

VM-Series for GCP



Two-tiered Application Environment on Google Cloud

Using GCP Deployment Manager to deploy a two-tiered application environment secured by the VM-Series next generation firewall

<https://www.paloaltonetworks.com>

Table of Contents

| | |
|----------------------------------------------------------|-----------|
| Version History | 3 |
| 1. About Deployment Manager 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 | 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 | 13 |
| 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 | 21 |
| 7. Access the firewall..... | 24 |
| 8. Access the Webserver..... | 27 |
| 9. Launch some attacks | 30 |
| 9.1 SSH from Web Server to DB Server | 30 |
| 9.2 SQL Brute force attack..... | 30 |
| 10. Cleanup | 32 |
| 10.1 Delete the deployment | 32 |
| 11. Conclusion | 32 |
| Appendix A..... | 32 |
| Troubleshooting tips | 32 |

Version History

| Version number | Comments |
|----------------|-------------------------|
| 1.0 | Initial Draft |
| 1.1 | Update links |
| 1.2 | Update topology diagram |

1. About Deployment Manager Templates

GCP Deployment Manager Templates, are Python (or Jinja) files that can launch nearly all GCP resources including VPCs, subnets, security groups, route tables, plus many more. Templates are used to simplify deployments and are key to fully automating security in the cloud.

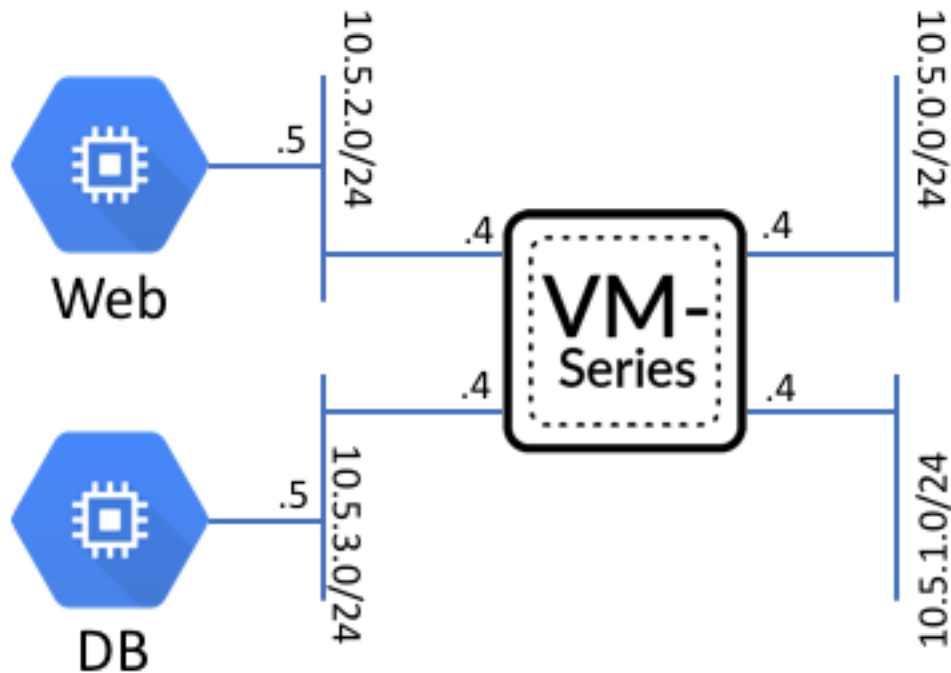
For more information on Templates refer to Google's documentation

<https://cloud.google.com/deployment-manager/docs/how-to#adding-templates>

There are also many sample templates available here

<https://github.com/GoogleCloudPlatform/deploymentmanager-samples/tree/master/templates>

This document will explain how to deploy a sample template that launches a WordPress server, a MySQL server, a VM-Series firewall and the subnets, as shown in the diagram below. The VM-Series uses a native bootstrapping feature that allows for additional configuration of the firewall (such as routes, security policies, etc.)



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 sample 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

The prerequisites required to successfully launch this template are listed below:

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:

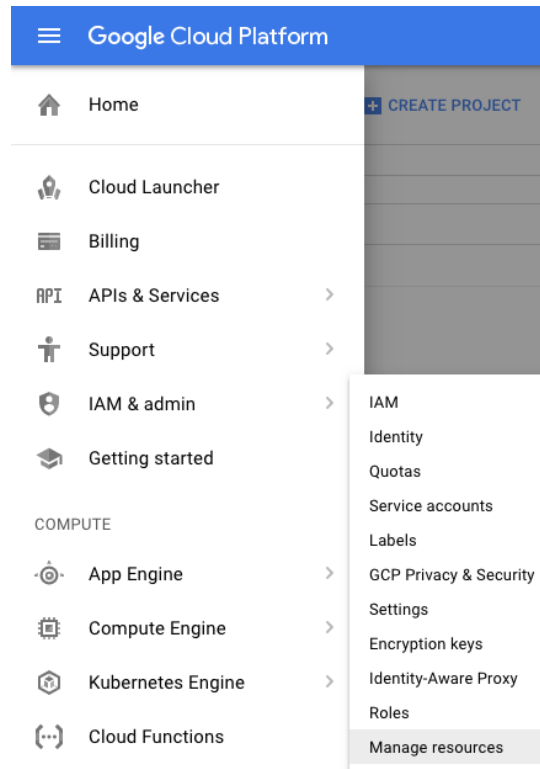
<https://cloud.google.com/sdk/>

4.3 Accept the EULA

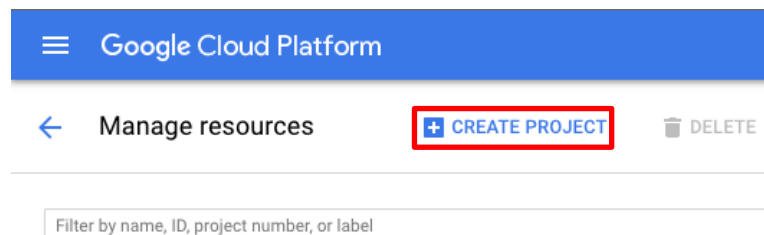
Accept the VM-Series EULA if required.

4.4 Create a Project

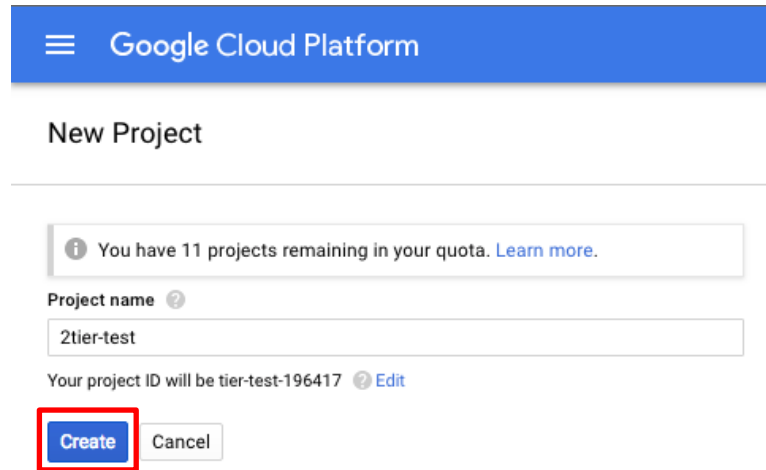
All GCP resources are deployed to a project, which is an organizational boundary that separates users, resources, billing information, etc. It is similar to an AWS VPC or an Azure Resource Group. By default, GCP will create a project upon creation of an account. If that is not the case or to create a 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**:



Google Cloud Platform

New Project

You have 11 projects remaining in your quota. [Learn more.](#)

Project name ?

2tier-test

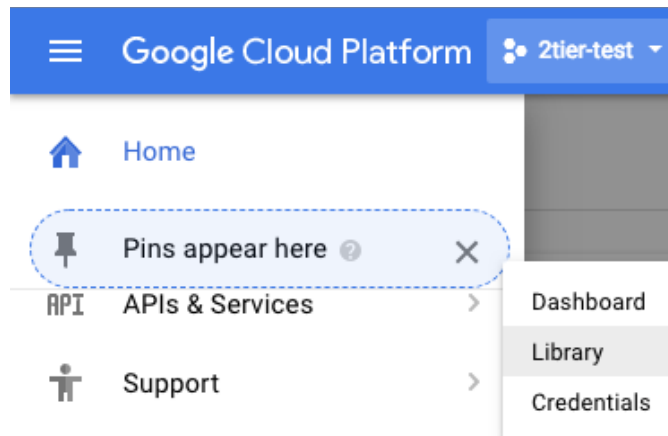
Your project ID will be tier-test-196417 ? [Edit](#)

Create Cancel

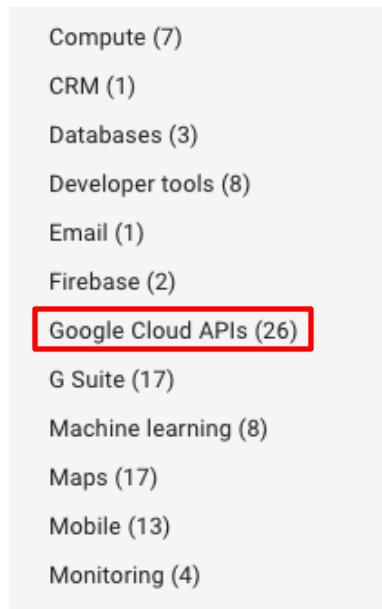
Note that project creation will take a few minutes.

4.5 Enable the API

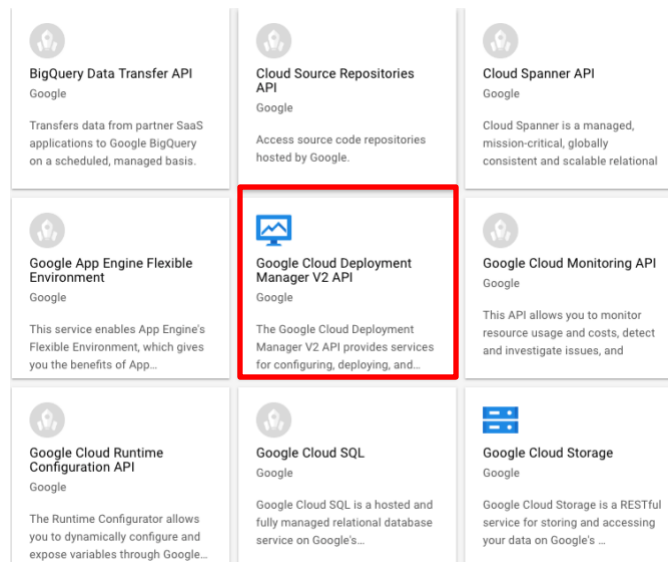
Deploying a template requires the API be enable on the project. Navigate to **APIs & Services > Library**:



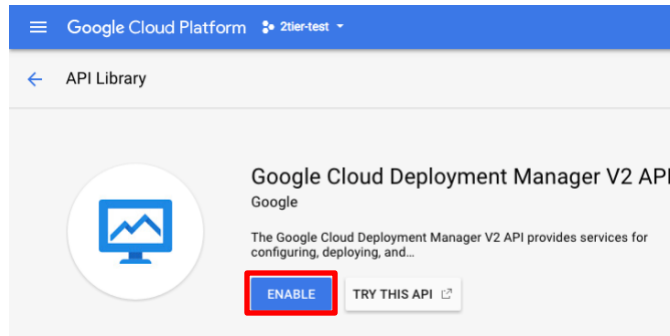
Select Google Cloud APIs on the left-hand-side:



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



Select **Enable**:

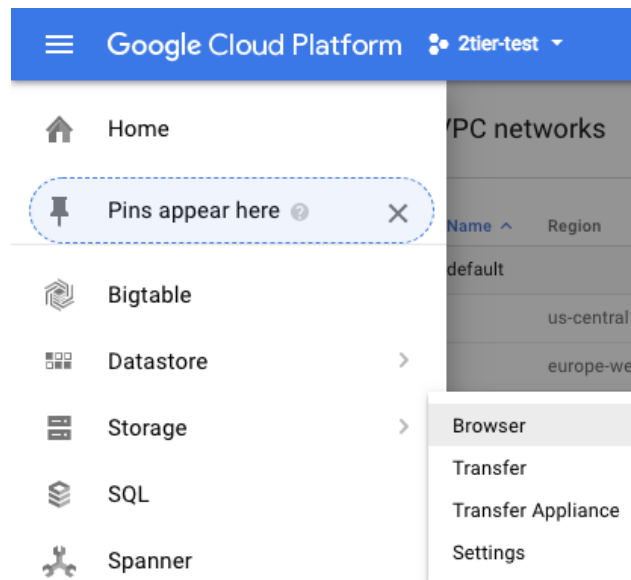


Enabling the API for the project will take a few minutes to complete.

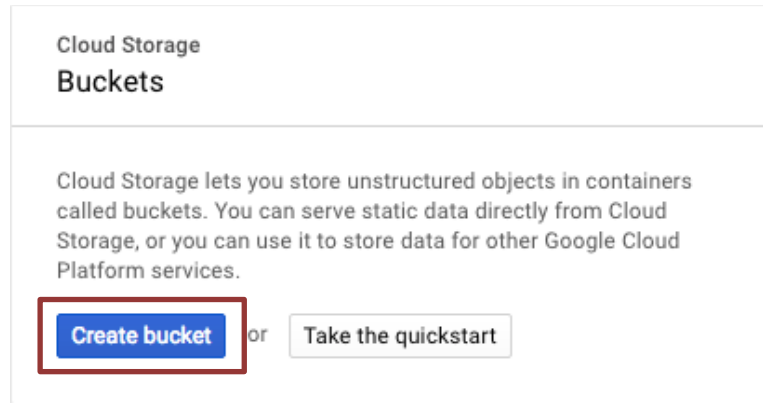
4.6 Create a Bootstrap Bucket

Bootstrapping is a feature of the VM-Series firewall that allows you to create and store a pre-defined configuration that can then be used by the firewall during boot-up. This ensures that the firewall is fully configured at initial boot-up, thereby removing the need for manual configuration. Bootstrapping also enables VM-Series deployments to be fully automated.

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



Click **Create Bucket**:

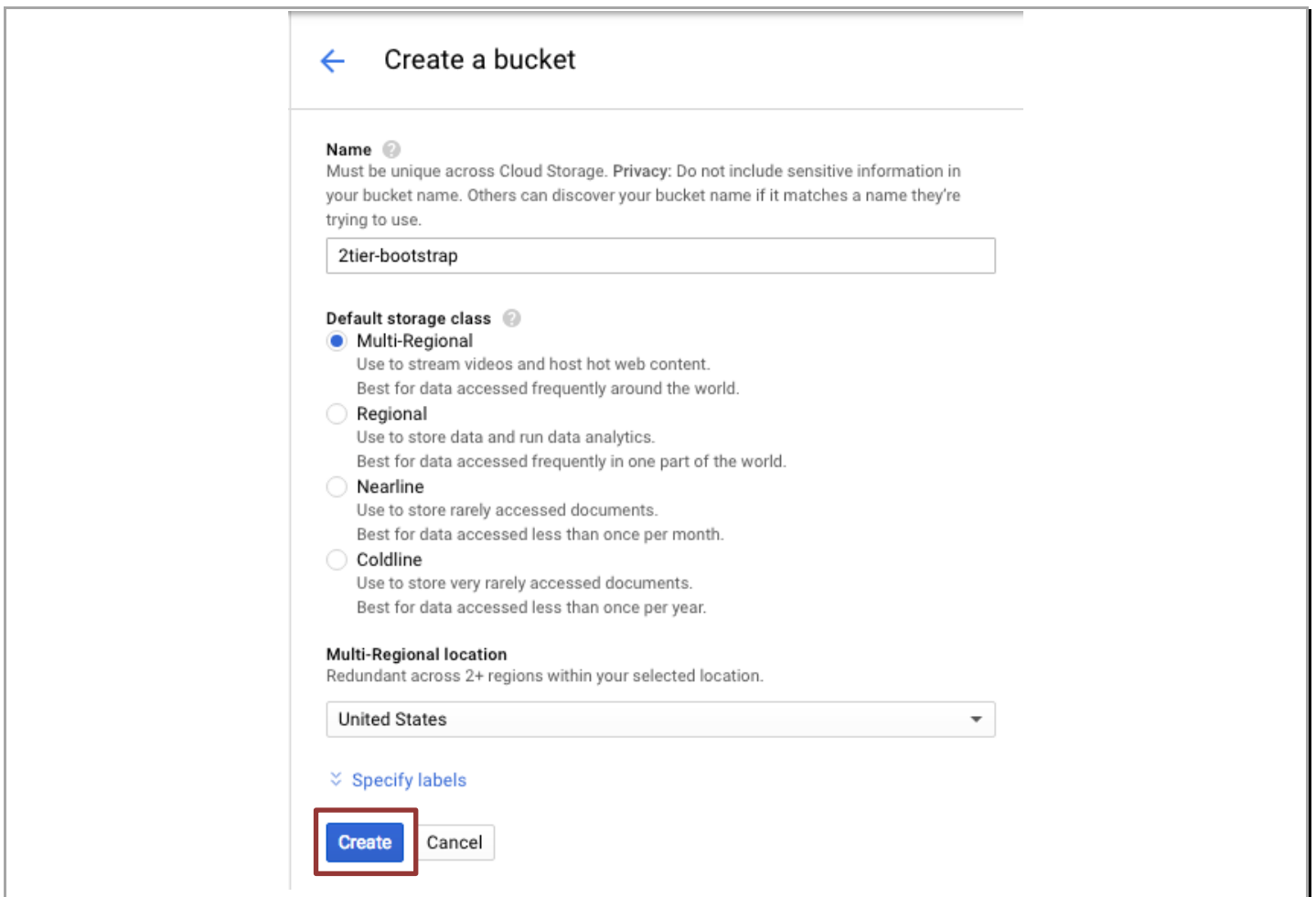


Cloud Storage
Buckets

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.

2tier-bootstrap

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.

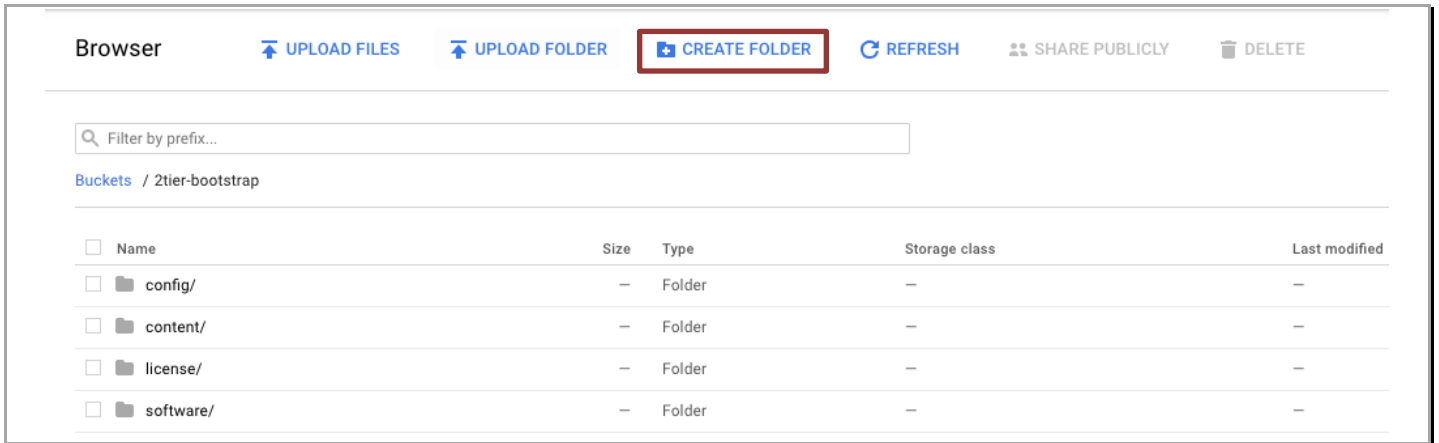
Multi-Regional location
Redundant across 2+ regions within your selected location.

United States

⌵ 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**:



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

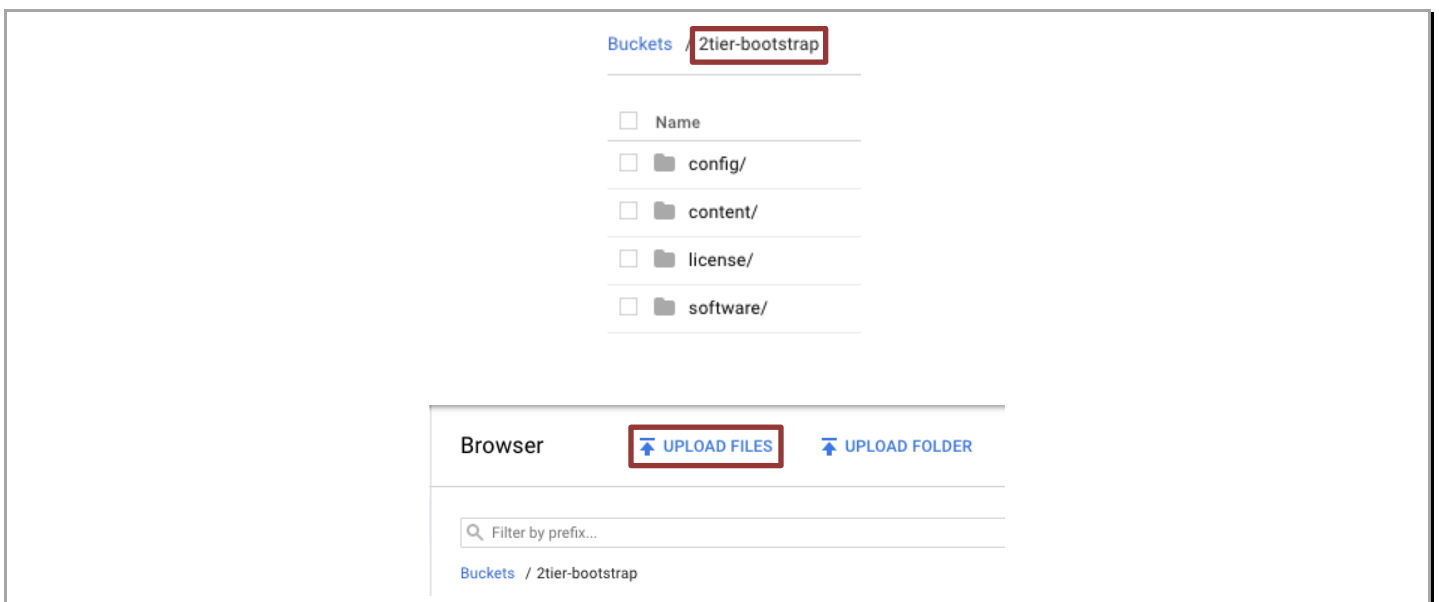
<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/bootstrap.xml>

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

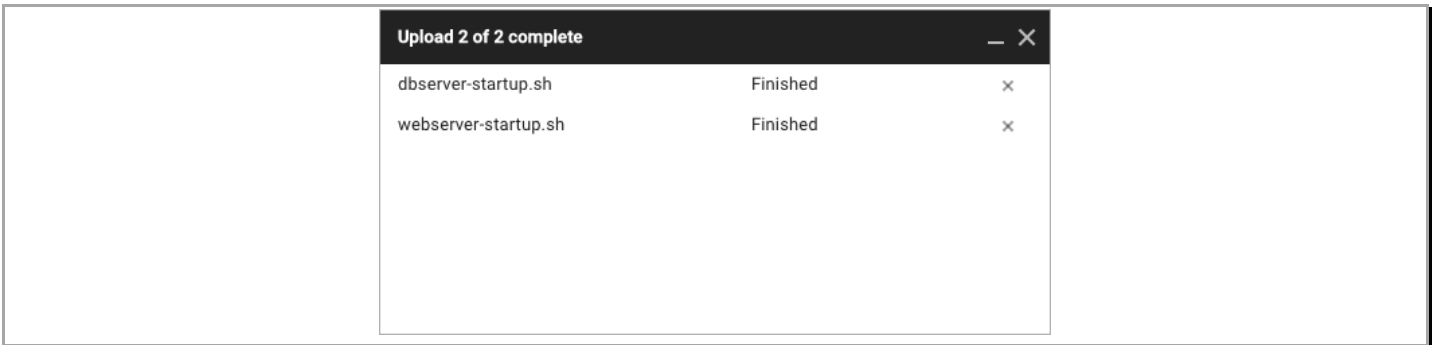
<https://raw.githubusercontent.com/PaloAltoNetworks/googlecloud/master/two-tier-template/dbserver-startup.sh>

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

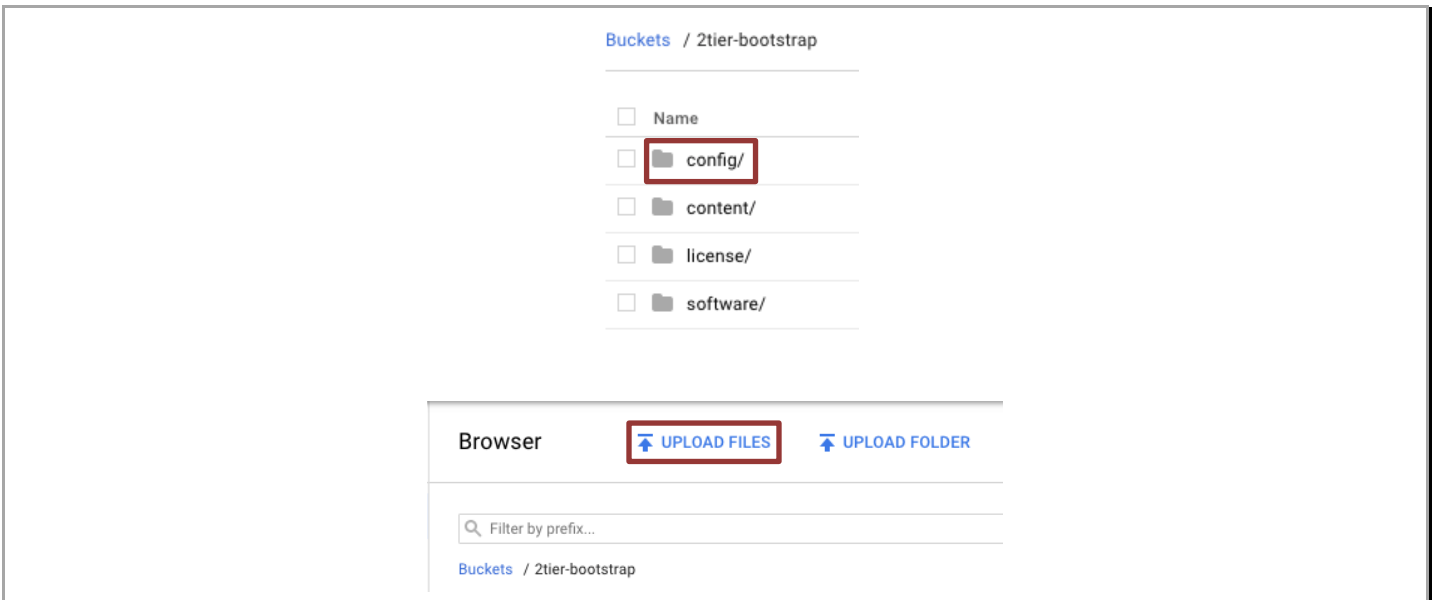
Now click on the root folder in the console and click **UPLOAD FILES**:



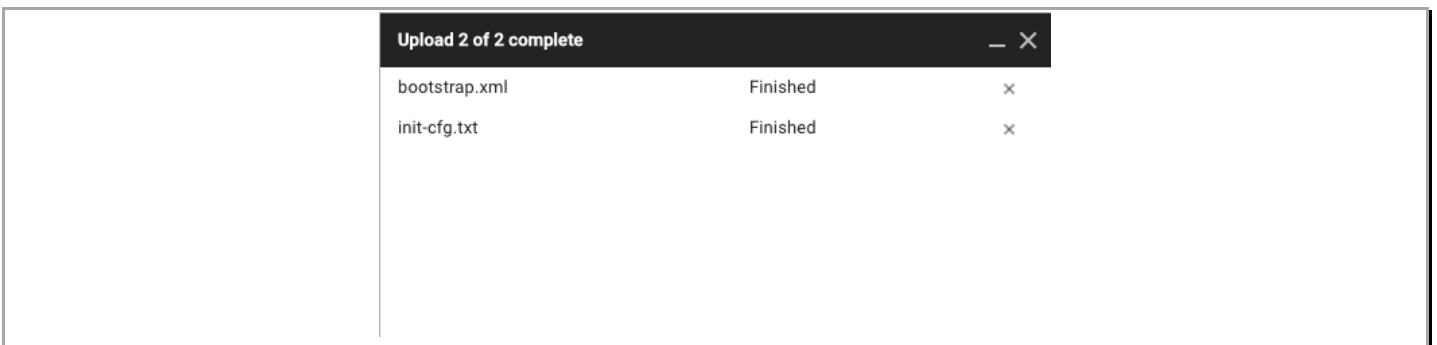
Select the two files (dbserver-startup.sh and dbserver-startup.sh) downloaded previously and click **Open**:



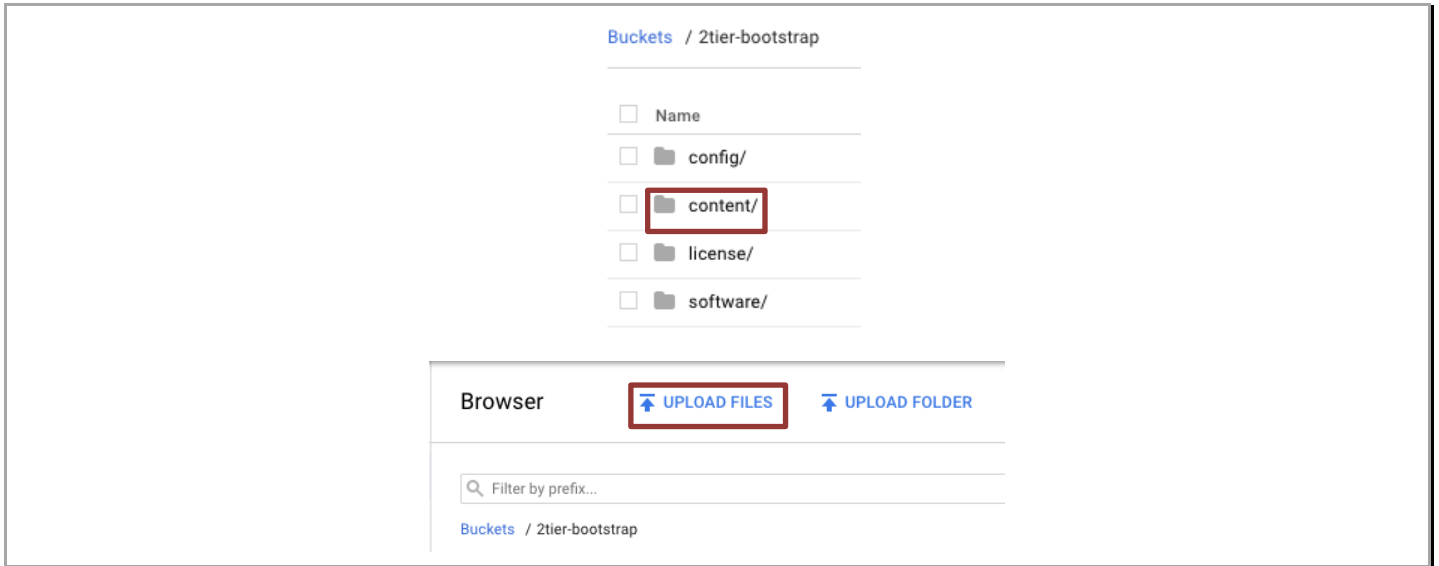
Now click on the **config** folder in the console and click **UPLOAD FILES**:



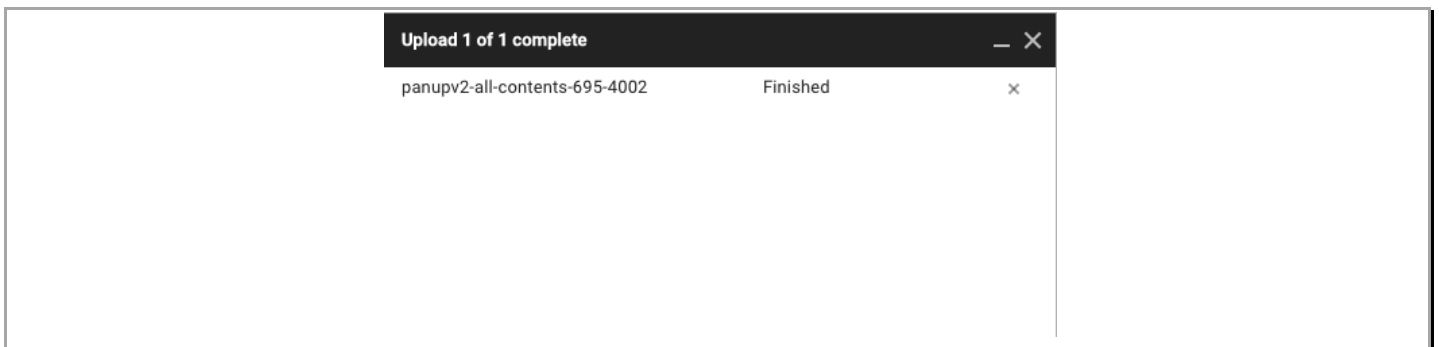
Select the two files (bootstrap.xml and init-cft.txt) downloaded previously and click **Open**:



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



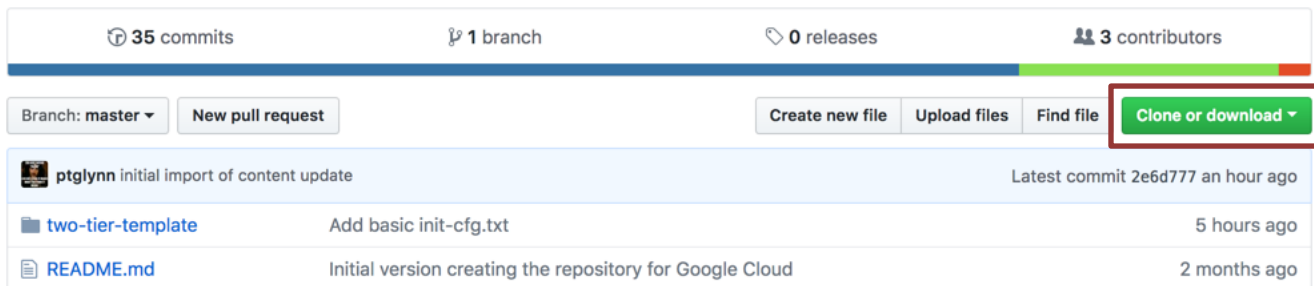
Select the content file (panupv2-all-contents-695-4002) downloaded previously and click **Open**:



NOTE: Please create the folders using the console. Creating folders locally on your machine and uploading them may not work as expected.

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:

| Name | Size | Kind | Date Added |
|-----------------------|-----------|-----------------|----------------|
| ▼ googlecloud-master | -- | Folder | Today at 19:08 |
| ▼ two-tier-template | -- | Folder | Today at 19:08 |
| webserver...mplate.py | 3 KB | TextWr...cument | Today at 19:08 |
| webserver-startup.sh | 3 KB | TextWr...cument | Today at 19:08 |
| vm-series...mplate.py | 4 KB | TextWr...cument | Today at 19:08 |
| two-tier-template.py | 7 KB | TextWr...cument | Today at 19:08 |
| two-tier-sample.yaml | 1 KB | TextWr...cument | Today at 19:08 |
| subnetwo...mplate.py | 1 KB | TextWr...cument | Today at 19:08 |
| sql-attack.html | 606 bytes | HTML | Today at 19:08 |
| network-template.py | 906 bytes | TextWr...cument | Today at 19:08 |
| init-cfg.txt | 156 bytes | Plain Text | Today at 19:08 |
| dbserver-template.py | 3 KB | TextWr...cument | Today at 19:08 |
| dbserver-startup.sh | 2 KB | TextWr...cument | Today at 19:08 |
| bootstrap.xml | 36 KB | TextWr...cument | Today at 19:08 |
| Readme.md | 1 KB | TextWr...cument | Today at 19:08 |
| README.md | 1 KB | TextWr...cument | Today at 19:08 |

4.9 Gather Information and Update the Template File

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

```

zone = ###ZONE (e.g. us-central1-a, us-east4-b, etc.)

region = ###Region (e.g. us-central1, us-east4, etc.)

sshkey = ## ssh key PUBLIC

    Format: <USERNAME>:<KEY FORMAT> <PUBLIC KEY> <USERNAME>

bootstrap_bucket = ###bootstrap bucket

scripts_bucket = ###bucket with web and db startup scripts

serviceaccount = ###GCP service account (Retrieve from IAM & admin > IAM)

    Format: <PROJECT NUMBER>-compute@developer.gserviceaccount.com

```

Once the information has been gathered, update the file “two-tier-template.py” with the information.

NOTE: All variables MUST be enclosed in quotes. Failure to do so will result in a deployment failure.

```
#Variables
zone = "us-central1"
region = "us-central1-a"
sshkey = "pglynn:ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDFdrfdjQAR/cgz
URE6o/1JlqhYq+kqHxcq2ZNM4yVDhmVw+ggQpqMdot02FdYyuxmHGuxlaLBJxPomqnKS
p6yHjgj+14G+ohaZJmnAwp6audmuGVED2ybVfopg6vXwYWHhWlJY77nDI+qCU5nTe29Y6
ZoSoObbYZAjf69MtAS1vnQodwnS96oJ8dghZ31+fym1Vt0yXYfgBOaN2+Bb+GQkWkxD7P
q+PEXwq/w+2j6wvflEFAUd5sLz8vOoUoDaVbTUo2AhGUQfwe5RlL3SC7si4Pt7Xiel+
koMnxg32RKzPS4dwzzf1FIIGeaaSgmQModnCHqb3g37sYjrmaUHjpglynn"
bootstrap_bucket = "2tier-bootstrap"
scripts_bucket = "2tier-bootstrap"
serviceaccount = "815575352539-compute@developer.gserviceaccount.com"
image = "panos-8-1-b53"
```

5. Launch the Template

Navigate to the directory containing the downloaded template files:

```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# ls
Readme.md                               subnet-template.py
bootstrap.xml                           two-tier-sample.yaml
dbserver-startup.sh                    two-tier-template.py
dbserver-template.py                  vm-series-template.py
init-cfg.txt                           webserver-startup.sh
network-template.py                   webserver-template.py
sql-attack.html
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template#
```

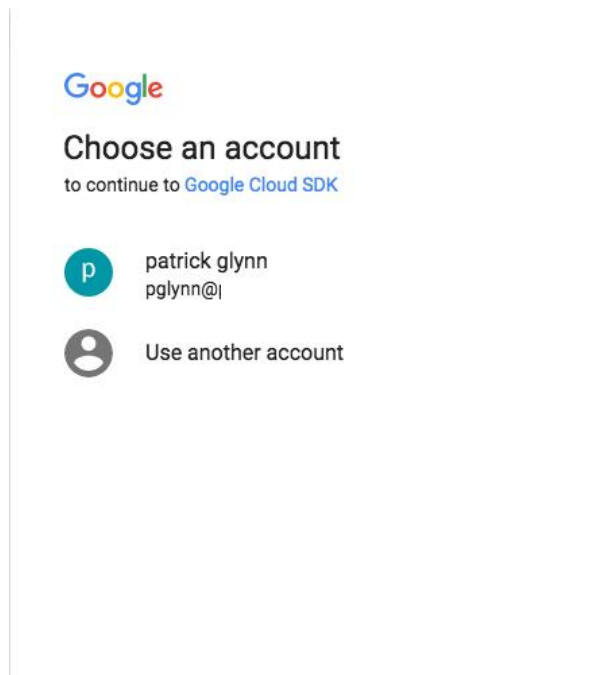
Authenticate to the GCP environment:

```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# gcloud auth login pglynn@
Go to the following link in your browser:

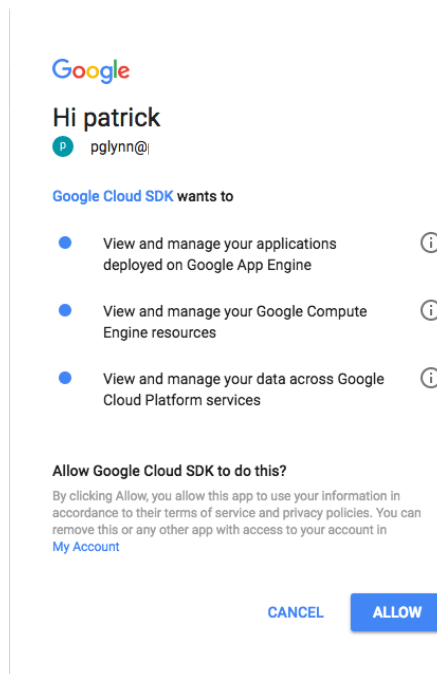
https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.o%3Aob&prompt=select_account&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&access_type=offline

Enter verification code: 
```

Copy/paste the link into a browser and select the account to authenticate:



Review the requested permissions and click **Allow**:



Copy the one-time verification code:



Please copy this code, switch to your application and paste it there:

4/AAC2EoRH6UAJWLhKHk7oYp8CWzQzZf7miebTJQxhRcw7EiSCEIE8ELo

Paste it into the window to complete the authentication request (ignore the warning):

```

root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# gcloud auth login pglynn@
Go to the following link in your browser:

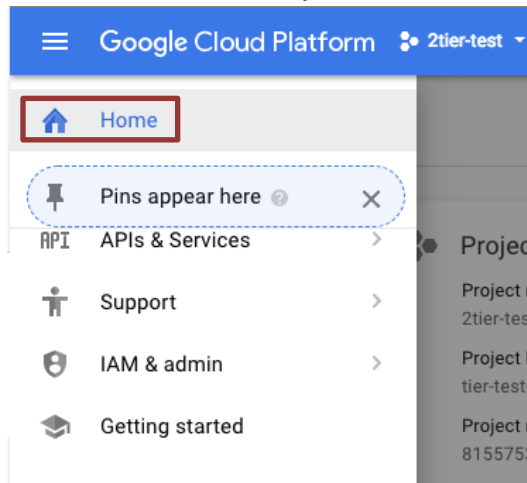
https://accounts.google.com/o/oauth2/auth?redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3Aoob&prompt=select_account&response_type=code&client_id=32555940559.apps.googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&access_type=offline

Enter verification code: 4/AAC2EoRH6UAJWLhKHk7oYp8CWzQzZf7miebTJQxhRcw7EiSCEIE8ELo
WARNING: `gcloud auth login` no longer writes application default credentials.
If you need to use ADC, see:
gcloud auth application-default --help

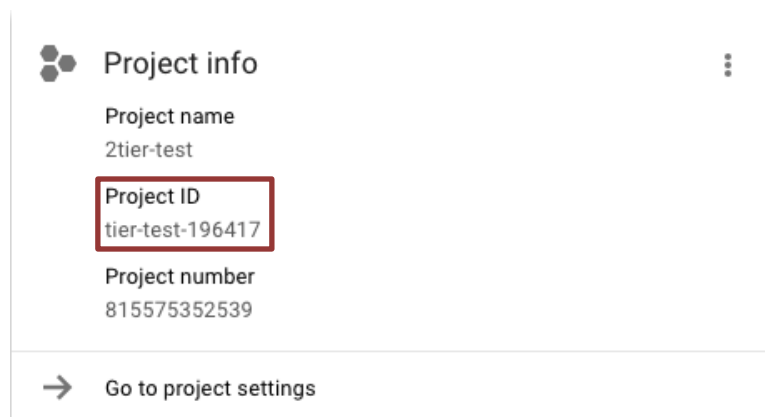
You are now logged in as [pglynn@].
Your current project is [pgtest-two-tier]. You can change this setting by running:
$ gcloud config set project PROJECT_ID
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# █

```

Select **Home** from the dropdown in the console:



Note the Project ID:



Set the target project for template deployment:

```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# gcloud config set project tier-test-196417
Updated property [core/project].
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template#
```

Initiate template deployment using the file “two-tier-sample.yaml” as the configuration file using the command “gcloud deployment-manager deployments create <deployment name> --config two-tier-sample.yaml --automatic-rollback-on-error” where <deployment name> is the tag used to identify the deployment (“deployment1” in this example):

```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# gcloud deployment-manager deployments create deployment1 --config two-tier-sample.yaml --automatic-rollback-on-error
```

Note: The “--automatic-rollback-on-error” flag instructs deployment manager to automatically roll back all changes in the event of a failure to deploy. For troubleshooting purposes, it may be desirable to omit this flag.

If all goes well, Deployment Manager will report success (state “COMPLETED” and no errors):

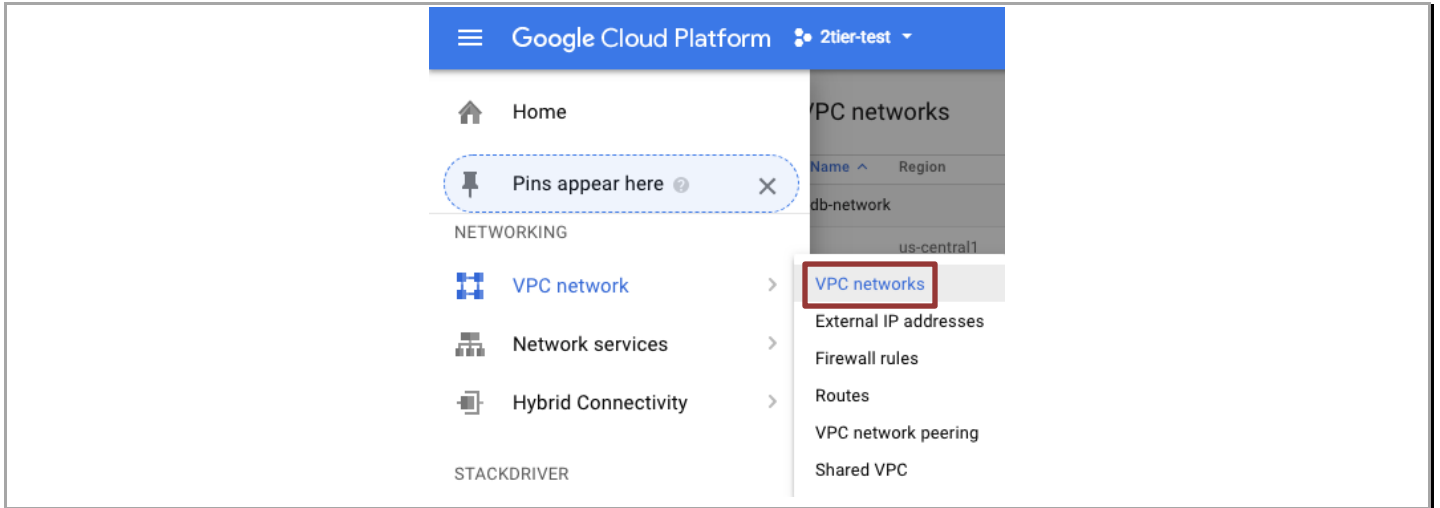
```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template# gcloud deployment-manager deployments create deployment1 --config two-tier-sample.yaml --automatic-rollback-on-error
The fingerprint of the deployment is GL54qACJ7Av8Yyjp_oWKyA==
Waiting for create [operation-1519750942493-5663497130948-c29c8c79-2093fbc2]... done.
Create operation operation-1519750942493-5663497130948-c29c8c79-2093fbc2 completed successfully.
```

| NAME | TYPE | STATE | ERRORS | INTENT |
|---------------------|-----------------------|-----------|--------|--------|
| db-firewall | compute.v1.firewall | COMPLETED | [] | |
| db-network | compute.v1.network | COMPLETED | [] | |
| db-route | compute.v1.route | COMPLETED | [] | |
| db-subnet | compute.v1.subnetwork | COMPLETED | [] | |
| db-vm | compute.v1.instance | COMPLETED | [] | |
| management-firewall | compute.v1.firewall | COMPLETED | [] | |
| mgmt-network | compute.v1.network | COMPLETED | [] | |
| mgmt-subnet | compute.v1.subnetwork | COMPLETED | [] | |
| untrust-firewall | compute.v1.firewall | COMPLETED | [] | |
| untrust-network | compute.v1.network | COMPLETED | [] | |
| untrust-subnet | compute.v1.subnetwork | COMPLETED | [] | |
| vm-series | compute.v1.instance | COMPLETED | [] | |
| web-firewall | compute.v1.firewall | COMPLETED | [] | |
| web-network | compute.v1.network | COMPLETED | [] | |
| web-route | compute.v1.route | COMPLETED | [] | |
| web-subnet | compute.v1.subnetwork | COMPLETED | [] | |
| web-vm | compute.v1.instance | COMPLETED | [] | |

```
root@2ccf4d39089e: ~/Development/GCP/googlecloud-master/two-tier-template#
```

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, mgmt-network, untrust-network, and web-network:

Palo Alto Networks GCP Deployment Manager Deployment Guide

VPC networks

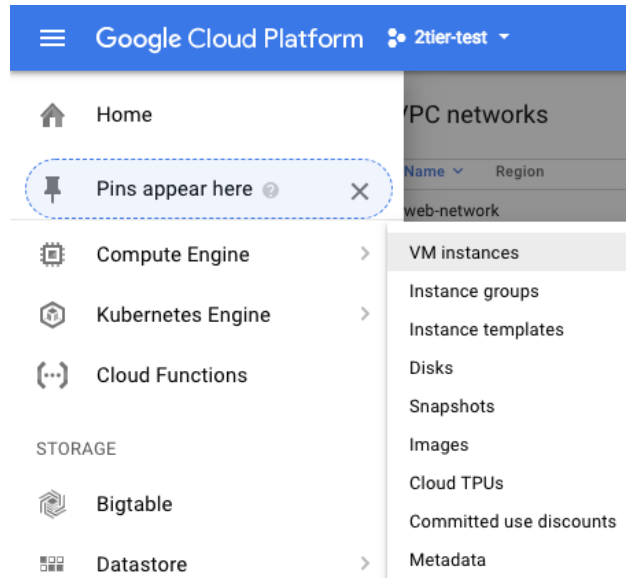
+ CREATE VPC NETWORK

REFRESH

| Name | Region | Subnets | Mode | IP addresses ranges | Gateways | Firewall Rules | Global dynamic routing |
|-----------------|-------------------------|----------------|--------|---------------------|------------|----------------|------------------------|
| web-network | | 1 | Custom | | | 1 | Off |
| | us-central1 | web-subnet | | 10.5.2.0/24 | 10.5.2.1 | | |
| untrust-network | | 1 | Custom | | | 1 | Off |
| | us-central1 | untrust-subnet | | 10.5.1.0/24 | 10.5.1.1 | | |
| mgmt-network | | 1 | Custom | | | 1 | Off |
| | us-central1 | mgmt-subnet | | 10.5.0.0/24 | 10.5.0.1 | | |
| default | | 15 | Auto | | | 4 | Off |
| | us-central1 | default | | 10.128.0.0/20 | 10.128.0.1 | | |
| | europe-west1 | default | | 10.132.0.0/20 | 10.132.0.1 | | |
| | us-west1 | default | | 10.138.0.0/20 | 10.138.0.1 | | |
| | asia-east1 | default | | 10.140.0.0/20 | 10.140.0.1 | | |
| | us-east1 | default | | 10.142.0.0/20 | 10.142.0.1 | | |
| | asia-northeast1 | default | | 10.146.0.0/20 | 10.146.0.1 | | |
| | asia-southeast1 | default | | 10.148.0.0/20 | 10.148.0.1 | | |
| | us-east4 | default | | 10.150.0.0/20 | 10.150.0.1 | | |
| | australia-southeast1 | default | | 10.152.0.0/20 | 10.152.0.1 | | |
| | europe-west2 | default | | 10.154.0.0/20 | 10.154.0.1 | | |
| | europe-west3 | default | | 10.156.0.0/20 | 10.156.0.1 | | |
| | southamerica-east1 | default | | 10.158.0.0/20 | 10.158.0.1 | | |
| | asia-south1 | default | | 10.160.0.0/20 | 10.160.0.1 | | |
| | northamerica-northeast1 | default | | 10.162.0.0/20 | 10.162.0.1 | | |
| | europe-west4 | default | | 10.164.0.0/20 | 10.164.0.1 | | |
| db-network | | 1 | Custom | | | 1 | Off |
| | us-central1 | db-subnet | | 10.5.3.0/24 | 10.5.3.1 | | |

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

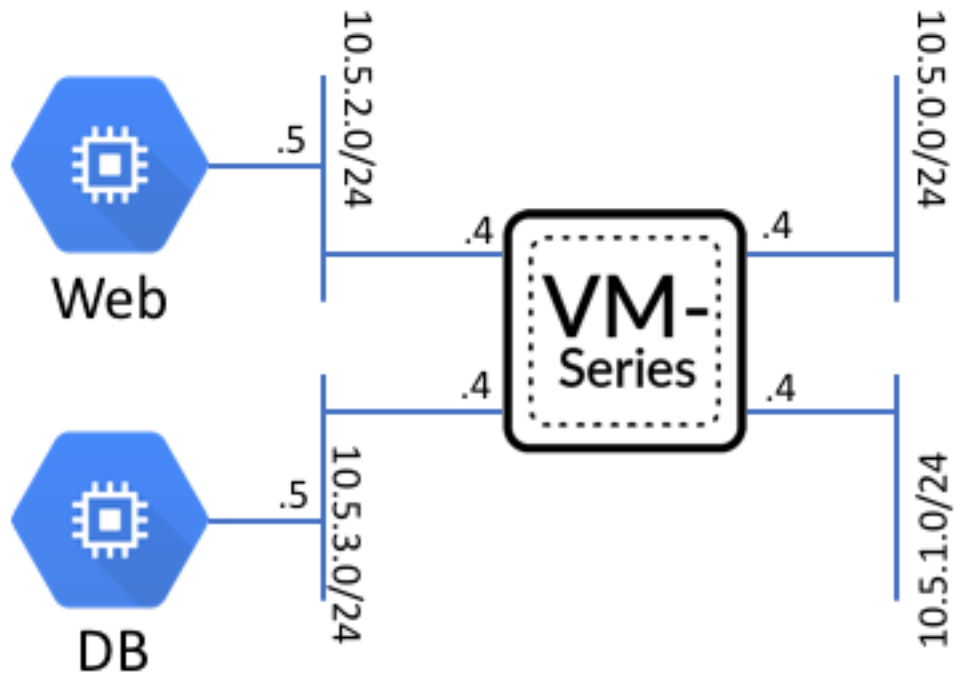
Deployed hosts can be viewed by navigating to **Compute Engine > VM Instances**:



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

| VM instances | | | | | | |
|-----------------------------------------------------------------------------------|---------------|----------------|-------------|-------------|---------|---|
| CREATE INSTANCE IMPORT VM REFRESH | | | | | | |
| Filter VM instances | | | | | | |
| <input type="checkbox"/> Name ^ | Zone | Recommendation | Internal IP | External IP | Connect | |
| <input type="checkbox"/> db-vm | us-central1-a | | 10.5.3.5 | None | SSH | ⋮ |
| <input type="checkbox"/> vm-series | us-central1-a | | 10.5.0.4 | 35.224.8.98 | SSH | ⋮ |
| <input type="checkbox"/> web-vm | us-central1-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. So once the template has been deployed successfully, it may take a few minutes before the firewall is up and you are able to log in. .

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

Palo Alto Networks GCP Deployment Manager 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 | |
|-------------------------------------|-----------|---------------|----------------|-------------|-------------|---------|--|
| <input type="checkbox"/> | db-vm | us-central1-a | | 10.5.3.5 | None | SSH | |
| <input checked="" type="checkbox"/> | vm-series | us-central1-a | Save \$52 / mo | 10.5.0.4 | 35.224.8.98 | SSH | |
| <input type="checkbox"/> | 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**

paloalto

NETWORKS

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-ctf-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

MGT MAC Address

42:01:0a:05:00:04

Model

PA-VM

Serial #

007200000042435

CPU ID

GCP-D7060200FFB881F

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

0000.00.00.000

GlobalProtect Clientless VPN Version

0

Time

Wed Feb 28 13:11:51 2018

Uptime

1 days, 0:43:45

System Resources

Management CPU

1%

Data Plane CPU

0%

Session Count

3 / 819200

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

paloalto | Logout | Last Login Time: 02/27/2018 12:31:46

Tasks

Language

Here are the interfaces to zone mappings:

Palo Alto Networks GCP Deployment Manager Deployment Guide

The screenshot shows the Palo Alto Networks management console. The top navigation bar includes tabs for Dashboard, ACC, Monitor, Policies, Objects, Network, and Device. The left sidebar lists various configuration 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. The main content area is titled 'Ethernet' and displays a table of 7 interfaces.

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

At the bottom of the interface list, there are buttons for 'Delete' and 'PDF/CSV'. The footer shows the user is logged in, with a last login time of 02/27/2018 12:31:46.

In the policies tab you can review the security policies:

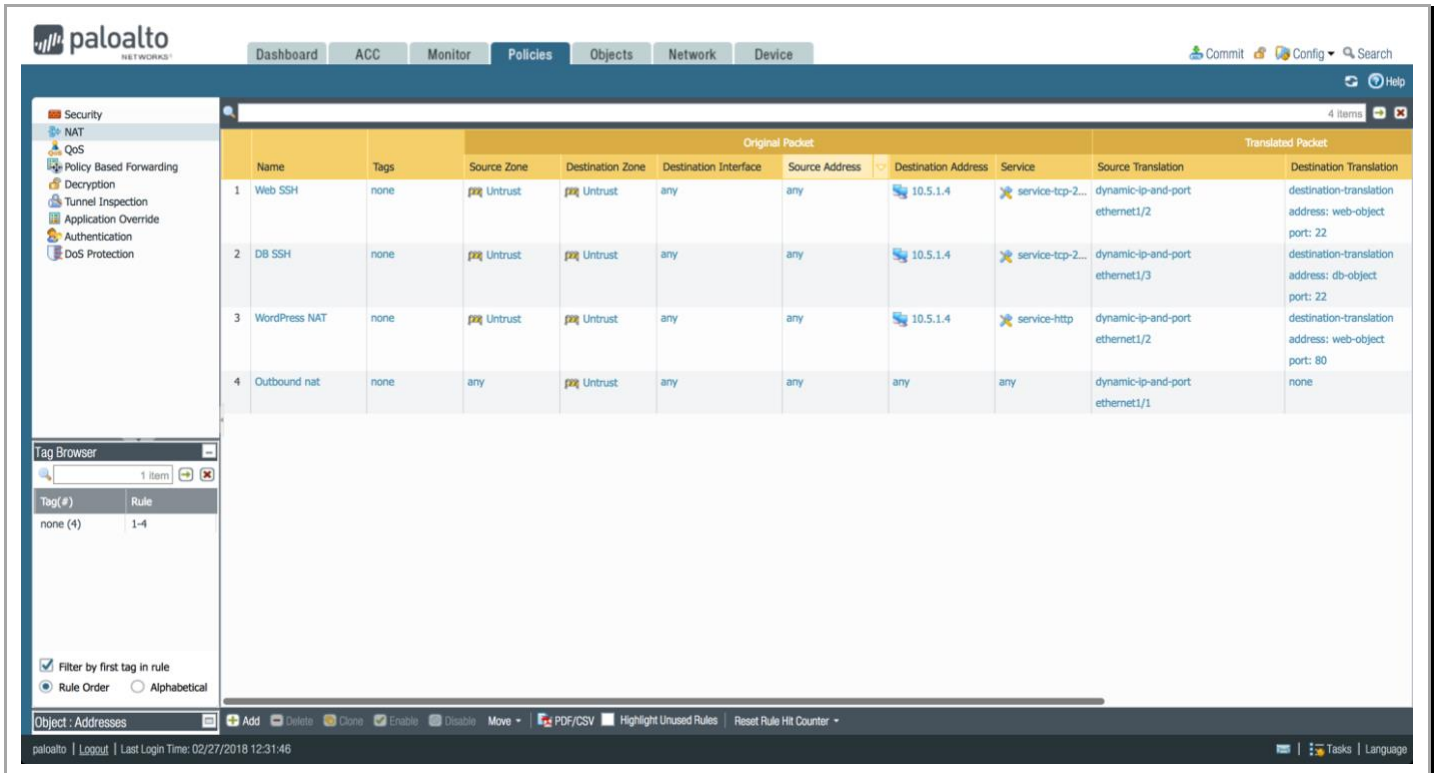
The screenshot shows the Palo Alto Networks management console with the 'Policies' tab selected. The left sidebar lists security-related categories like NAT, QoS, Policy Based Forwarding, Decryption, Tunnel Inspection, Application Override, Authentication, and DoS Protection. The main content area displays a table of 8 security rules.

| Name | Tags | Type | Zone | Address | User | HIP Profile | Zone | Address | Hit Count | Last Hit | First Hit |
|-----------------------|------|-----------|---------|------------|------|-------------|-------------|-----------|-----------|---------------------|---------------------|
| 1 SSH inbound | none | universal | Untrust | any | any | any | Web | any | - | - | - |
| 2 SSH 221-222 inbound | none | universal | Untrust | any | any | any | Db | any | 0 | - | - |
| 3 Allow all ping | none | universal | any | any | any | any | any | any | 0 | - | - |
| 4 Web browsing | none | universal | Untrust | any | any | any | Web | any | 60 | 2018-02-28 13:03:26 | 2018-02-28 13:03:26 |
| 5 Allow all outbound | none | universal | Db | any | any | any | Untrust | any | 35239 | 2018-02-28 13:12:35 | 2018-02-28 13:12:35 |
| 6 Web to DB | none | universal | any | web-object | any | any | any | db-object | 0 | - | - |
| 7 Intrazone-default | none | intrazone | any | any | any | any | (intrazone) | any | 4 | 2018-02-27 12:31:51 | 2018-02-27 12:31:51 |
| 8 Interzone-default | none | interzone | any | any | any | any | any | any | 0 | - | - |

Below the table, there is a 'Tag Browser' section with a search bar and a list of tags. The 'Filter by first tag in rule' checkbox is checked. At the bottom, there are buttons for 'Add', 'Delete', 'Clone', 'Override', 'Revert', 'Enable', 'Disable', 'Move', 'PDF/CSV', 'Highlight Unused Rules', and 'Reset Rule Hit Counter'.

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

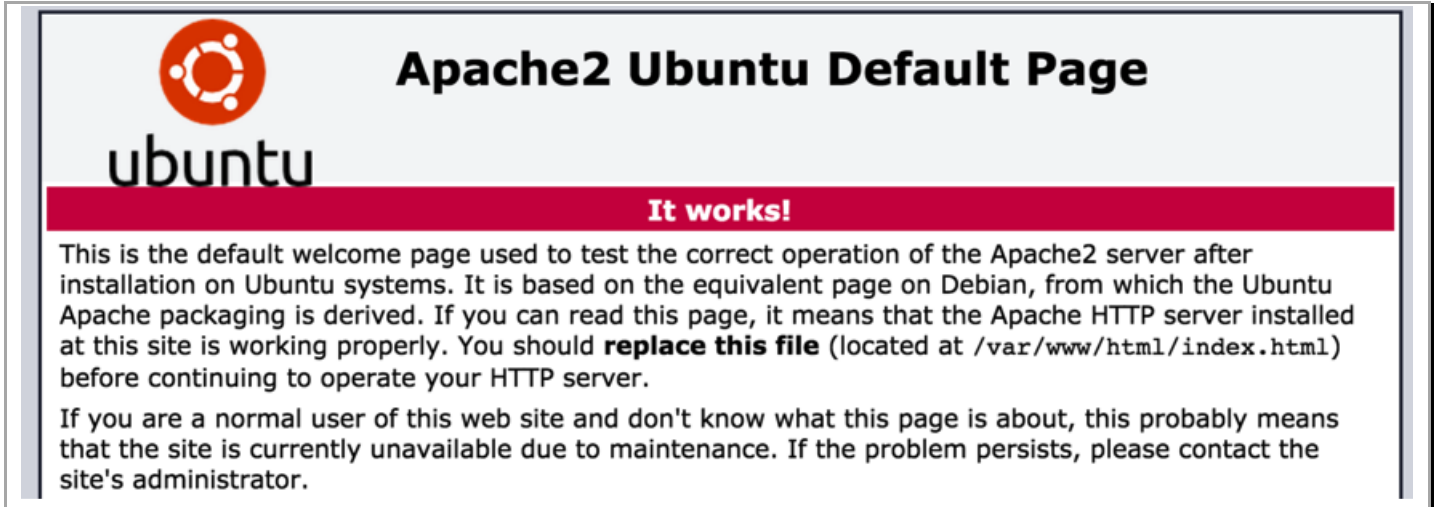
Navigate to the list of VM instances and click on the firewall. Locate the public IP address of the untrust-network interface:

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 | |

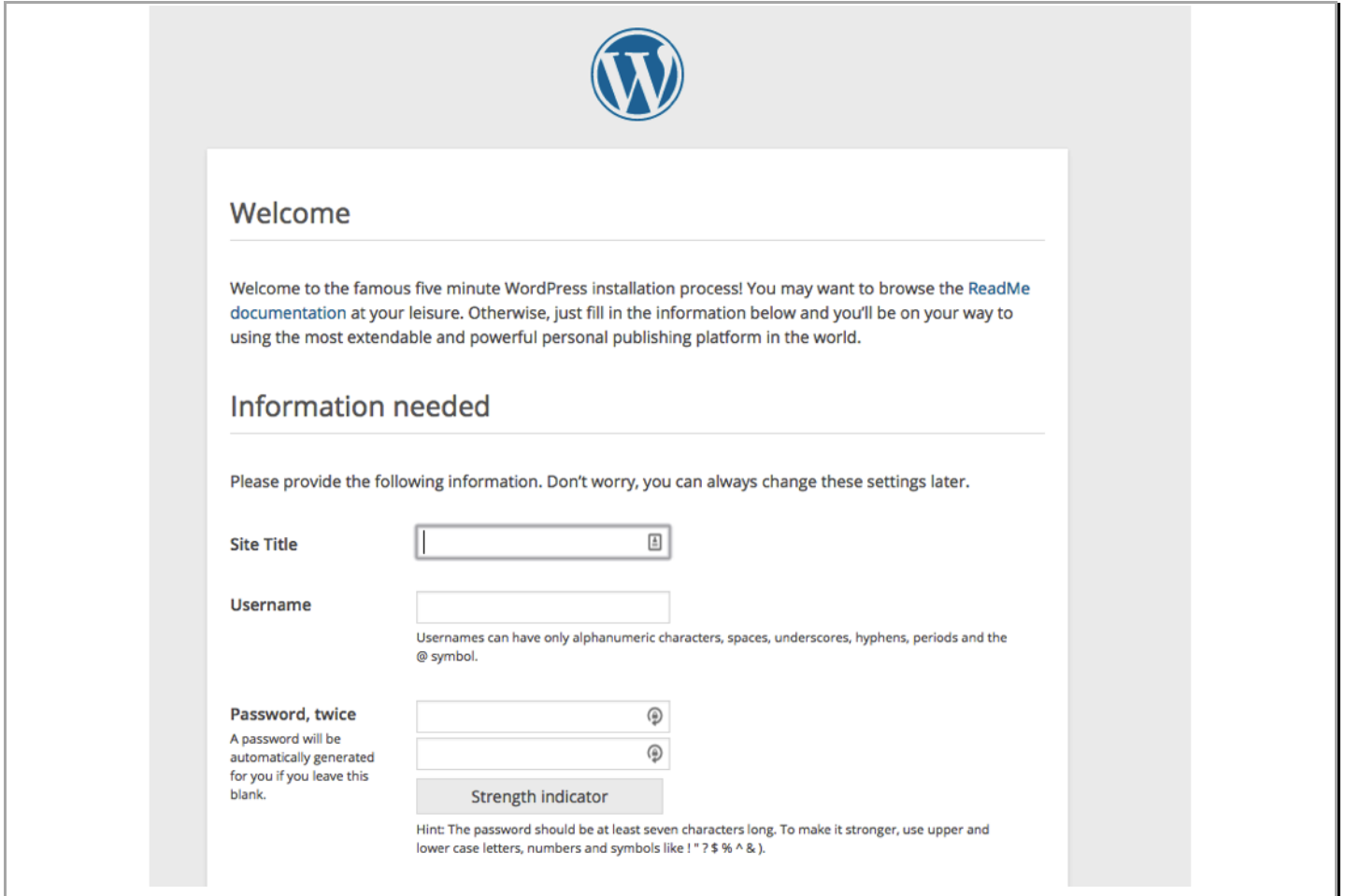
Open a new browser tab and type `http://<untrust-network-IP>/` and you should see:




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

| 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 |

The next step is to verify that East-West traffic can pass through the firewall. In the browser, head to the WordPress server (<http://<untrust-network-IP>/wordpress>). You should see the WordPress welcome screen:





Welcome

Welcome to the famous five minute WordPress installation process! You may want to browse the [ReadMe documentation](#) at your leisure. Otherwise, just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username

Username can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

Password, twice









A password will be automatically generated for you if you leave this blank.

Strength indicator

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? \$ % ^ &).

Note: You don't need to configure the new WordPress server for the purpose of this deployment. 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 the 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:

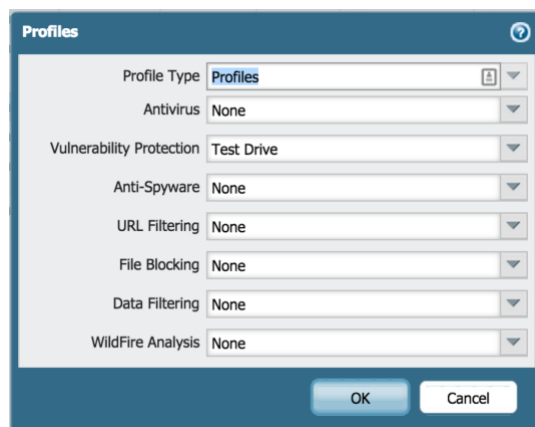
| | Receive Time | Type | From Zone | To Zone | Source | Source User | Destination | To Port | Application | Action | Rule | Session End Reason | Bytes |
|--|----------------|------|-----------|---------|----------|-------------|-------------|---------|----------------|--------|-------------------|--------------------|-------|
| | 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 | Zone | Address | User | HIP Profile | Zone | Address | Hit Count | Last Hit | First Hit | Application | Service | Action | Profile | Options |
|------|-----------|---------|---------|------|-------------|-------------|---------|-----------|---------------------|---------------------|-------------|---------------------|--------|---------|---------|
| none | universal | Untrust | any | any | any | Web | any | - | - | - | ping | application-default | Allow | none | |
| none | universal | Untrust | any | any | any | Db | any | 0 | - | - | ping | 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 | 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 | Allow | none | |
| none | universal | Db | any | any | any | Untrust | any | 87 | 2018-02-28 15:28:21 | 2018-02-28 15:28:21 | any | service-http | Allow | none | |
| none | universal | any | any | any | any | Web | any | 8 | 2018-02-28 15:30:49 | 2018-02-28 15:28:21 | mysql | application-default | Allow | none | |
| none | intrazone | any | any | any | any | (intrazone) | any | 4 | 2018-02-28 15:27:42 | 2018-02-28 15:27:42 | any | any | Allow | none | |
| none | interzone | any | any | any | any | any | any | 2 | 2018-02-28 15:33:52 | 2018-02-28 15:33:52 | 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 | Receive Time | Type | Name | From Zone | To Zone | Source address | Source User | Destination address | To Port | Application | Action | Severity |
|--------|----------------|---------------|------------------------------|-----------|---------|----------------|-------------|---------------------|---------|-------------|--------------|---------------|
| Threat | 02/28 15:37:48 | vulnerability | MySQL Login Authentication F | 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 with the template, feel free to play around with various thins. If done, cleanup as follows. From the CLI, issue the command “gcloud deployment-manager deployments delete <deployment name>” where <deployment name> is the tag used to identify the deployment (“deployment1” in this example):

```
root@2ccf4d39089e: /# gcloud deployment-manager deployments delete deployment1
The following deployments will be deleted:
- deployment1

Do you want to continue (y/N)? y

done.
Delete operation operation-1519861499988-5664e54d08821-676e434e-cb6fceb2 completed successfully.
root@2ccf4d39089e: /#
```

This should delete all the resources created via the template.

11. Conclusion

You have successfully deployed a sample template in GCP and demonstrated how the next generation VM-Series firewall can not only secure traffic inbound into your project, but within the project itself.

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.