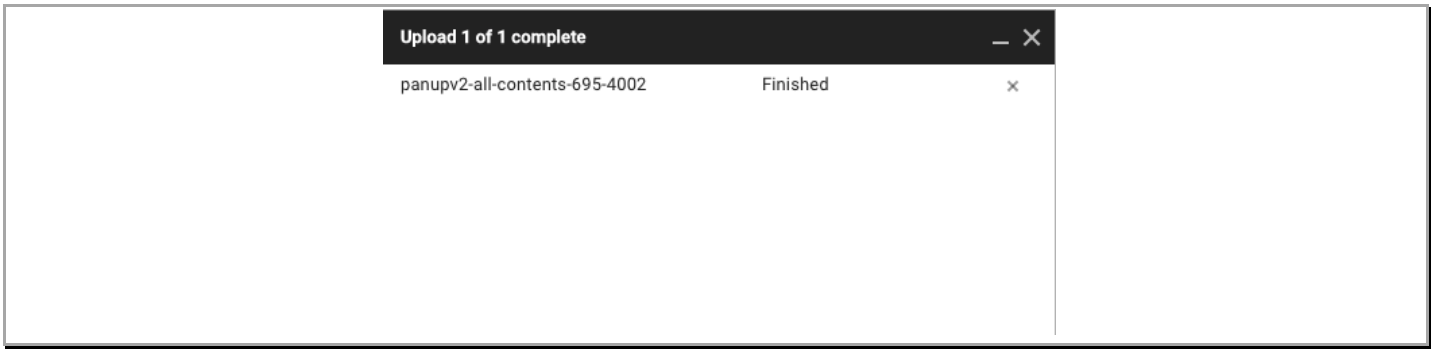
Select the two files (bootstrap.xml and init-cft.txt) downloaded previously and click **Open**:



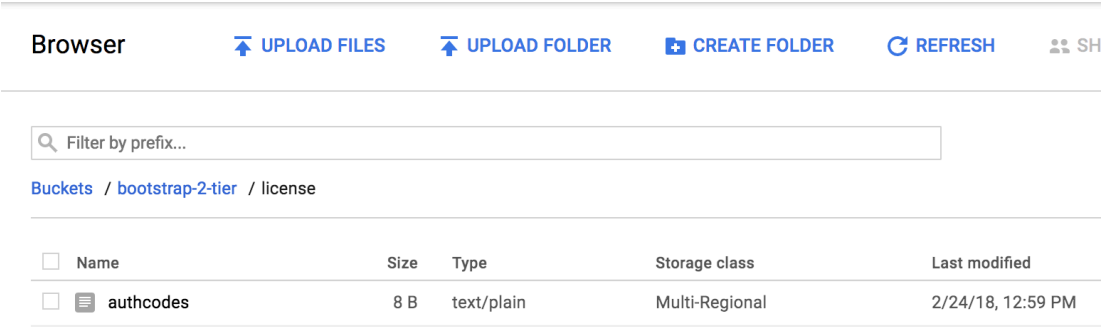
Now click on the **content** folder in the console and click **UPLOAD FILES:**



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.

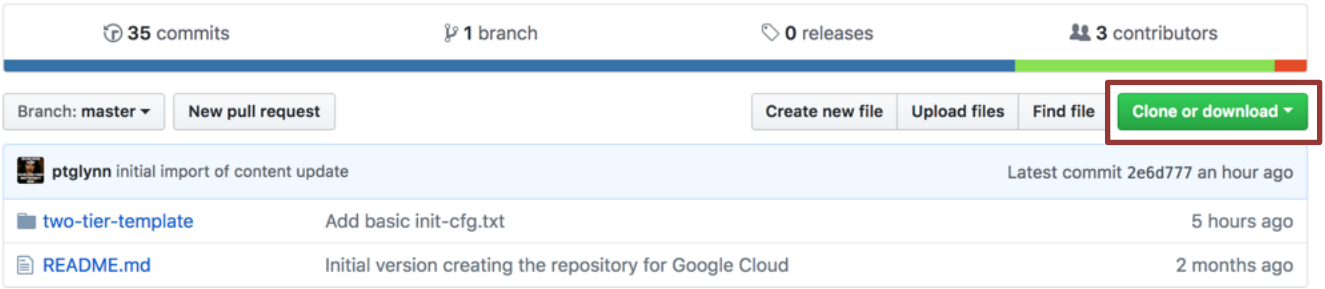


Note: This is txt file with the authcode list in the file but there is no extension. Make sure to remove the extension.

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**:



4.8 Extract the Files

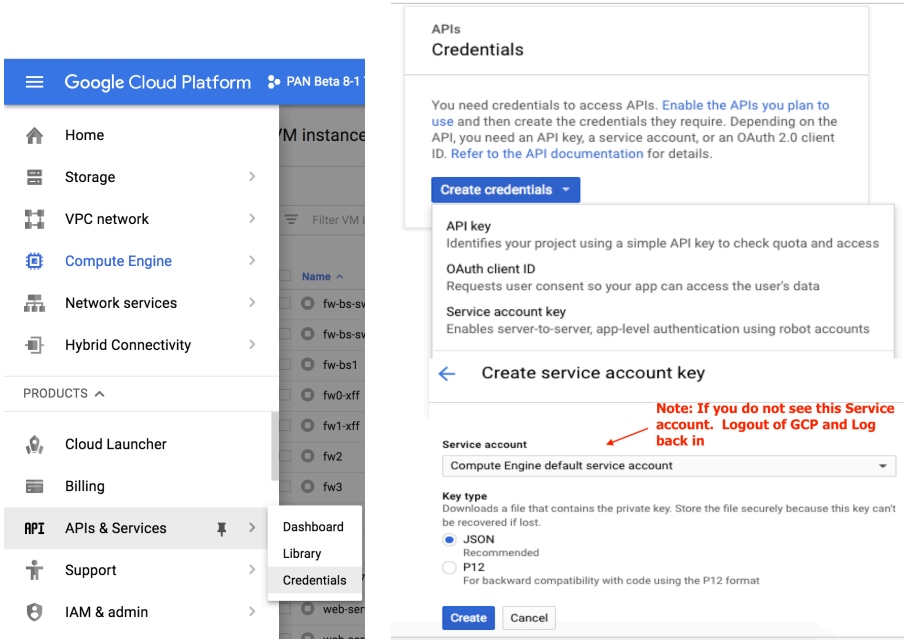
Unzip the template files:

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}"
```

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

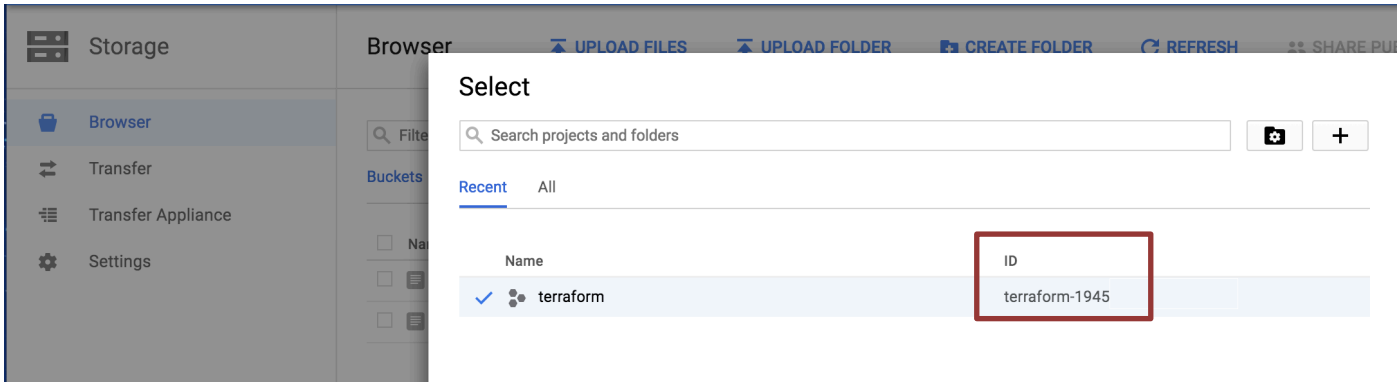
Navigate to a command shell navigate to the directory containing the downloaded template files:

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:



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.

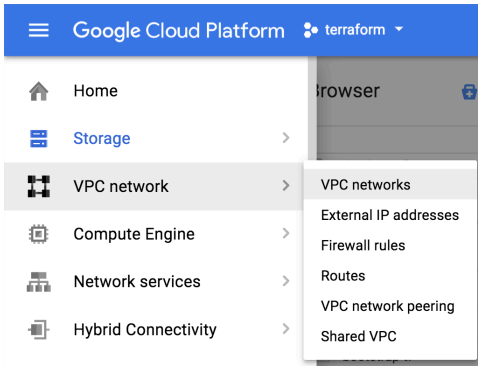
```
Plan: 12 to add, 0 to change, 0 to destroy.

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):

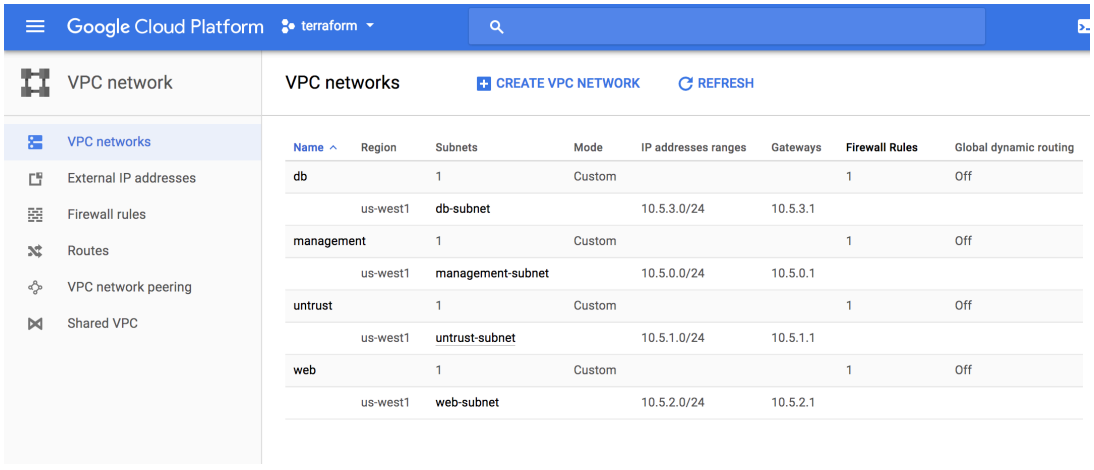
```
Apply complete! Resources: 12 added, 0 changed, 0 destroyed.
```

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**:

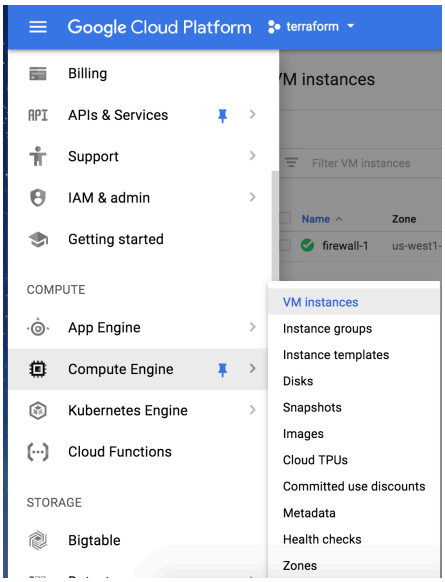


The template creates four networks: db-network, management-network, untrust-network, and web-network:

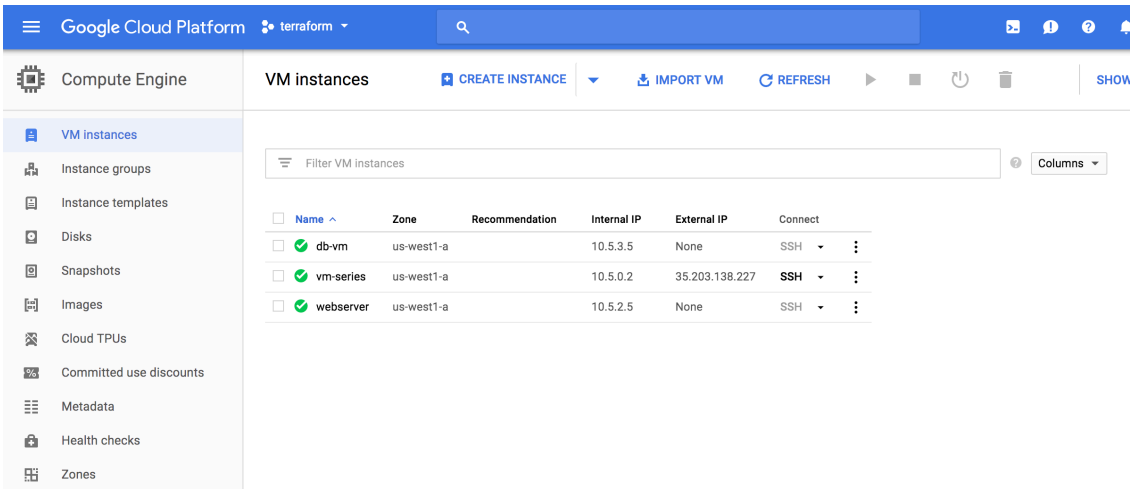


Note: The default network was automatically created when the project was instantiated. It can be ignored or deleted.

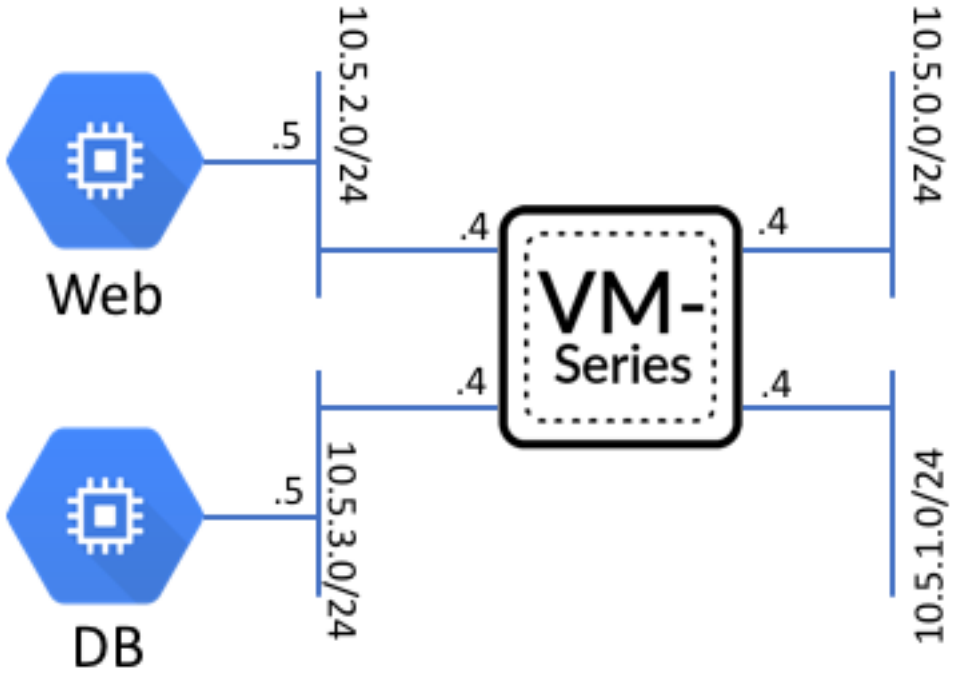
Deployed hosts can be viewed by navigating to **Compute Engine > VM Instances**:



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



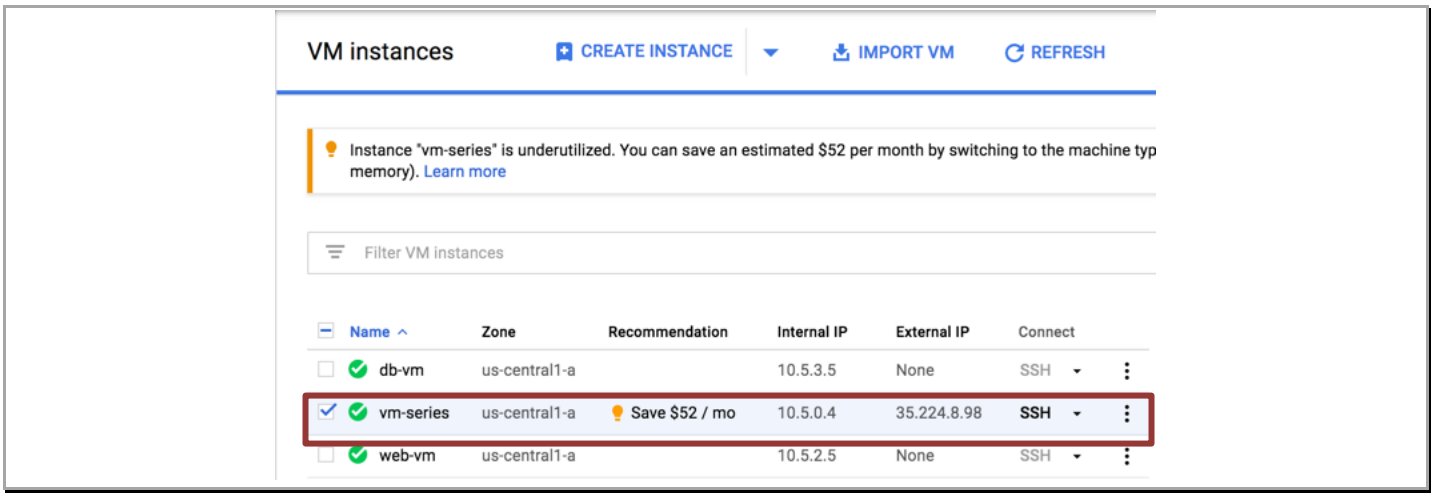
All of this matches the topology shown previously:



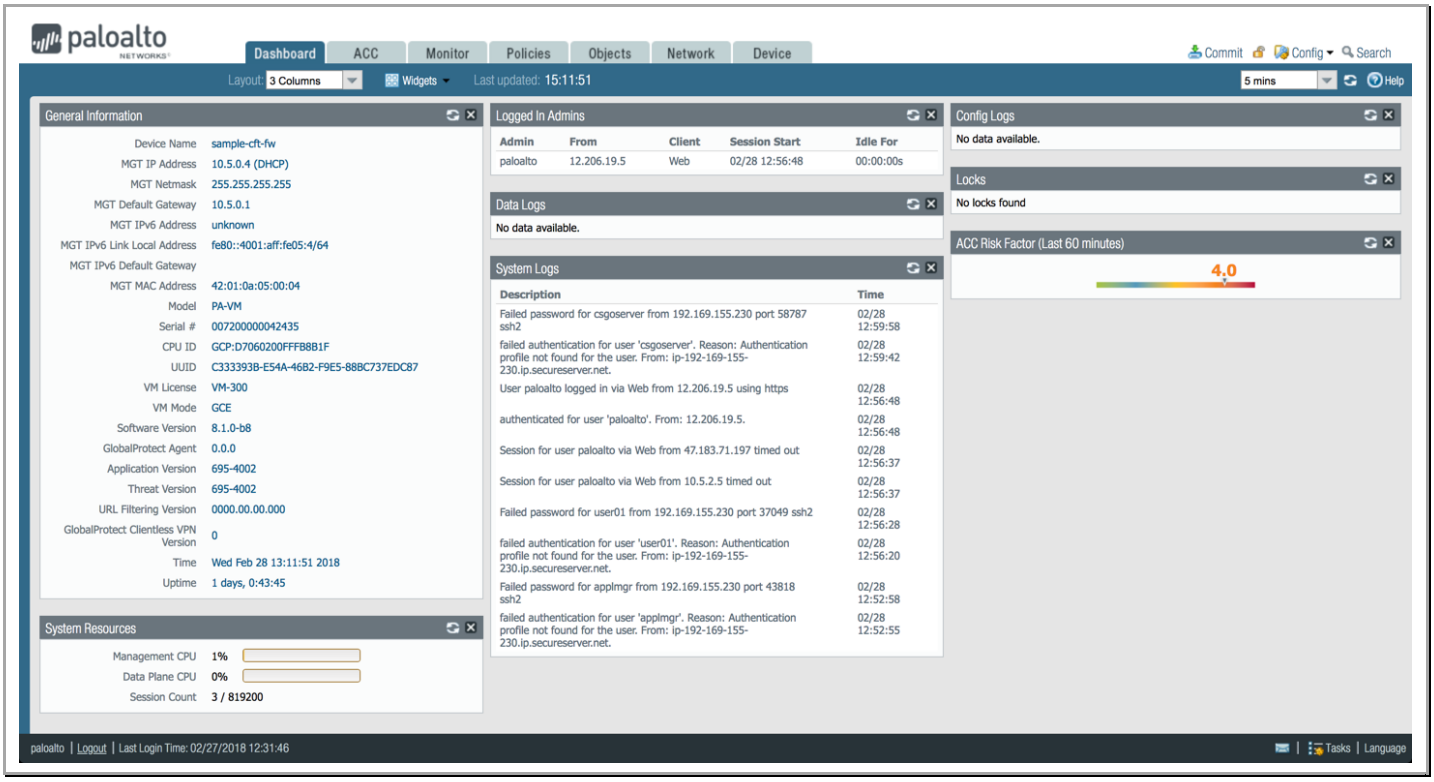
7. Access the firewall

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.

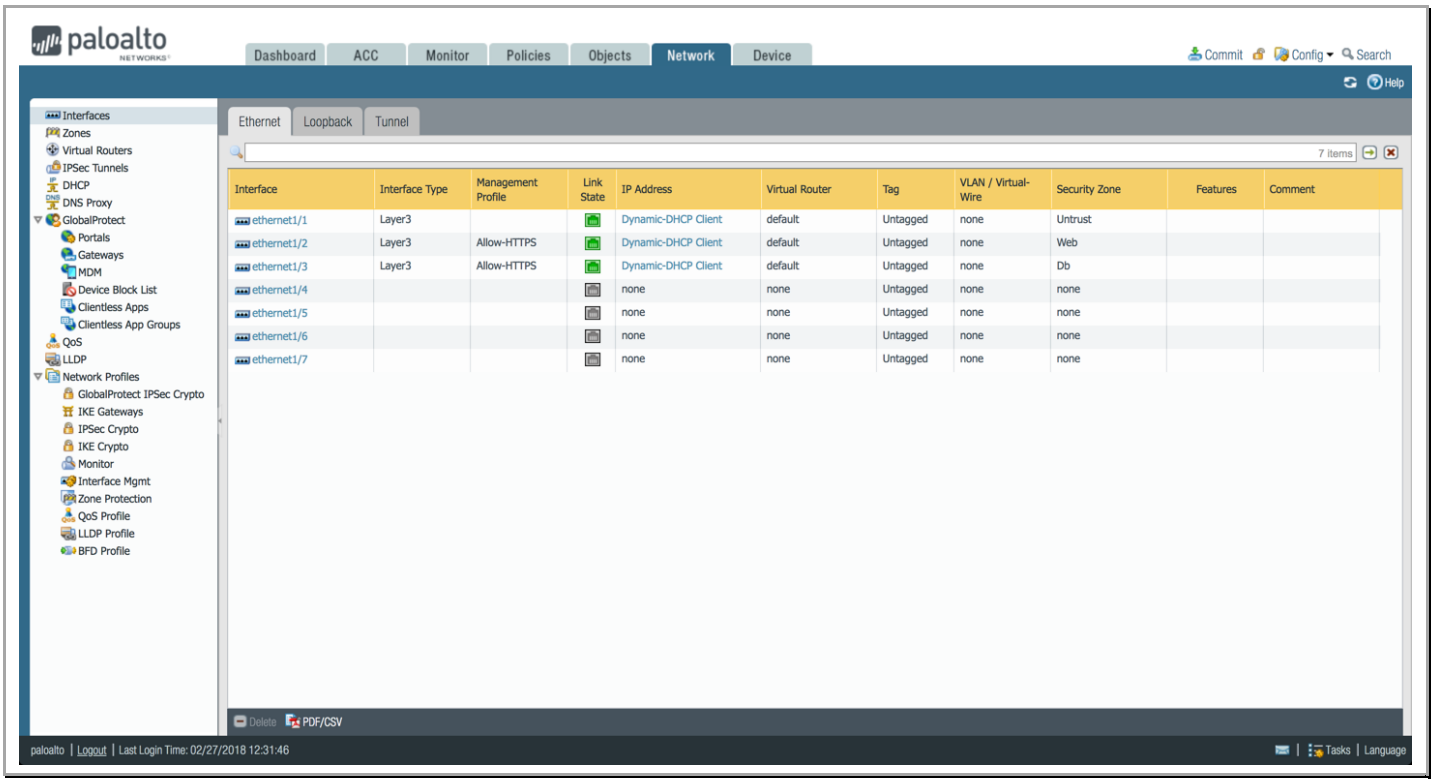
Once the template deployment is complete, the VM-Series firewall will show a green checkmark indicating it is running. On the right side is the management IP address of the VM-Series firewall:



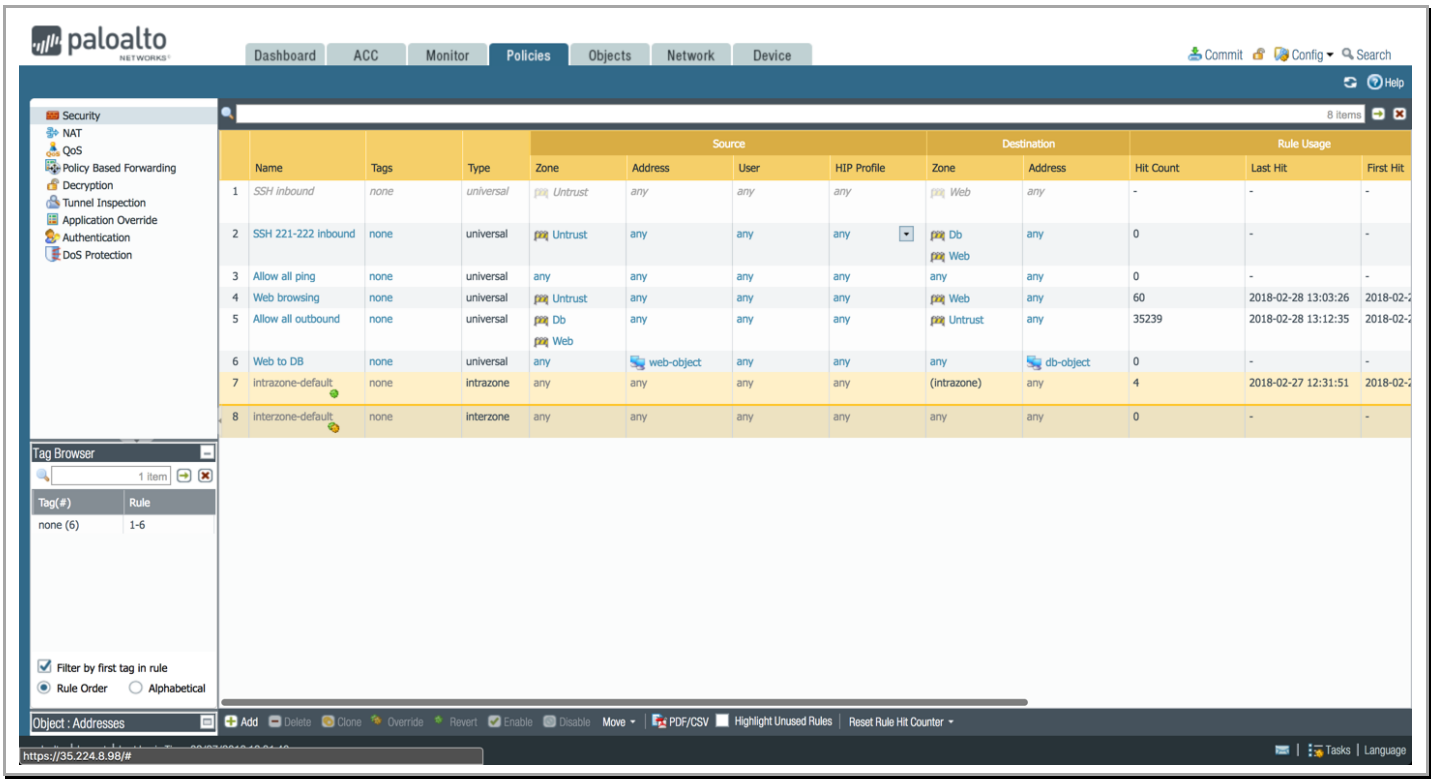
You should now be able to login to the firewall using the **username: paloalto** and password: **Pal0Alt0@123**



Here are the interfaces to zone mappings:

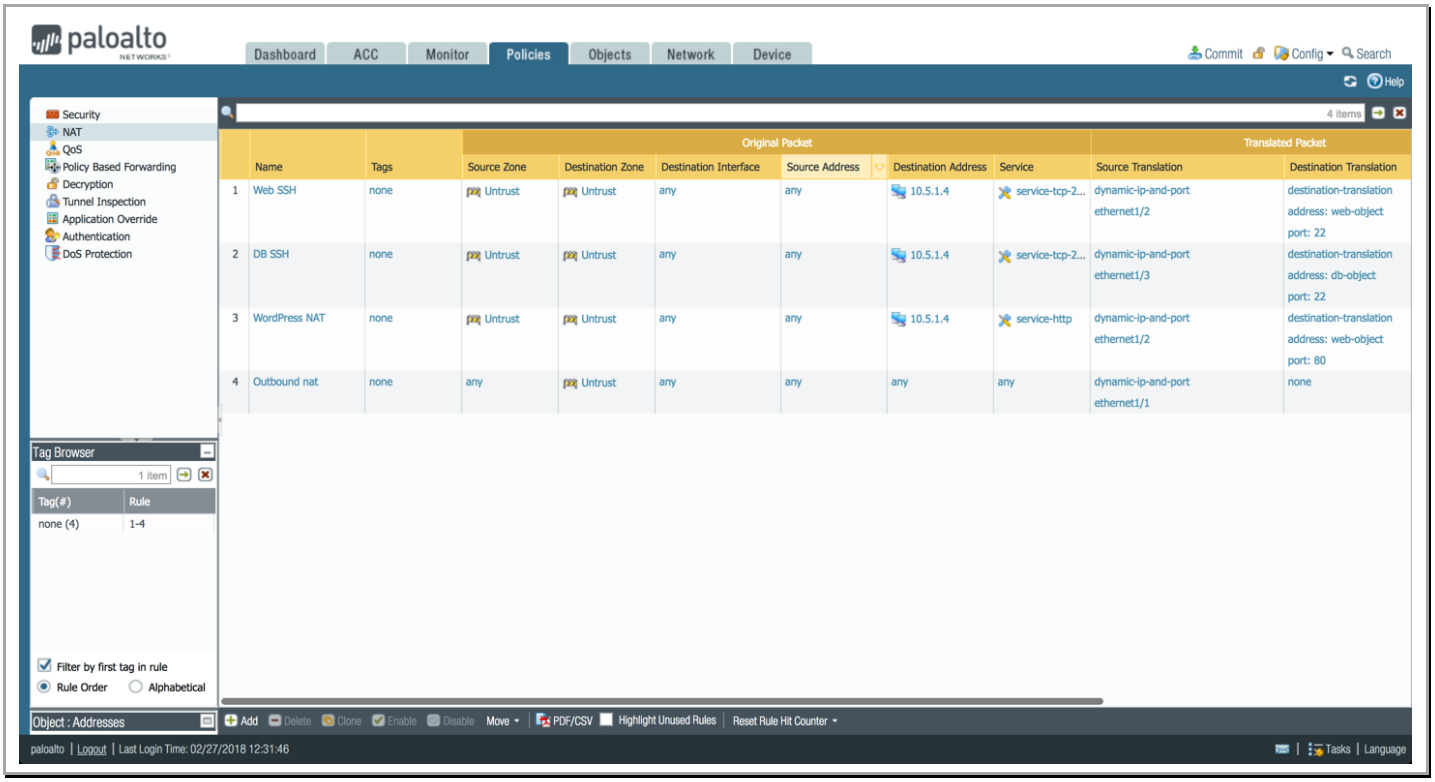


In the policies tab you can review the security policies:



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.



8. Access the Webserver

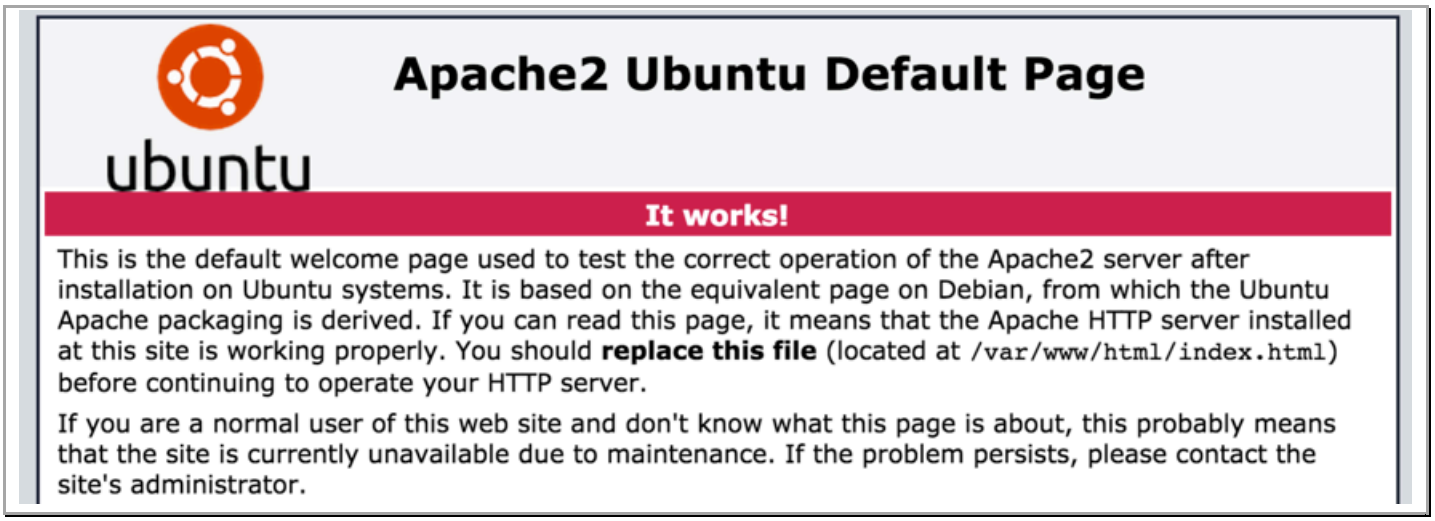
Navigate to the list of VM instances and click on the firewall. Locate the public IP address of the untrust-network interface:

Creation time
Feb 27, 2018, 2:17:57 PM

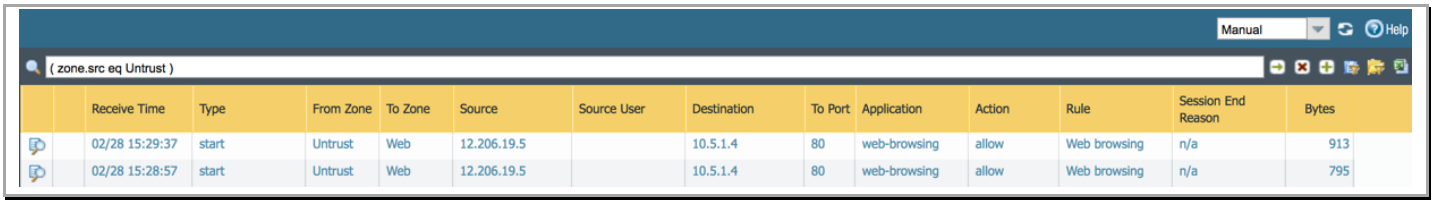
Network interfaces

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	

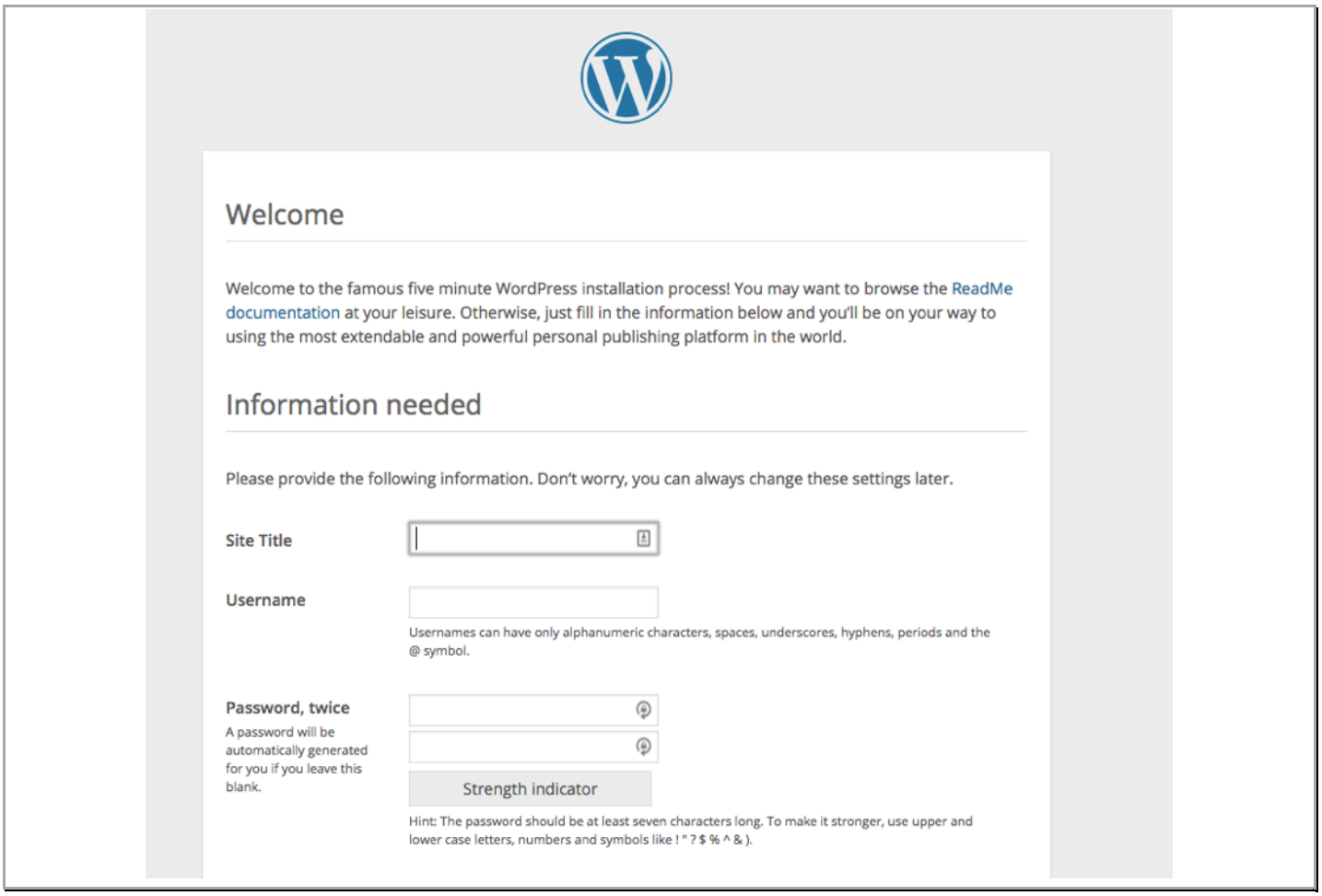
Open a new browser tab and type http://<untrust-network-IP>/ and you should see:



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

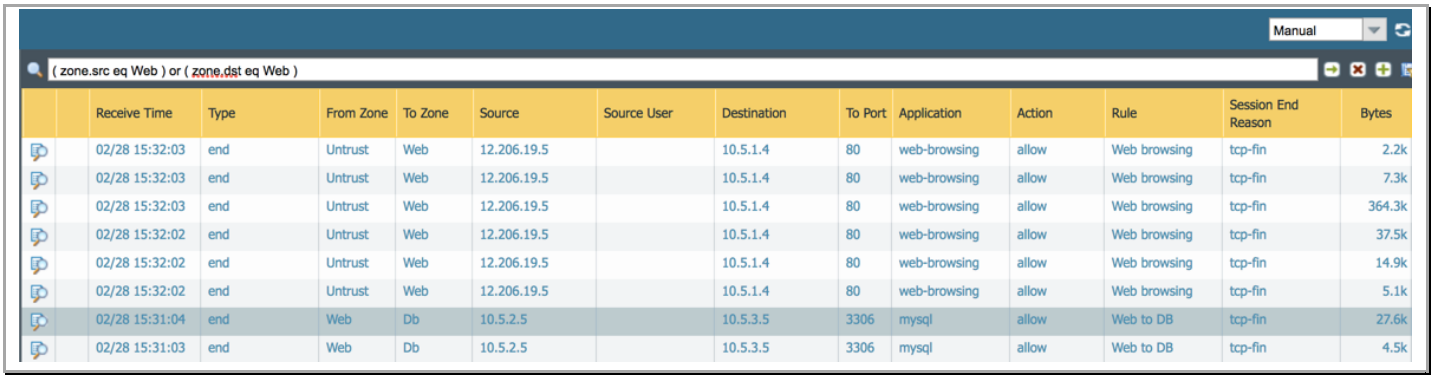


Now let us verify we pass east-West traffic through the firewall. In the browser, head to the wordpress server (http://<untrust-network-IP>/wordpress>). You should see the WordPress welcome screen:



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:



		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

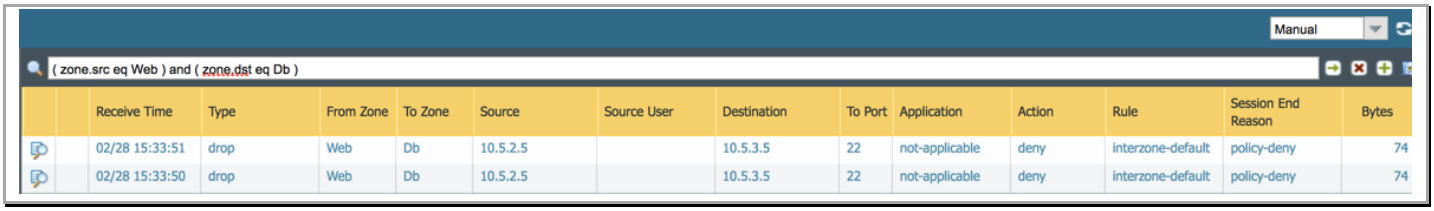
9.1 SSH from Web Server to DB Server

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.**

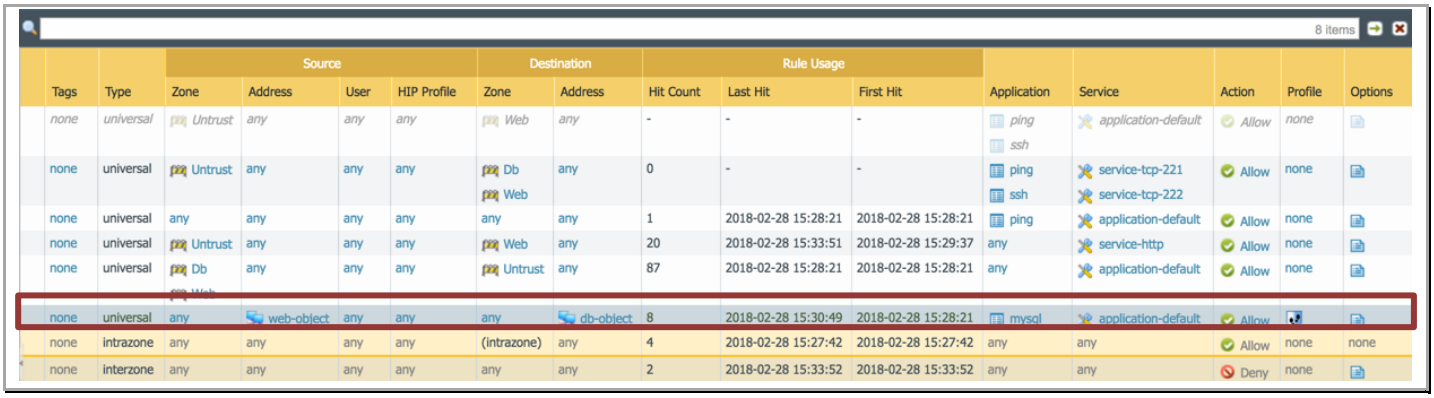
LAUNCH WEB TO DB SSH ATTEMPT

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:

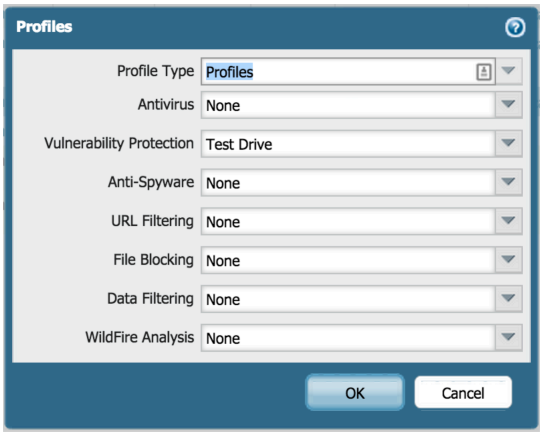


9.2 SQL Brute force attack

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:



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

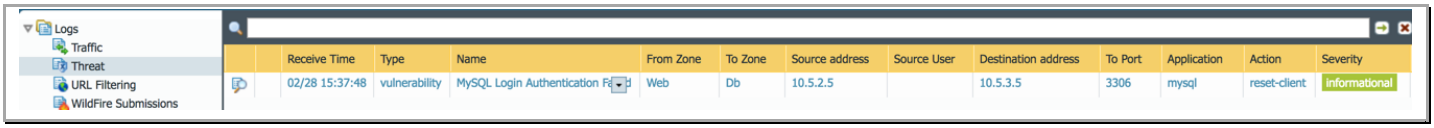


Note the Vulnerability Protection profile. This is a custom profile created just for this lab. It is part of the default vulnerability protection profile but is called out separately for the purpose of this demo environment.

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.

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:



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
- 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.