

GCP Transit VPC with Advanced Peering and East/West traffic using Load Balanced VM Series firewalls.

Terraform Build Guide



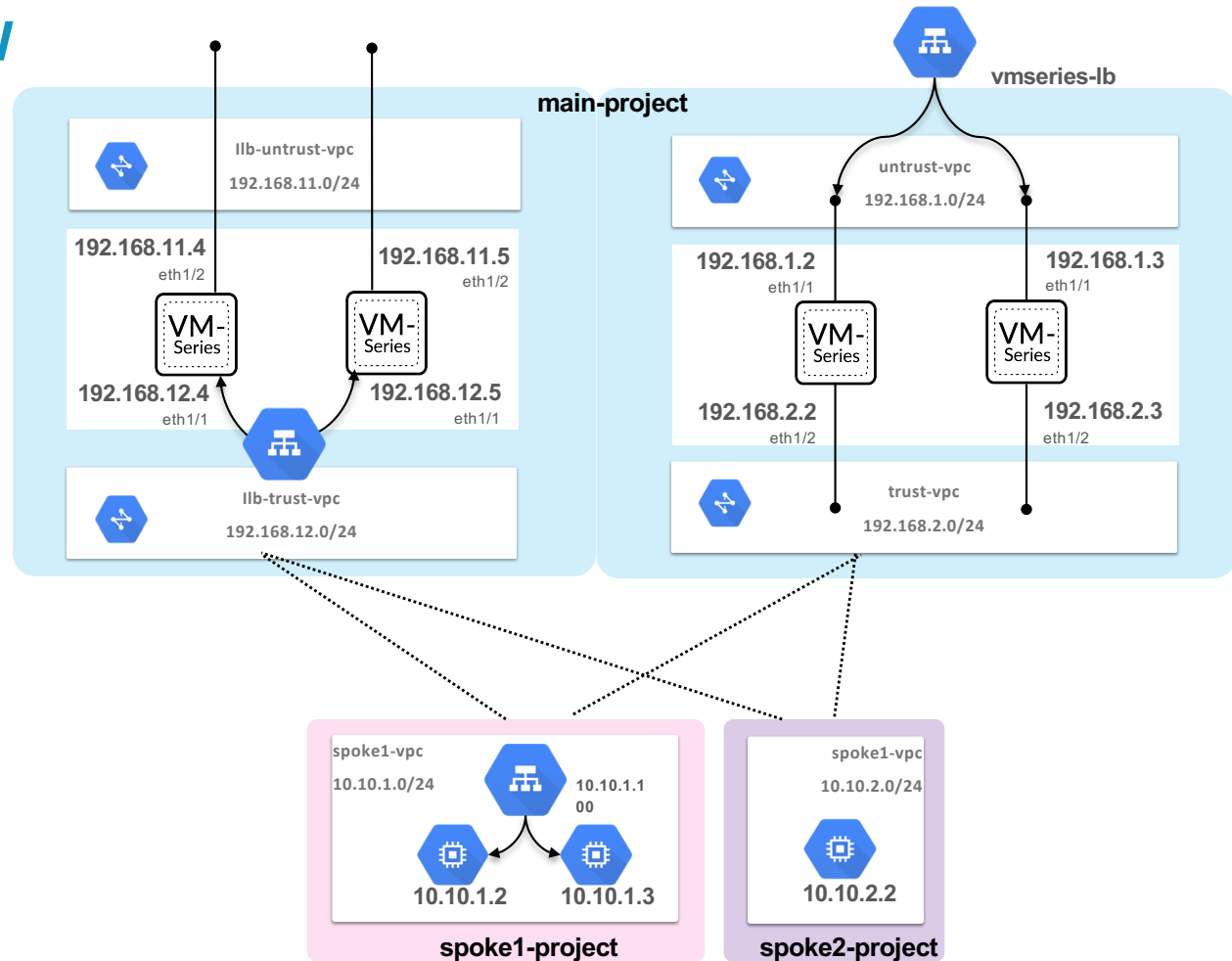
SUPPORT POLICY

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 4 VM-Series firewalls and two peered VPCs. 2 VM-Series provide N/S resiliency for inbound traffic while another 2 provide E/W and outbound resiliency.
- 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.

NOTE: There are a few steps that need to be completed after the Terraform deployment. This is because when this was created, the terraform provider did not yet support the functionality.



CONFIGURE GCP API & RETRIEVE API CREDENTIALS

STEP 1. CREATE A PROJECT

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

Select from **MRM.WORLD**

NEW PROJECT

Search projects and folders

RECENT ALL

	Name	ID
✓	[REDACTED]	[REDACTED]

CANCEL OPEN

New Project

Project name *
host-project

Project ID: **host-project-242119** It cannot be changed later. [EDIT](#)

Organization
mrm.world

This project will be attached to mrm.world.

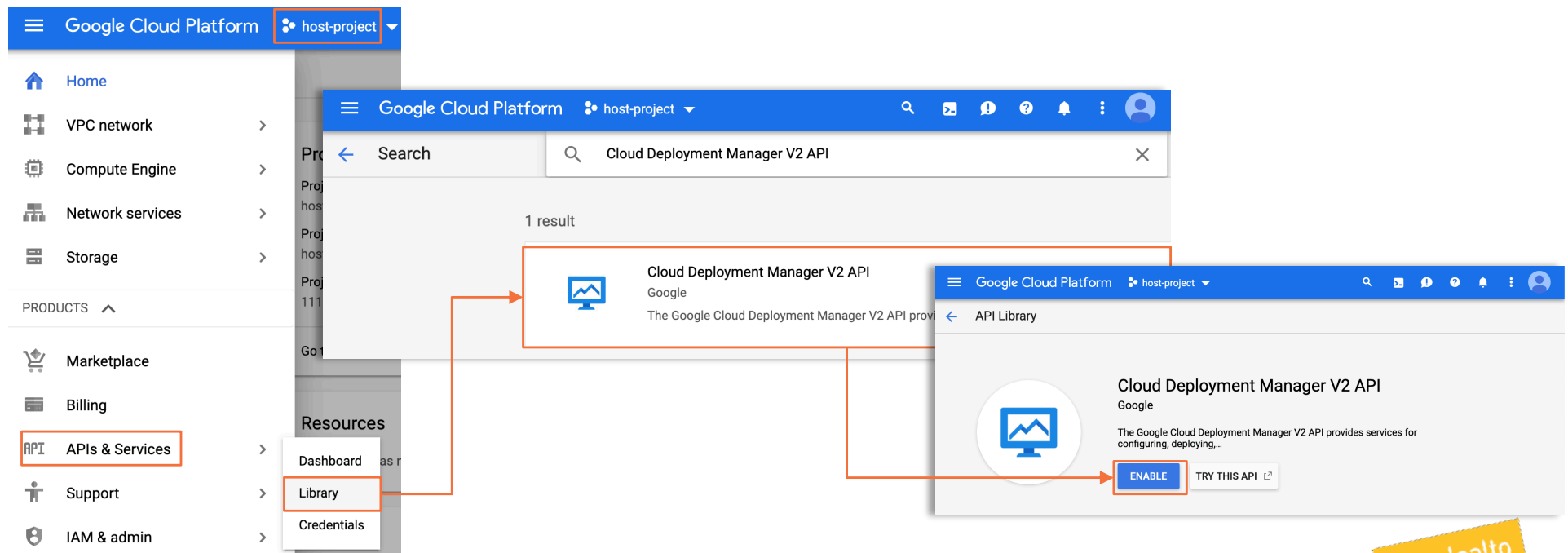
Location *
mrm.world [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

STEP 2. ENABLE GOOGLE COMPUTE API

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



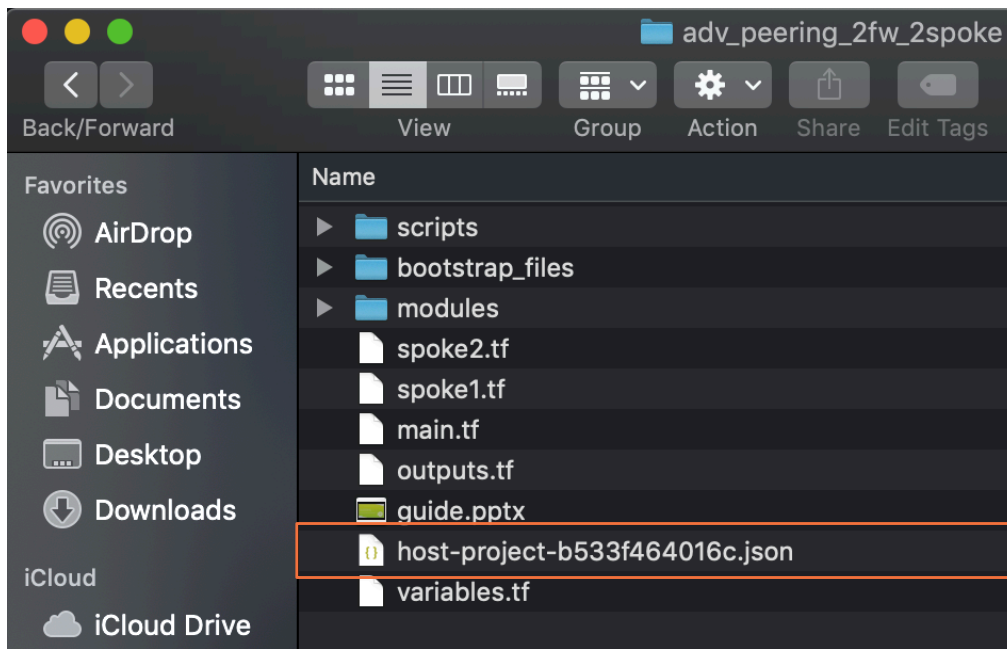
STEP 3. RETRIEVE API CREDENTIALS

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

The image shows two screenshots from the Google Cloud Platform interface. The left screenshot shows the 'APIs & Services' section with 'Credentials' selected in the left sidebar. The 'Create credentials' dropdown menu is open, showing options: 'API key', 'OAuth client ID', 'Service account key', and 'Help me choose'. The 'Service account key' option is highlighted with a red box. The right screenshot shows the 'Create service account key' dialog. The 'Service account' dropdown is set to 'Compute Engine default service account' (highlighted with a red box). The 'Key type' section has 'JSON' selected (highlighted with a red box) and 'P12' as an alternative. The 'Create' button is visible at the bottom.

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.

The screenshot displays the Google Cloud Platform interface for managing VPC networks. The top navigation bar shows 'Google Cloud Platform' and 'host-project'. The left sidebar lists various network services, with 'VPC networks' highlighted. The main content area shows a table of VPC networks. The 'default' network is highlighted, and its details are shown in a modal window. The 'DELETE VPC NETWORK' button is highlighted in the modal window.

Name	Region	Subnets	Mode	IP addresses ranges	Gateways	Firewall Rule
default		20	Auto			4
	us-central1	default		10.128.0.0/20	10.128.0.1	
	europa-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	

VPC network details

default

Description
Default network for the project

Subnet creation mode
Auto subnets

Dynamic routing mode
Regional

DNS server policy
None

Subnets | Static internal IP addresses | Firewall rules | Routes | VPC Network Peering | Private service



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.

```
#####
# main.tf PROJECT ID & AUTHFILE
#####
1 references
variable "main_project" {
  description = "Existing project ID for main project (all resources deployed in main.tf)"
  default     = "host-project-242119"
}

1 references
variable "main_project_authfile" {
  description = "Authentication file for main project (all resources deployed in main.tf)"
  default     = "host-project-b533f464016c.json"
}

#####
# spoke1.tf PROJECT ID & AUTHFILE
#####
1 references
variable "spoke1_project" {
  description = "Existing project for spoke1 (can be the same as main project and can be same as main project)."
  default     = "host-project-242119"
}

1 references
variable "spoke1_project_authfile" {
  description = "Authentication file for spoke1 project (all resources deployed in spoke1.tf)"
  default     = "host-project-b533f464016c.json"
}

#####
# spoke2.tf PROJECT ID & AUTHFILE
#####
1 references
variable "spoke2_project" {
  description = "Existing project for spoke2 (can be the same as main project and can be same as main project)."
  default     = "host-project-242119"
}

1 references
variable "spoke2_project_authfile" {
  description = "Authentication file for spoke2 project (all resources deployed in spoke2.tf and can be same as main project)"
  default     = "host-project-b533f464016c.json"
}
```

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 AAAAB3NzaC1yc2EAAAADAQABAAQDBAmjFRPLEwSvNH41yU/7ouw7vB0BJzprcMssu
}
```

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: 74 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, There is also a need to create a default route in the ilb-trust-subnet pointing to, the internal load balancer,

```
]
GLB-ADDRESS      = http://34.102.167.223
MGMT-URL-FW1     = https://35.239.219.110
MGMT-URL-FW2     = https://35.232.77.73
MGMT-URL-ILB-FW3 = https://34.67.53.20
MGMT-URL-ILB-FW4 = https://34.68.219.215
SSH-SPOKE1-VM1-FW1 = ssh ubuntu@35.224.112.224 -p 221 -i <INSERT KEY>
SSH-SPOKE1-VM1-FW2 = ssh ubuntu@34.68.152.104 -p 221 -i <INSERT KEY>
SSH-SPOKE1-VM2-FW1 = ssh ubuntu@35.224.112.224 -p 222 -i <INSERT KEY>
SSH-SPOKE1-VM2-FW2 = ssh ubuntu@34.68.152.104 -p 222 -i <INSERT KEY>
SSH-SPOKE2-FW1    = ssh ubuntu@35.224.112.224 -p 223 -i <INSERT KEY>
SSH-SPOKE2-FW2    = ssh ubuntu@34.68.152.104 -p 223 -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**

Google Cloud Platformhost-project

VPC network

VPC networks
External IP addresses
Firewall rules
Routes
VPC network peering
Shared VPC
Serverless VPC access

Peering connection details

EDITDELETE

spoke1-to-trust
Peer VPC network is connected
Your VPC network
spoke1-vpc
Peered VPC network
trust-vpc
Peered project ID
host-project-242119
Exchange custom routes
You can choose to import or export static and dynamic routes over the VPC peering connection

☒ Import custom routes
☒ Export custom routes

SaveCancel

VPC network peeringCREATE PEERING CONNECTIONREFRESHDELETE

Filter resources

Columns

Name	Your VPC network	Peered VPC network	Peered project ID	Status	Exchange custom routes
ilb-trust-to-spoke1	ilb-trust-vpc	spoke1-vpc	djs-ilb-2019	Active	Import & Export custom routes
ilb-trust-to-spoke2	ilb-trust-vpc	spoke2-vpc	djs-ilb-2019	Active	Import & Export custom routes
spoke1-to-ilb-trust	spoke1-vpc	ilb-trust-vpc	djs-ilb-2019	Active	Import & Export custom routes
spoke1-to-trust	spoke1-vpc	trust-vpc	djs-ilb-2019	Active	Import & Export custom routes
spoke2-to-ilb-trust	spoke2-vpc	ilb-trust-vpc	djs-ilb-2019	Active	Import & Export custom routes
spoke2-to-trust	spoke2-vpc	trust-vpc	djs-ilb-2019	Active	Import & Export custom routes
trust-to-spoke1	trust-vpc	spoke1-vpc	djs-ilb-2019	Active	Import & Export custom routes
trust-to-spoke2	trust-vpc	spoke2-vpc	djs-ilb-2019	Active	Import & Export custom routes

NOTE: If you see the following error, you are clicking too fast. **SLOW DOWN**. Wait a few seconds and click Save again.

When done, your peering connections should look like this.

Failed to update peering trust-to-spoke1: Request contains an invalid argument.

15 | © 2019 Palo Alto Networks, Inc. All Rights Reserved.

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							
All Dynamic Peering							
<p>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 to the VPC network peering details page for the complete list of imported custom routes, and the routing order for information about how GCP resolves conflicts.</p>							
<input type="text" value="Filter resources"/>							
<input type="checkbox"/>	Name	Description	Destination IP range	Priority	Instance tags	Next hop	Network
<input type="checkbox"/>	default-route-37b1d99771c898c2	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	lib-mgmt-vpc
<input type="checkbox"/>	default-route-5c18d59f80be30d0	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	lib-untrust-vpc
<input type="checkbox"/>	default-route-7008cfe9c56ff72f	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	untrust-vpc
<input type="checkbox"/>	default-route-8e638f7222facf44	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	mgmt-vpc
<input checked="" type="checkbox"/>	default-route-90a7e45d0600c046	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	trust-vpc
<input checked="" type="checkbox"/>	default-route-ab494b3d8bb9755e	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	spoke1-vpc
<input checked="" type="checkbox"/>	default-route-faa12b2e21141daf	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	spoke2-vpc
<input type="checkbox"/>	default-route-faf6ddcc8c0c542	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	default
<input checked="" type="checkbox"/>	default-route-fc3e9cbf619733e9	Default route to the Internet.	0.0.0.0/0	1000	None	Default internet gateway	lib-trust-vpc
<input type="checkbox"/>	default-route-48e78f8640b3186	Default local route to the subnetwork 10.10.1.0/24.	10.10.1.0/24	1000	None	Virtual network spoke1-vpc	spoke1-vpc

NOTE: This is required because GCP Peering has a restriction that when a dynamic route overlaps with a subnet route in a peer network. For dynamic routes, the destination ranges that overlap with a subnet route from the peer network are silently dropped. GCP uses the subnet route. <https://cloud.google.com/vpc/docs/vpc-peering>

STEP 10. CREATE A DEFAULT ROUTE IN THE ILB-TRUST-VPC SUBNET

Go to: **VPC Network → Routes**

- Click Create Route and enter the following info to create a default route that will be propagated to the spoke subnets:

← Create a route

Name *
ilb

Description

Network *
ilb-trust-vpc
The network that the route applies to

Destination IP range *
0.0.0.0/0

Priority *
33

Instance tags

Next hop
Specify a forwarding rule of internal TCP/UDP load balancer

Forwarding rule name *
ilbnh-all

CREATE CANCEL

Equivalent [REST](#) or [command line](#)

- Click Create

TEST TRAFFIC FLOWS

POST DEPLOYMENT VALIDATION

1. After the deployment is complete navigate to Network services > Load balancing and click the spoke1-intlb-backend-0
2. If the targets are showing unhealthy as displayed in the diagram on the right, Navigate to Compute Engine > VM Instances and reset the Spoke 1 VMs.
3. Otherwise, it is possible to login to the VMs using the login information from the deployment and execute the following:

```
sudo apt-get update
sudo apt-get install -y php
sudo apt-get install -y apache2
sudo apt-get install -y php7
sudo apt-get install -y libapache2-mod-php7
sudo wget -O /var/www/html/index.php
https://raw.githubusercontent.com/wwce/terraform/master/gcp/adv_peering_2fw_2spoke/scripts/showheader.s.php
sudo systemctl restart apache2
```

```
Apply complete! Resources: 74 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,
There is also a need to create a default route in the ilb-trust-subnet pointing to,
the internal load balancer,

]
GLB-ADDRESS = http://34.102.167.223
MGMT-URL-FW1 = https://35.239.219.110
MGMT-URL-FW2 = https://35.226.119.108
MGMT-URL-ILB-FW3 = https://34.68.212.223
MGMT-URL-ILB-FW4 = https://34.69.41.227
SSH-SPOKE1-VM1-FW2 = ssh ubuntu@35.232.77.73 -p 221 -i <INSERT KEY>
SSH-SPOKE1-VM2-FW1 = ssh ubuntu@34.68.152.104 -p 222 -i <INSERT KEY>
SSH-SPOKE2-FW1 = ssh ubuntu@34.68.152.104 -p 223 -i <INSERT KEY>
SSH-SPOKE2-FW2 = ssh ubuntu@35.232.77.73 -p 223 -i <INSERT KEY>
```

Google Cloud Platform djs-ilb-2019

Network services Load balancer details EDIT DELETE

Load balancing

Cloud DNS

Cloud CDN

Cloud NAT

Traffic Director

spoke1-intlb-backend-0

Frontend

Protocol	Subnetwork	IP:Ports	DNS name
TCP	spoke1-subnet (10.10.1.0/24)	10.10.1.100:80,443	

Backend

Region: us-central1 Network: spoke1-vpc Endpoint protocol: TCP Session affinity: None Health check: spoke1-intlb-ch

Advanced configurations

Instance group	Zone	Health	Autoscaling	Use as failover group
spoke1-intlb-group	us-central1-a	0 / 2	Off	No

Compute Engine VM instances CREATE INSTANCE IMPORT VM REFRESH START STOP RESET

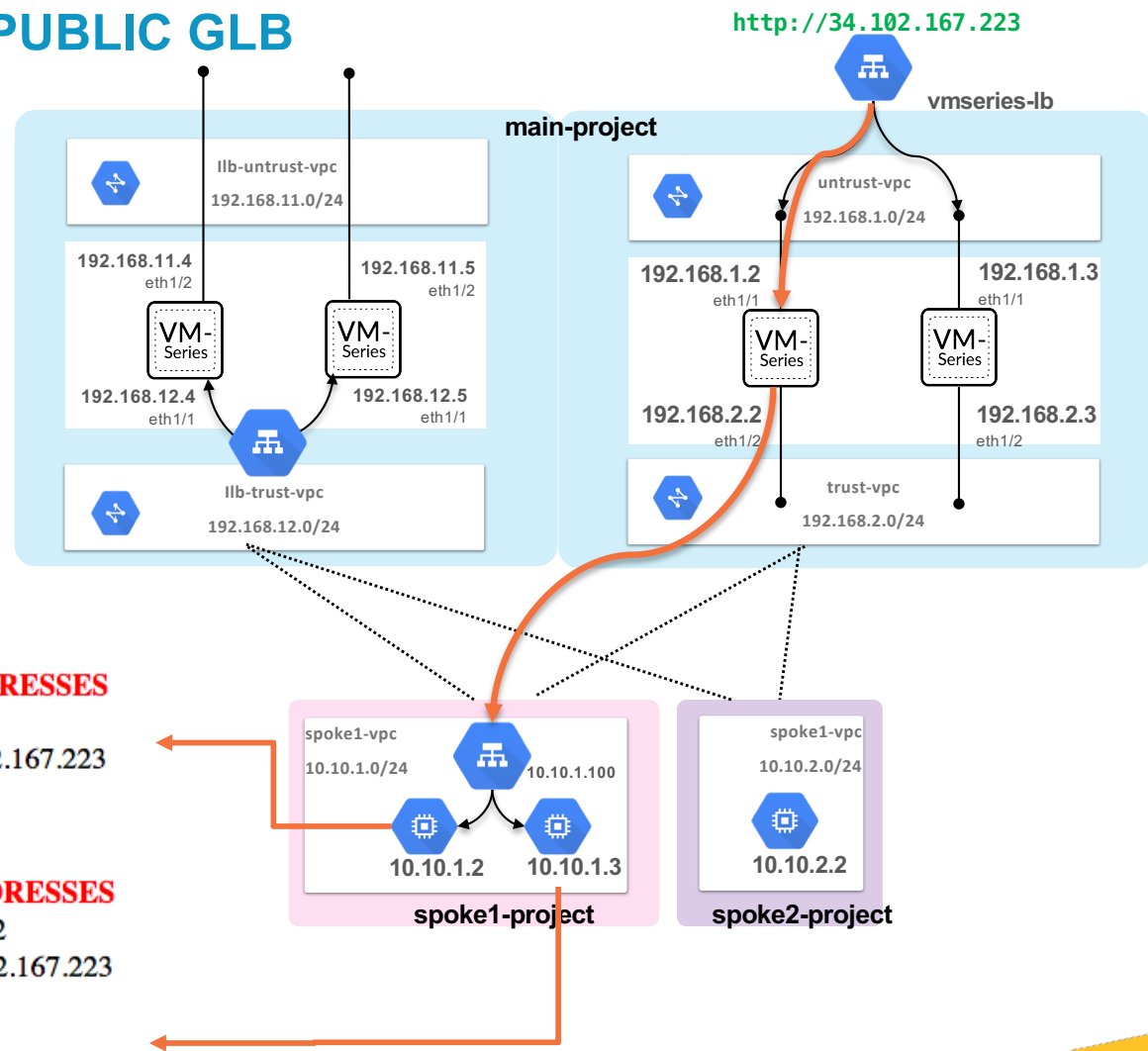
Filter VM instances Columns

Name	Zone	Recommendation	In use by	Internal IP	External IP	Connect
spoke1-vm1	us-central1-a	spoke1-intlb-group	10.10.1.3 (nic0)	None	SSH	
spoke1-vm2	us-central1-a	spoke1-intlb-group	10.10.1.2 (nic0)	None	SSH	
spoke2-vm1	us-central1-a		10.10.2.2 (nic0)	None	SSH	
vmseries01	us-central1-a	vmseries01-ig	192.168.1.2 (nic0)	104.198.75.182	SSH	
vmseries02	us-central1-b	vmseries02-ig	192.168.1.3 (nic0)	34.67.60.161	SSH	
vmseries03	us-central1-a	vmseries03-ig	192.168.12.4 (nic0)	34.67.246.148	SSH	
vmseries04	us-central1-b	vmseries04-ig	192.168.12.5 (nic0)	35.225.0.83	SSH	

Resetting VM succeeded

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 on firewall VMs 1 and 2 to view load balancing functionality.



SOURCE & DESTINATION ADDRESSES

INTERVAL: 0.0001981258392334

SOURCE IP: [REDACTED], 34.102.167.223

LOCAL IP: 10.10.1.3

VM NAME: spoke1-vm2

SOURCE & DESTINATION ADDRESSES

INTERVAL: 0.00021600723266602

SOURCE IP: [REDACTED], 34.102.167.223

LOCAL IP: 10.10.1.2

VM NAME: spoke1-vm1

TEST EAST/WEST THROUGH INTERNAL LOAD BALANCER

1. From the Terraform output, copy `SSH-SPOKE2-FW1 = ssh ubuntu@35.224.112.224 -p 223 -i <INSERT KEY>` command prompt and insert the path to the SSH key that was used in the variables.tf file.
2. Once logged into the VM perform a curl to one of the web servers in spoke1 and check the firewall logs on the firewall VMs 3 and 4.

```
SJOMACC0N5JHD4:~ dspears$ ssh ubuntu@35.224.112.224 -p 223 -i ~/.ssh/davejspears
The authenticity of host '[35.224.112.224]:223 ([35.224.112.224]:223)' can't be established.
ECDSA key fingerprint is SHA256:glc9LoNL8EBZ4camESwHZci0efW9gRVXPCtJTBn67w.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[35.224.112.224]:223' (ECDSA) to the list of known hosts.
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1042-gcp x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

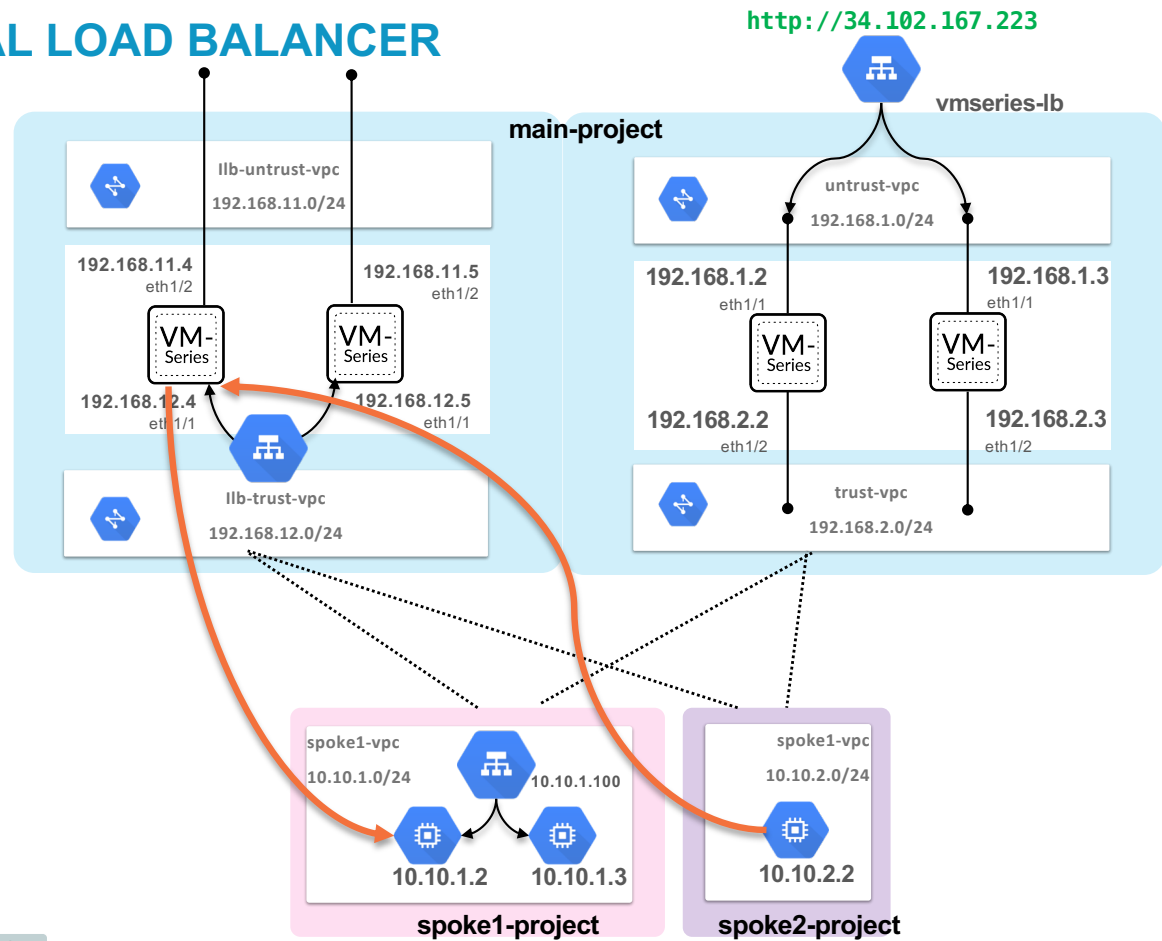
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

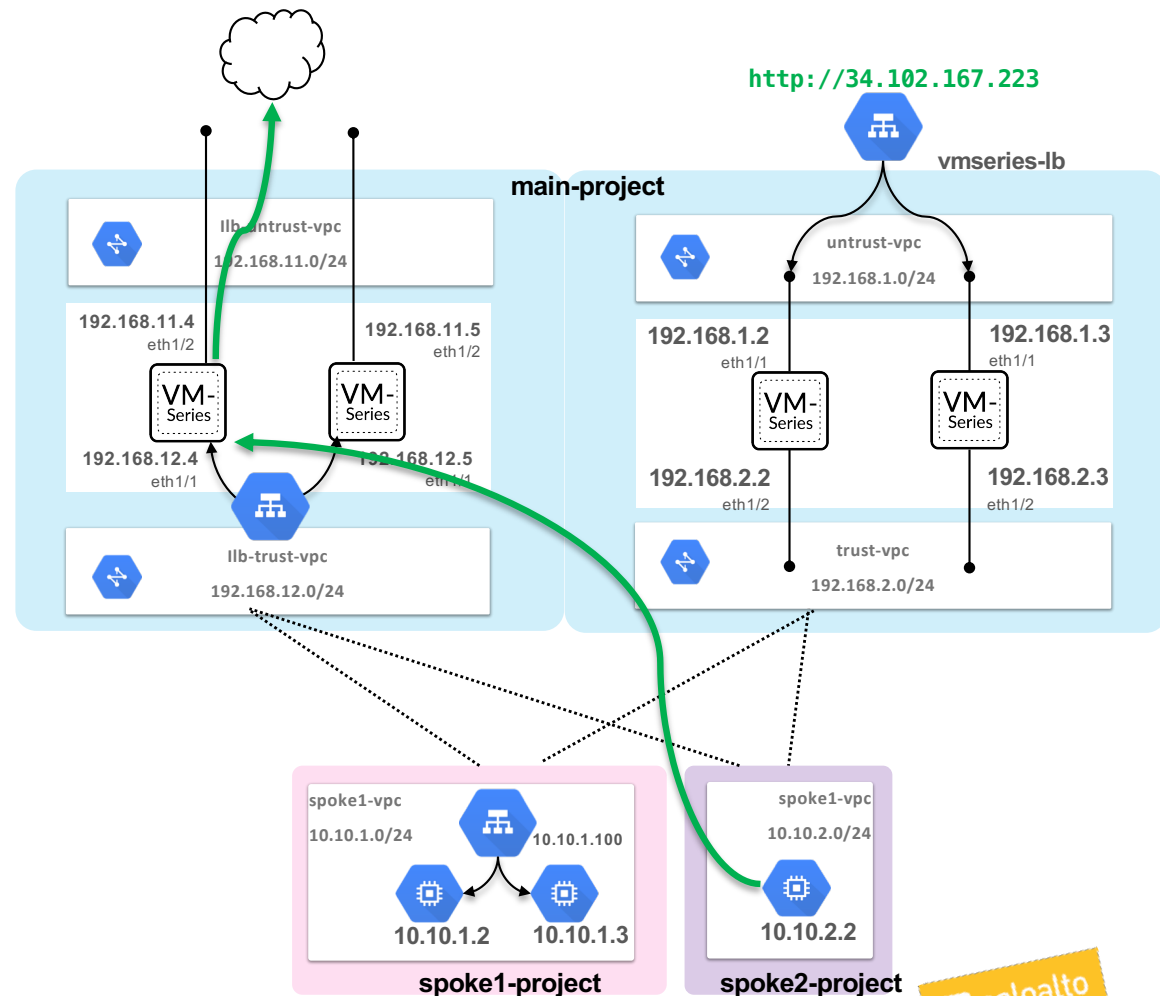
ubuntu@spoke2-vm1:~$ curl 10.10.1.2 | grep Title
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 463 100 463 0 0 69718 0 --:--:-- --:--:-- --:--:-- 77166
ubuntu@spoke2-vm1:~$
```

	Receive Time	Type	From Zone	To Zone	Source	Destination	To Port	Application	Action	Rule
	10/02 10:37:09	end	trust	trust	10.10.2.2	10.10.1.2	80	web-browsing	allow	EW



TEST OUTBOUND THROUGH INTERNAL LOAD BALANCER

1. From the Terraform output, copy `SSH-SPOKE2-FW1 = ssh ubuntu@35.224.112.224 -p 223 -i <INSERT KEY>` command prompt and insert the path to the SSH key that was used in the variables.tf file.
2. Once logged into the VM perform a curl to one of the web servers in spoke1 and check the firewall logs on the firewall VMs 3 and 4.



TEST OUTBOUND

1. Open the command prompt from the sever in spoke1 or get the ssh information from the terraform output and reconnect.
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.

```
Apply complete! Resources: 74 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.
    There is also a need to create a default route in the ilb-trust-subnet pointing to,
    the internal load balancer,
    ]
    GLB-ADDRESS = http://34.102.167.223
    MGMT-URL-FW1 = https://35.239.229.110
    MGMT-URL-FW2 = https://35.232.77.73
    MGMT-URL-ILB-FW3 = https://34.67.53.20
    MGMT-URL-ILB-FW4 = https://34.68.219.215
    SSH-SPOKE1-VMI-FW1 = ssh ubuntu@35.224.112.224 -p 221 -i <INSERT KEY>
    SSH-SPOKE1-VMI-FW2 = ssh ubuntu@34.68.252.104 -p 221 -i <INSERT KEY>
    SSH-SPOKE1-VMI-FW3 = ssh ubuntu@35.224.112.224 -p 222 -i <INSERT KEY>
    SSH-SPOKE1-VMI-FW4 = ssh ubuntu@34.68.183.184 -p 222 -i <INSERT KEY>
    SSH-SPOKE2-FW1 = ssh ubuntu@35.224.112.224 -p 223 -i <INSERT KEY>
    SSH-SPOKE2-FW2 = ssh ubuntu@35.224.112.224 -p 223 -i <INSERT KEY>
```

FW3 Egress Traffic

	Receive Time	Type	From Zone	To Zone	Source	Destination	To Port	Application	Action	Rule	Session End Reason	Bytes
	10/02 10:48:08	end	trust	untrust	10.10.2.2	35.239.123.202	80	apt-get	allow	allow-all-out-1	tcp-fin	15.7M
	10/02 10:36:56	end	trust	untrust	10.10.2.2	91.189.95.15	80	web-browsing	allow	allow-all-out-1	tcp-fin	5.7k
	10/02 10:26:02	end	trust	untrust	10.10.1.3	151.101.0.133	443	ssl	allow	allow-all-out-1	tcp-rst-from-client	8.6k
	10/02 10:25:53	end	trust	untrust	10.10.1.3	35.184.34.241	80	apt-get	allow	allow-all-out-1	tcp-fin	1.3M
	10/02 10:25:53	end	trust	untrust	10.10.1.2	35.239.123.202	80	apt-get	allow	allow-all-out-1	tcp-fin	1.3M
	10/02 10:25:45	end	trust	untrust	10.10.1.3	35.184.34.241	80	apt-get	allow	allow-all-out-1	tcp-fin	1.6M

FW4 Egress Traffic

	Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action	Rule	Session End Reason	Bytes
	10/02 10:48:08	end	trust	untrust	10.10.2.2		91.189.91.24	80	apt-get	allow	allow-all-out-1	tcp-fin	4.1k
	10/02 10:48:08	end	trust	untrust	10.10.2.2		91.189.92.191	80	apt-get	allow	allow-all-out-1	tcp-fin	19.3k
	10/02 10:48:06	end	trust	untrust	10.10.2.2		91.189.91.24	80	apt-get	allow	allow-all-out-1	tcp-fin	808.6k
	10/02 10:48:06	end	trust	untrust	10.10.2.2		91.189.91.24	80	apt-get	allow	allow-all-out-1	tcp-fin	204.5k
	10/02 10:48:06	end	trust	untrust	10.10.2.2		91.189.91.24	80	apt-get	allow	allow-all-out-1	tcp-fin	7.2k
	10/02 10:48:06	end	trust	untrust	10.10.2.2		91.189.91.24	80	apt-get	allow	allow-all-out-1	tcp-fin	494.2k

