GCP Transit VPC with Advanced Peering

Terraform Build Guide

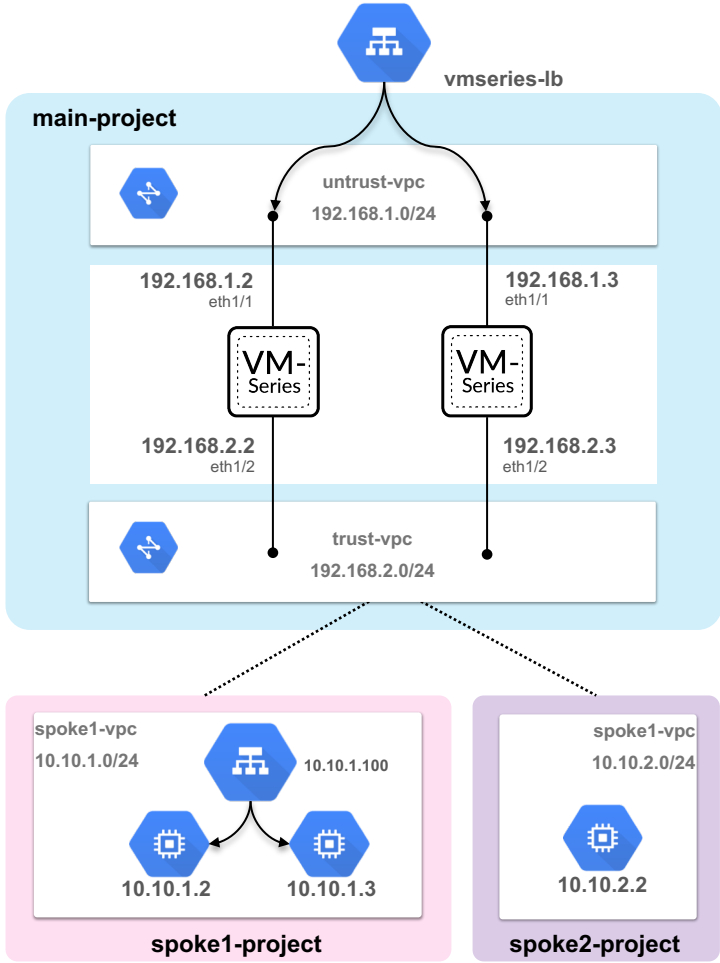
Matt McLimans, Public Cloud Consultant Engineer

SUPPORT POLICY

This 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) or sites other than our official Downloads page on https://support.paloaltonetworks.com are provided under the best effort policy.

DEPLOYMENT OVERVIEW

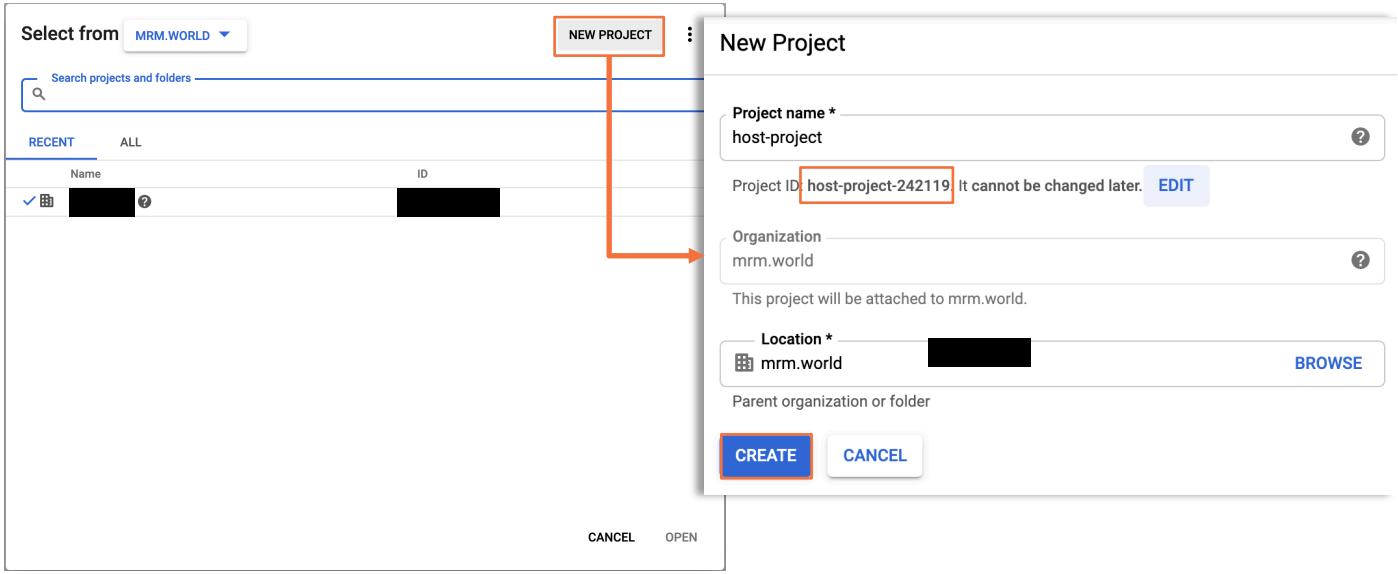
- Terraform builds 2 VM-Series firewalls and two peered VPCs.
- Spoke1 VPC has 1 internal load balancer and 2 backend Web servers (configured with Apache)
- Spoke2 VPC has 1 Linux host.
- spoke1-vpc & spoke2-vpc can be deployed into the same project as the VM-Series or in different projects.



CONFIGURE GCP API & RETRIEVE API CREDENTIALS

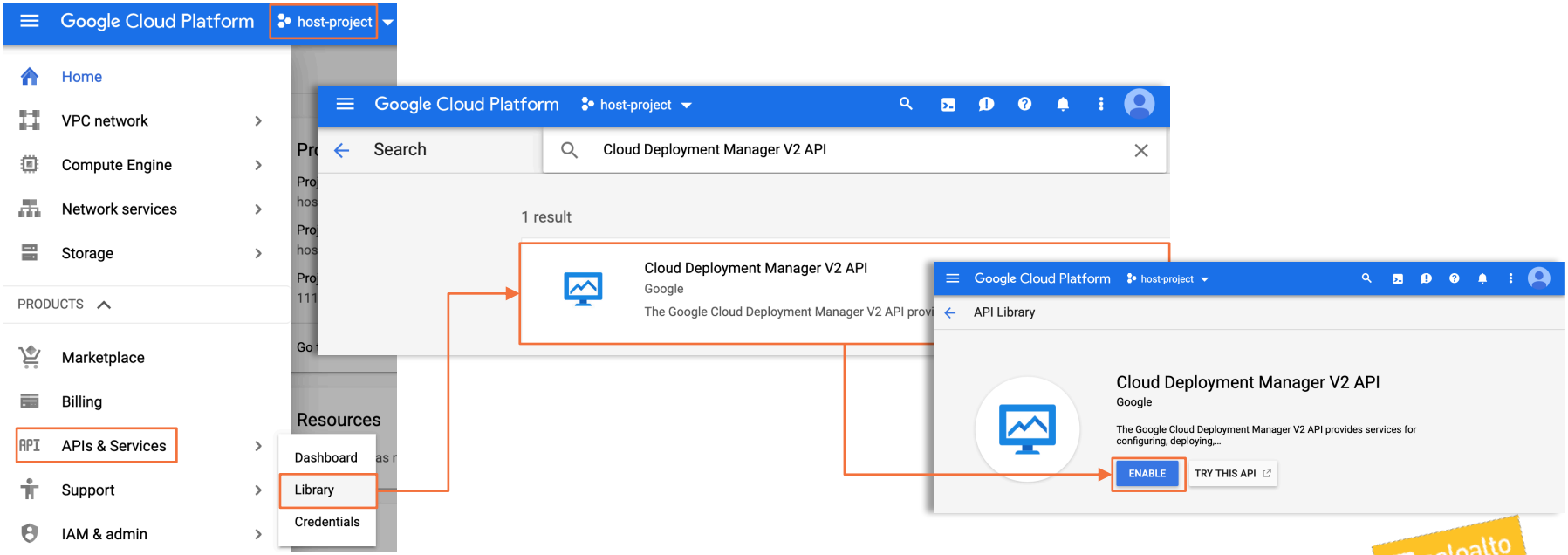
STEP 1. CREATE A PROJECT

1. Create a GCP Project
2. Record the Project ID.



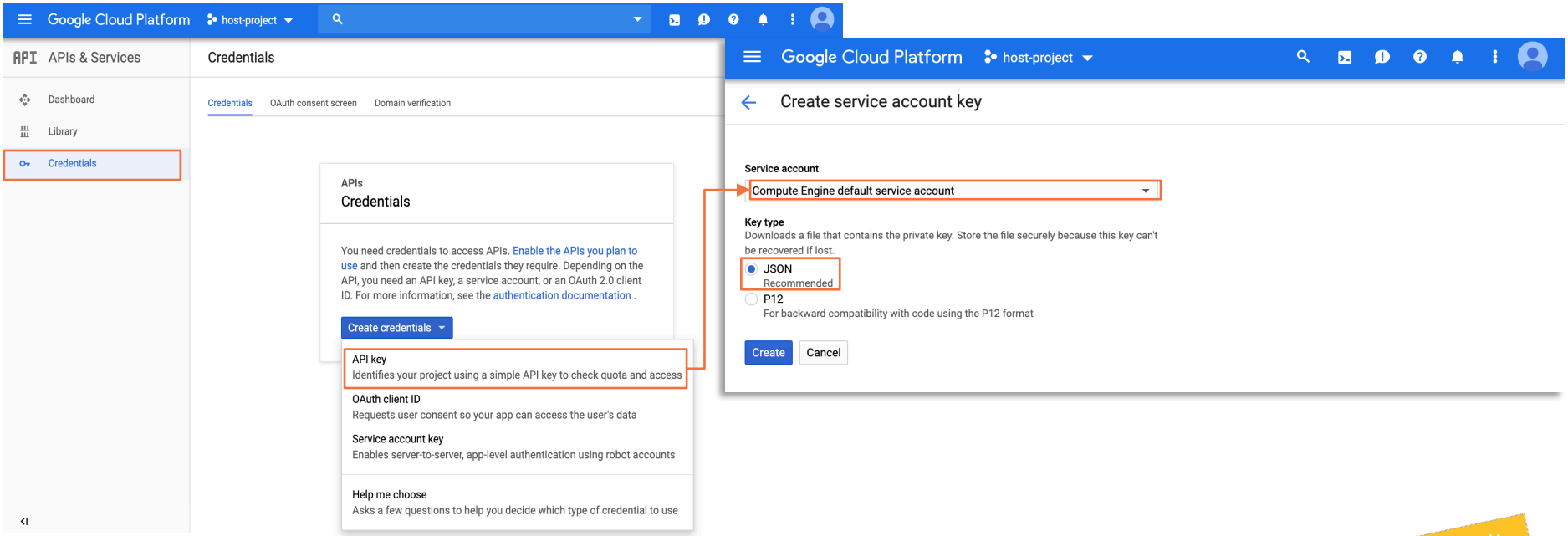
STEP 2. ENABLE GOOGLE COMPUTE API

1. Go to **API & Services → Library**
2. Search for **Cloud Deployment Manager V2 API** and click **Enable**



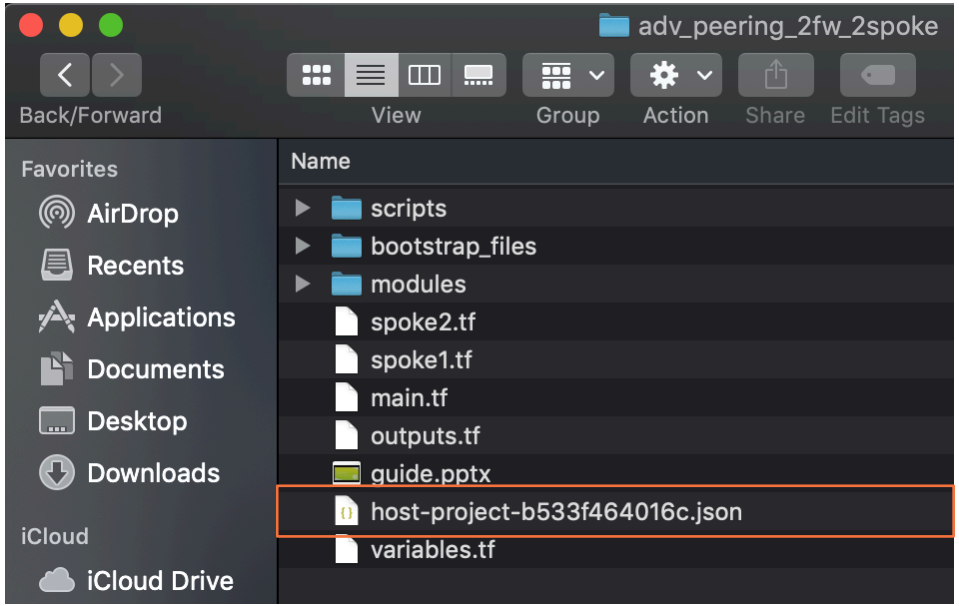
STEP 3. RETRIEVE API CREDENTIALS

1. Go to **API & Services → Credentials → Create Credentials → Service account key**
2. Select **Compute Engine default service account** and select **JSON** as the key type



STEP 4. RETRIEVE API CREDENTIALS

1. Move the downloaded key into the main directory of the Terraform build.
2. Download the Terraform build from

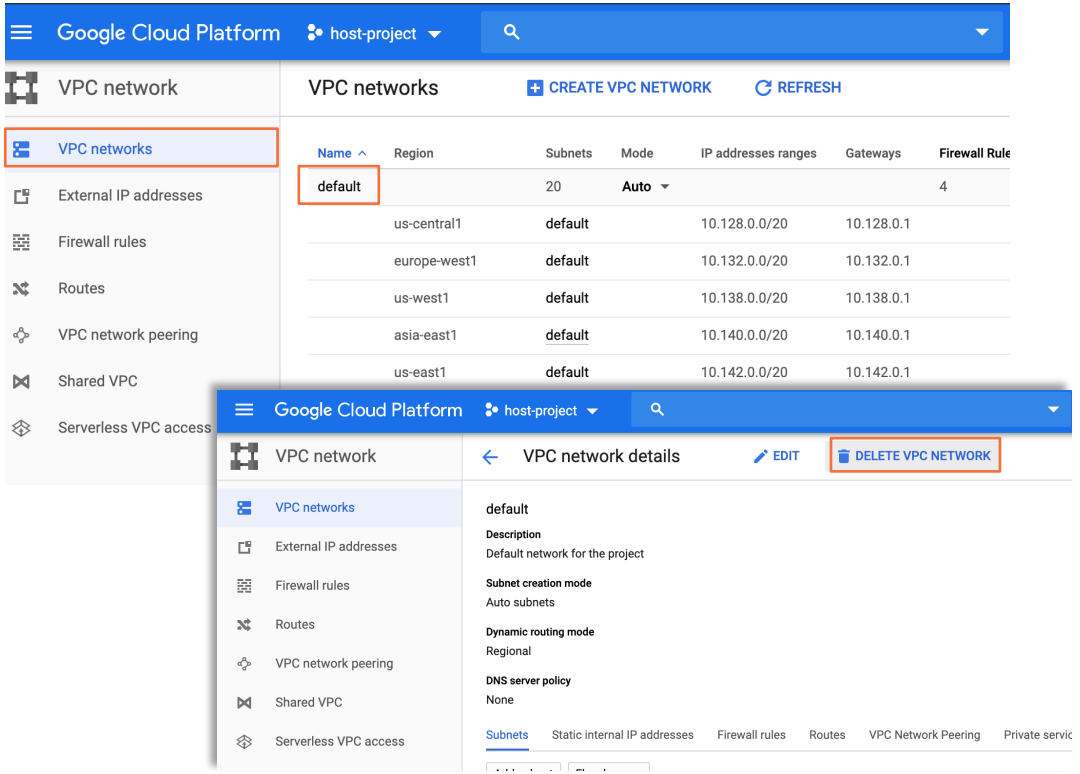


Repeat STEPS1-4 if you want Spoke1 & Spoke2 to reside in different projects than the VM-Series

DELETE DEFAULT NETWORK

Every new project has a default VPC. Each project has a soft maximum of 5 VPCs.

If you are deploying everything to the same project, make sure you either delete the default VPC in the project or ask for a quota increase.



EDIT VARIABLES.TF

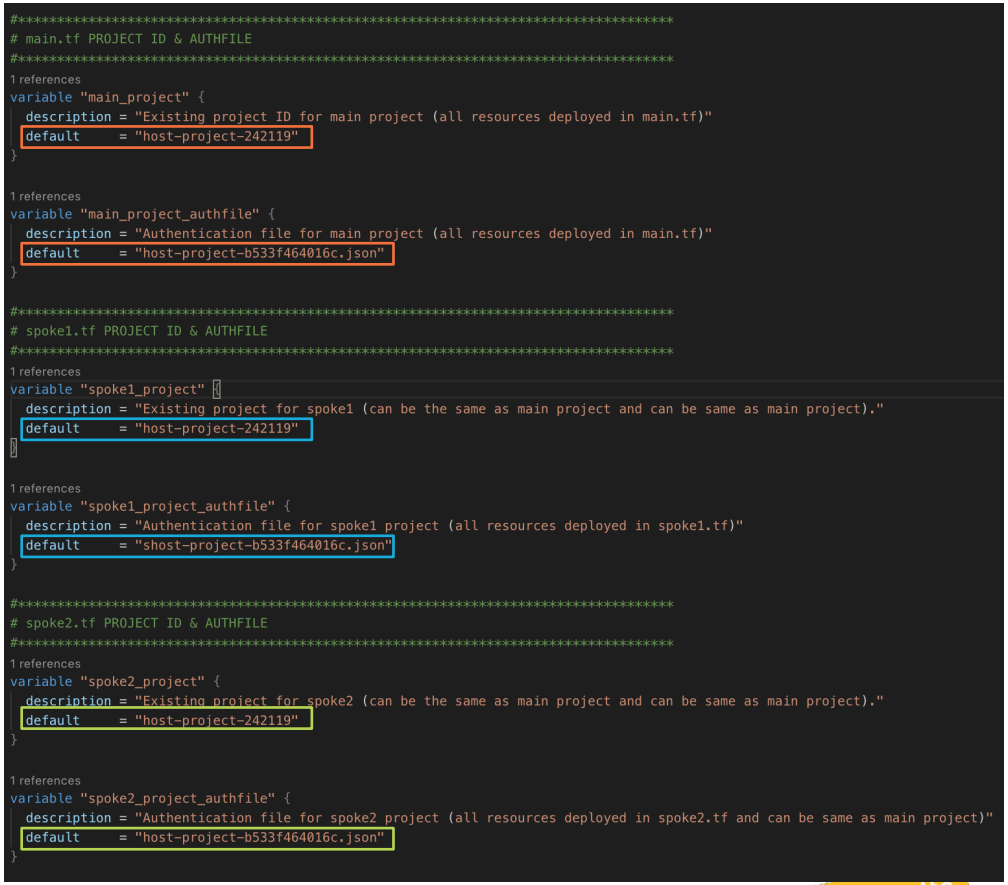
STEP 5. ADJUST VARIABLES.TF

1. Open **variables.tf** in a text editor.

2. Enter the project ID for each project in:
   - main_project
   - spoke1_project
   - spoke2_project

3. Enter the corresponding key file for the projects in:
   1. main_project_authfile
   2. spoke1_project_authfile
   3. spoke2_project_authfile

4. In this example, we are deploying everything to the same project (host-project-242119), so the project ID and authfile value will be the same for main, spoke1, and spoke2 environments.



STEP 6. SSH KEY FOR UBUNTU VM & VM-SERIES LICENSE TYPE

1. Create an SSH key for instances in the Spoke VPCs.

```
$ ssh-keygen -t rsa -f ~/.ssh/ubuntukey -C ubuntu

<enter passphrase x 2>

$ chmod 600 ~/.ssh/ubuntukey

$ cat ~/.ssh/ubuntukey.pub
```

```
#*********************************************************************************
# UBUNTU SSH KEY
#*********************************************************************************
2 references
variable "ubuntu_ssh_key" {
  default = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDBAmjFRPLEwSvNH41yU/7ouw7vBOBJzprcMssu
}
```

2. Copy CAT output and paste it as the default value for **ubuntu_ssh_key** inside **variables.tf**

3. Uncomment the **vmseries_image** to the license SKU that you want.

```
1 references
variable "vmseries_image" {
  # default = "https://www.googleapis.com/compute/v1/projects/paloaltonetworksgcp-public/global/images/vmseries-byol-814"
  default = "https://www.googleapis.com/compute/v1/projects/paloaltonetworksgcp-public/global/images/vmseries-bundle1-814"
  # default = "https://www.googleapis.com/compute/v1/projects/paloaltonetworksgcp-public/global/images/vmseries-bundle2-814"
}
```

SAVE VARIABLES.TF

RUN TERRAFORM

STEP 7. RUN TERRAFORM

1. terraform init

```
adv_peering_2fw_2spoke mmclimans$ 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.
```

2. terraform apply

```
adv_peering_2fw_2spoke mmclimans$ terraform apply
…
…
…
Plan: 49 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
```

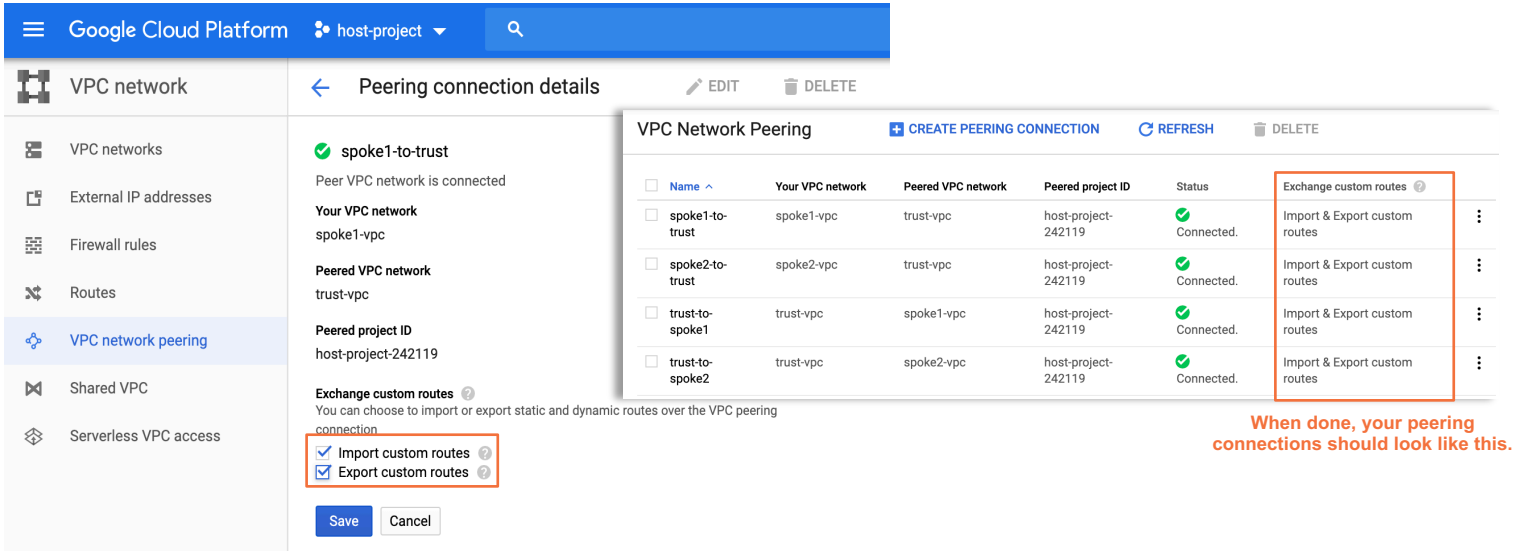
You will receive this output once the deployment has completed.

```
Apply complete! Resources: 49 added, 0 changed, 0 destroyed.

Outputs:

                              IMPORTANT!! PLEASE READ!!                                    = [
    ================================================================================,
    Before proceeding, you must enable import/export custom routes on all peering links,
    and remove the default (0.0.0.0/0) route from TRUST, SPOKE1, and SPOKE2 VPCs,
    ================================================================================
]
GLB-ADDRESS     = http://35.244.207.26
MGMT-URL-FW1    = https://35.245.168.131
MGMT-URL-FW2    = https://35.199.45.71
SSH-SPOKE1-FW1 = ssh ubuntu@35.230.184.204 -p 221 -i <INSERT KEY>
SSH-SPOKE1-FW2 = ssh ubuntu@35.194.81.140 -p 221 -i <INSERT KEY>
SSH-SPOKE2-FW1 = ssh ubuntu@35.230.184.204 -p 222 -i <INSERT KEY>
SSH-SPOKE2-FW2 = ssh ubuntu@35.194.81.140 -p 222 -i <INSERT KEY>
```

STEP 8. ENABLE IMPORT/EXPORT CUSTOM ROUTES

Go to: **VPC Network → VPC network peering**

- For **EACH PEER**, enable **Import custom routes** & **Export custom routes**



When done, your peering connections should look like this.

STEP 9. DELETE TRUST & SPOKE DEFAULT INTERNET ROUTES

Go to: **VPC Network → Routes**

- Delete **spoke1-vpc**, **spoke2-vpc**, & **trust-vpc** default route to the internet.

Routes CREATE ROUTE REFRESH DELETE

One or more VPC networks in this project has been configured to import custom routes using VPC Network Peering. Any imported custom dynamic routes are omitted from this list, and some route conflicts might complete list of imported custom routes, and the routing order for information about how GCP resolves conflicts.

Filter resources

Name	Description	Destination IP range	Priority	Instance tags	Next hop	Network
default-to-vmseries01		0.0.0.0/0	100	None	Instance vmseries01 (zone us-east4-a)	trust-vpc
default-to-vmseries02		0.0.0.0/0	100	None	Instance vmseries02 (zone us-east4-b)	trust-vpc
peering-route-d475ceb4173e9483	Auto generated route via peering [spoke1-to-trust].	0.0.0.0/0	100	None	Network peering spoke1-to-trust	spoke1-vpc
peering-route-e13564b53e9f2e09	Auto generated route via peering [spoke1-to-trust].	0.0.0.0/0	100	None	Network peering spoke1-to-trust	spoke1-vpc
peering-route-15ad69d2a49fb174	Auto generated route via peering [spoke2-to-trust].	0.0.0.0/0	100	None	Network peering spoke2-to-trust	spoke2-vpc
peering-route-f6da3dad37748741	Auto generated route via peering [spoke2-to-trust].	0.0.0.0/0	100	None	Network peering spoke2-to-trust	spoke2-vpc
default-route-20628ec1d28a95c6	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	mgmt-vpc
default-route-df8a810aaf3353d8	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	spoke1-vpc
default-route-28c163fa61f105cd	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	spoke2-vpc
default-route-3b1dd6054bde92ae	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	trust-vpc
default-route-f6ad36676090d8c6	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	untrust-vpc
peering-route-4671025c54e80b82	Auto generated route via peering [trust-to-spoke1].	10.10.1.0/24	1000	None	Network peering trust-to-spoke1	trust-vpc
default-route-54d2fe18ad71c197	Default local route to the subnetwork 10.10.1.0/24.	10.10.1.0/24	1000	None	Virtual network spoke1-vpc	spoke1-vpc

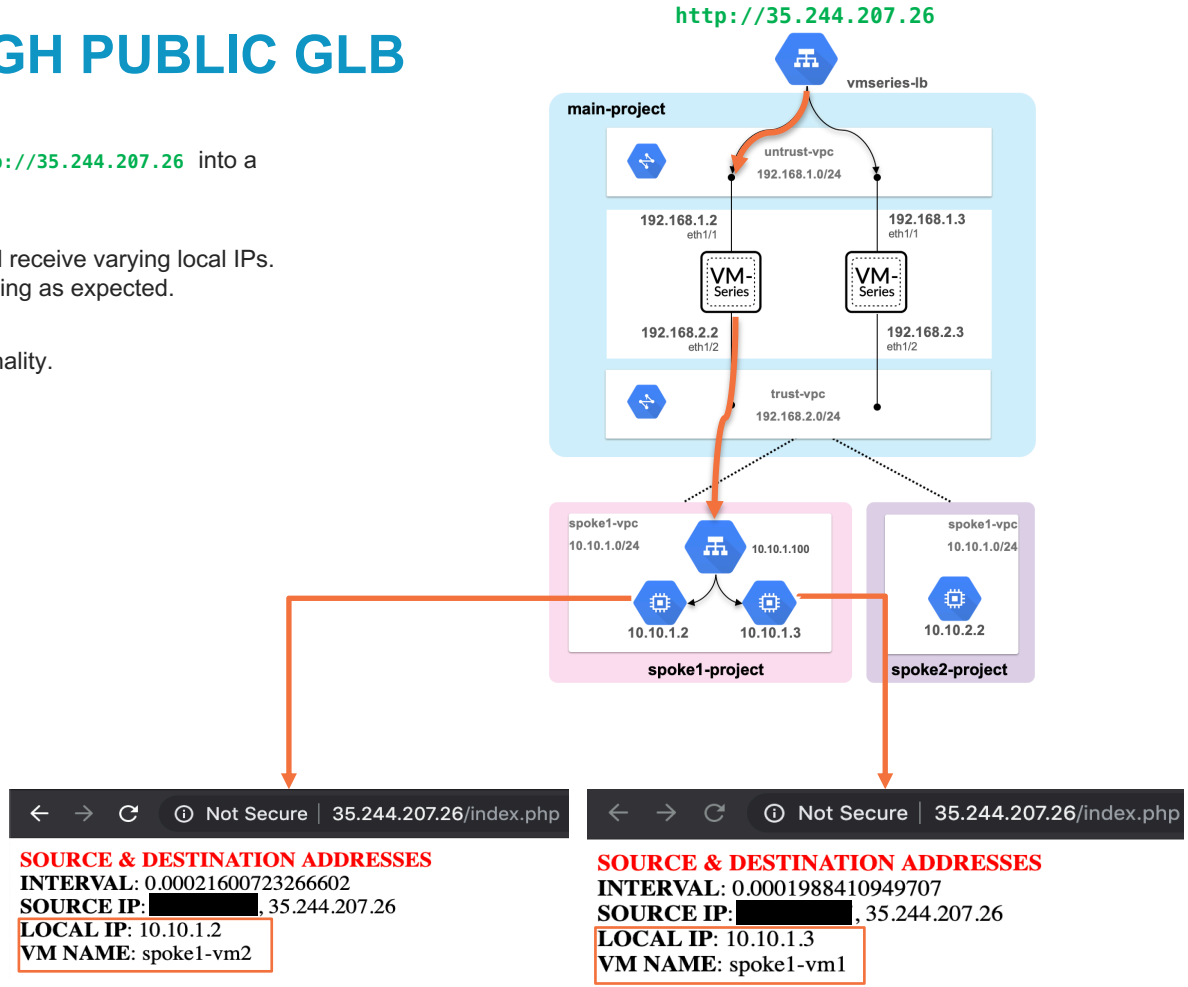
TEST TRAFFIC FLOWS

TEST INBOUND THROUGH PUBLIC GLB

1. From the Terraform output, copy `GLB-ADDRESS = http://35.244.207.26` into a web browser.

2. Once the page resolves, on each refresh you should receive varying local IPs. This indicates that ingress load balancing is functioning as expected.

3. View the firewall logs to view load balancing functionality.



TEST OUTBOUND

```
GLB-ADDRESS     = http://35.244.207.26
MGMT-URL-FW1    = https://35.245.168.131
MGMT-URL-FW2    = https://35.199.45.71
SSH-SPOKE1-FW1 = ssh ubuntu@35.230.184.204 -p 221 -i <INSERT KEY>
SSH-SPOKE1-FW2 = ssh ubuntu@35.194.81.140 -p 221 -i <INSERT KEY>
SSH-SPOKE2-FW1 = ssh ubuntu@35.230.184.204 -p 222 -i <INSERT KEY>
SSH-SPOKE2-FW2 = ssh ubuntu@35.194.81.140 -p 222 -i <INSERT KEY>
```

1. SSH through either firewall to a backend server.
2. Test egress connectivity (i.e. `sudo apt-get update`).
3. View the firewall logs. The egress request should flow through both firewalls since we are leveraging ECMP.

FW1 Egress Traffic

Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action
05/29 17:50:09	end	trust-zone	untrust-zone	10.10.1.3		91.189.88.161	80	apt-get	allow
05/29 17:50:08	end	trust-zone	untrust-zone	10.10.1.2		91.189.88.162	80	apt-get	allow
05/29 17:49:52	start	trust-zone	untrust-zone	10.10.1.3		91.189.88.161	80	apt-get	allow
05/29 17:49:52	start	trust-zone	untrust-zone	10.10.1.3		91.189.88.161	80	web-browsing	allow

FW2 Egress Traffic

(zone.src eq trust-zone)

Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application
05/29 17:50:08	end	trust-zone	untrust-zone	10.10.1.2		91.189.88.162	80	apt-get
05/29 17:50:07	end	trust-zone	untrust-zone	10.10.1.3		91.189.88.161	80	apt-get
05/29 17:49:53	start	trust-zone	untrust-zone	10.10.1.2		91.189.88.162	80	apt-get
05/29 17:49:53	start	trust-zone	untrust-zone	10.10.1.2		91.189.88.162	80	web-browsing