GCP Transit VPC

VPC Peering & Internal Load Balancer as Next Hop

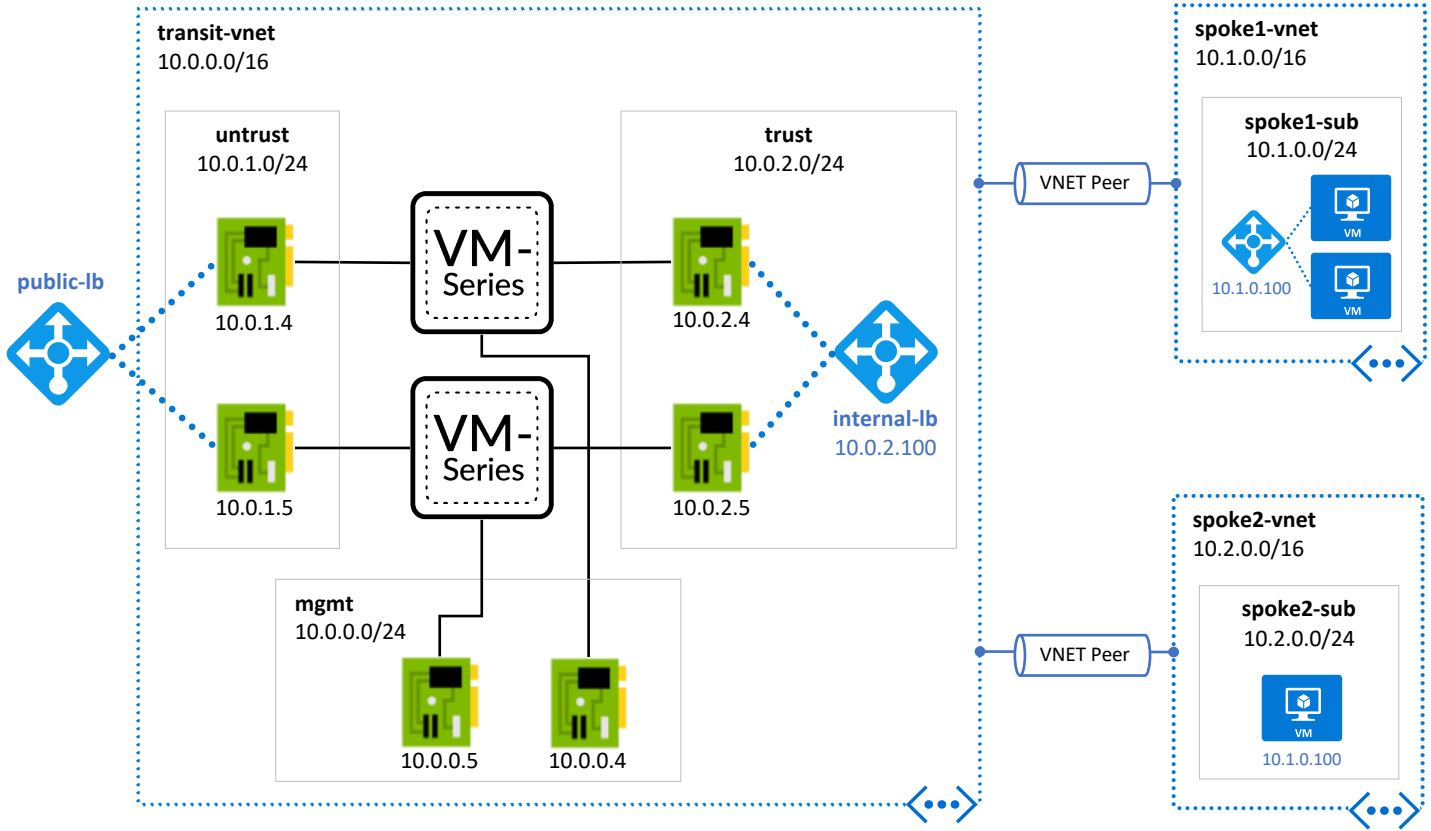
Matt McLimans
Public Cloud Consultant Engineer

paloalto NETWORKS®

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.

- VM-Series deployed between public & internal load balancers to secure spoke VNET traffic:
  - Inbound
  - Outbound
  - East-west
- The spoke VNETs connect to the transit VNET via peering.
- Spoke1 contains Web VMs frontended by an internal load balancer.
- Spoke2 contains a Linux VM.



Requirements

1. A valid Azure subscription
2. A valid VM-Series authcode (if using BYOL)

PART 1
BUILD ENVIRONMENT

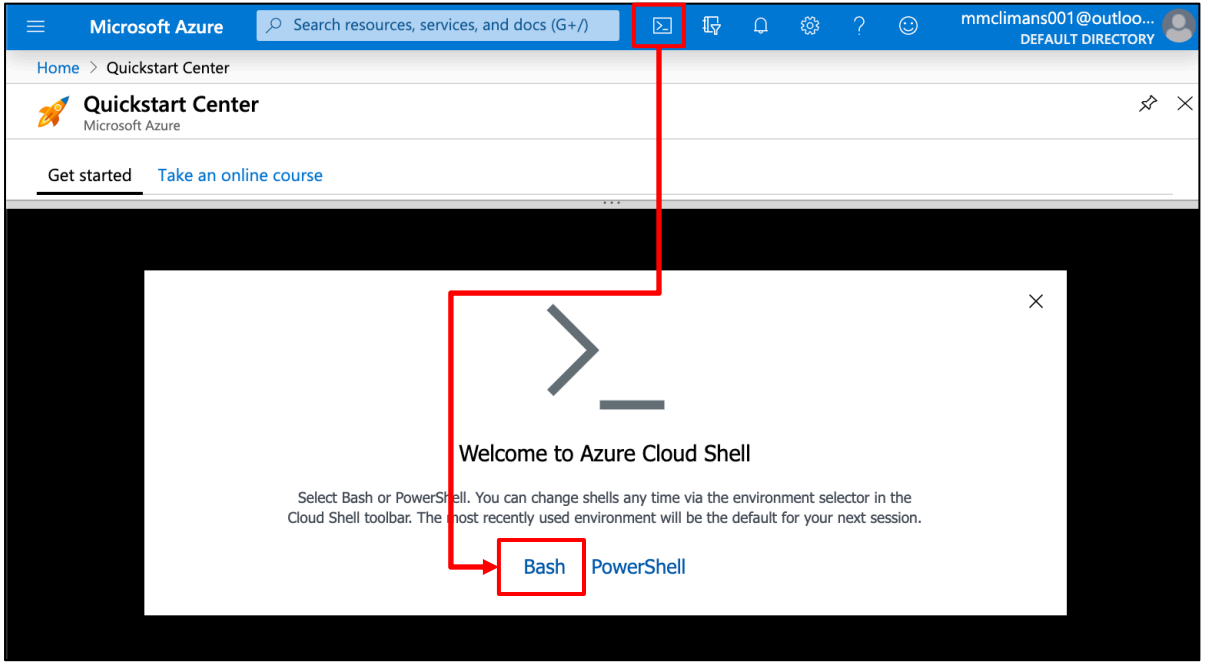
Step 1. Open Azure Cloud Shell

1. Log into Azure Portal
   https://portal.azure.com

2. Click  to open Cloud Shell.

3. Select **Bash**

Note, you will be prompted to create a storage account if you do not have one.



Step 2. Accept VM-Series EULA

You must accept Marketplace EULAs for the desired VM-Series license SKU:

1. BYOL

```
az vm image terms accept --urn paloaltonetworks:vmseries1:byol:9.0.1
```

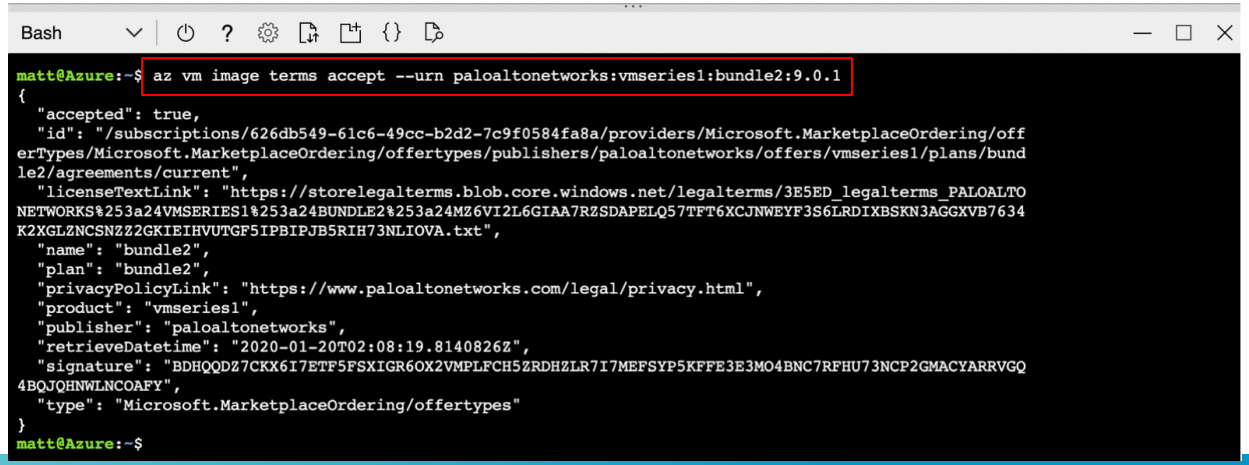
2. Bundle1

```
az vm image terms accept --urn paloaltonetworks:vmseries1:bundle1:9.0.1
```

3. Bundle2

```
az vm image terms accept --urn paloaltonetworks:vmseries1:bundle2:9.0.1
```

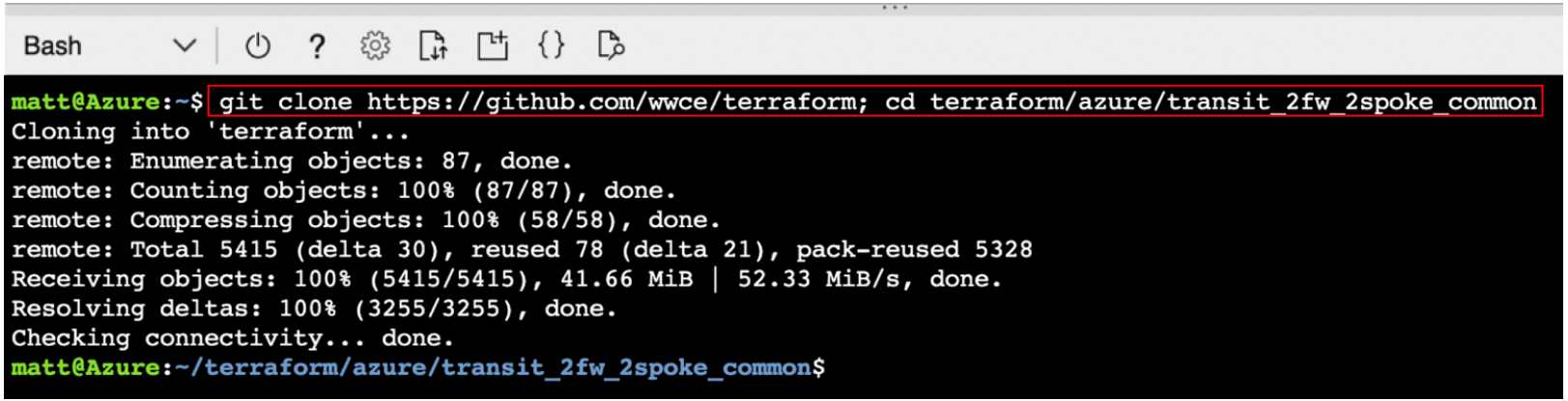
For the desired offering/license, copy & paste the command it into your cloud shell.



Step 3. Download Github Repo

1. Clone the Github repo & change directories to `transit_2fw_2spoke_common`

```
$ git clone https://github.com/wwce/terraform; cd terraform/azure/transit_2fw_2spoke_common
```

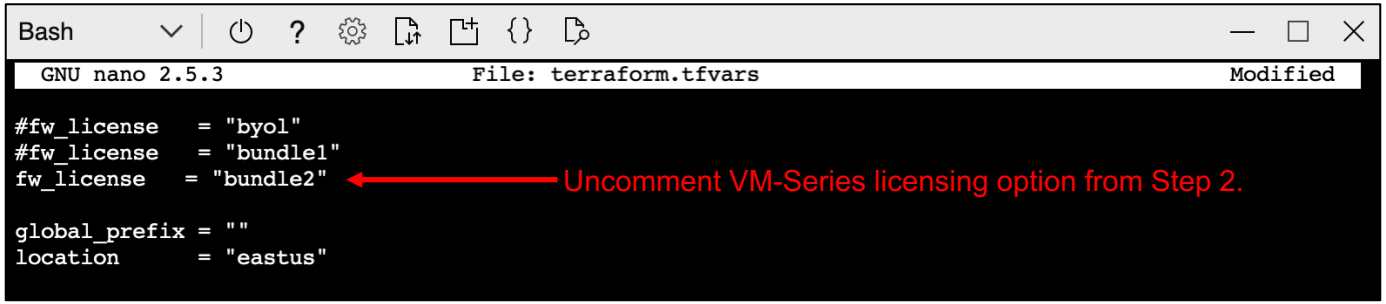


Step 4. Update terraform.tfvars

1. Open the terraform.tfvars file

```
$ nano terraform.tfvars
```

2. Uncomment **only one** value for **fw_license** (delete # to uncomment)

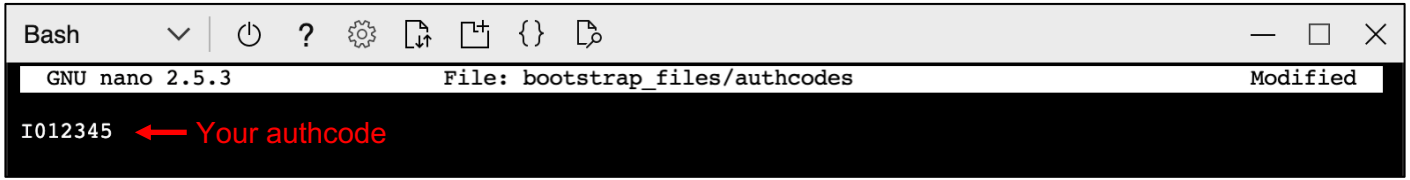


3. Save file and exit: `ctrl-x` → `y` → `enter`

BYOL ONLY

- You can manually license the firewalls post deployment.
- If you want to license the firewalls during deployment (via bootstrapping):
  1. Ensure authcode is registered with the Palo Alto Networks support site.
  2. Open bootstrap_files/authcodes & enter your authcode.

```
$ nano bootstrap_files/authcodes
```



  3. Save file and exit: `ctrl-x` → `y` → `enter`

STEP 5. DEPLOY BUILD

Initialize and apply the Terraform build

```
$ terraform init
```

```
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$ 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.
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$
```

```
$ terraform apply
```

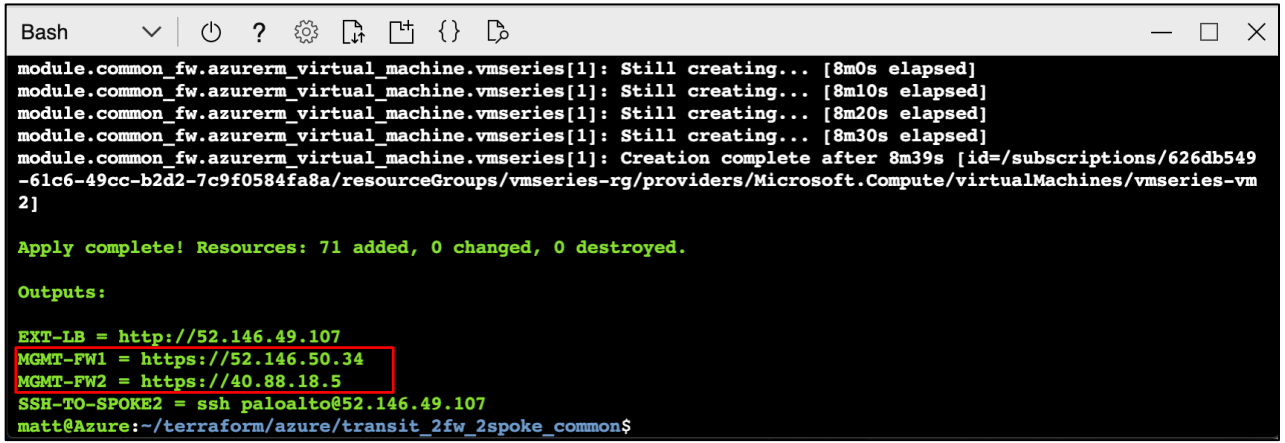
```
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$ terraform apply

Plan: 71 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
```

Enter **yes** to start build

Step 6. Build Completion & Firewall Login

- Once the build completes, the following output will be displayed

It may take up to 15 minutes for the firewalls to fully boot.

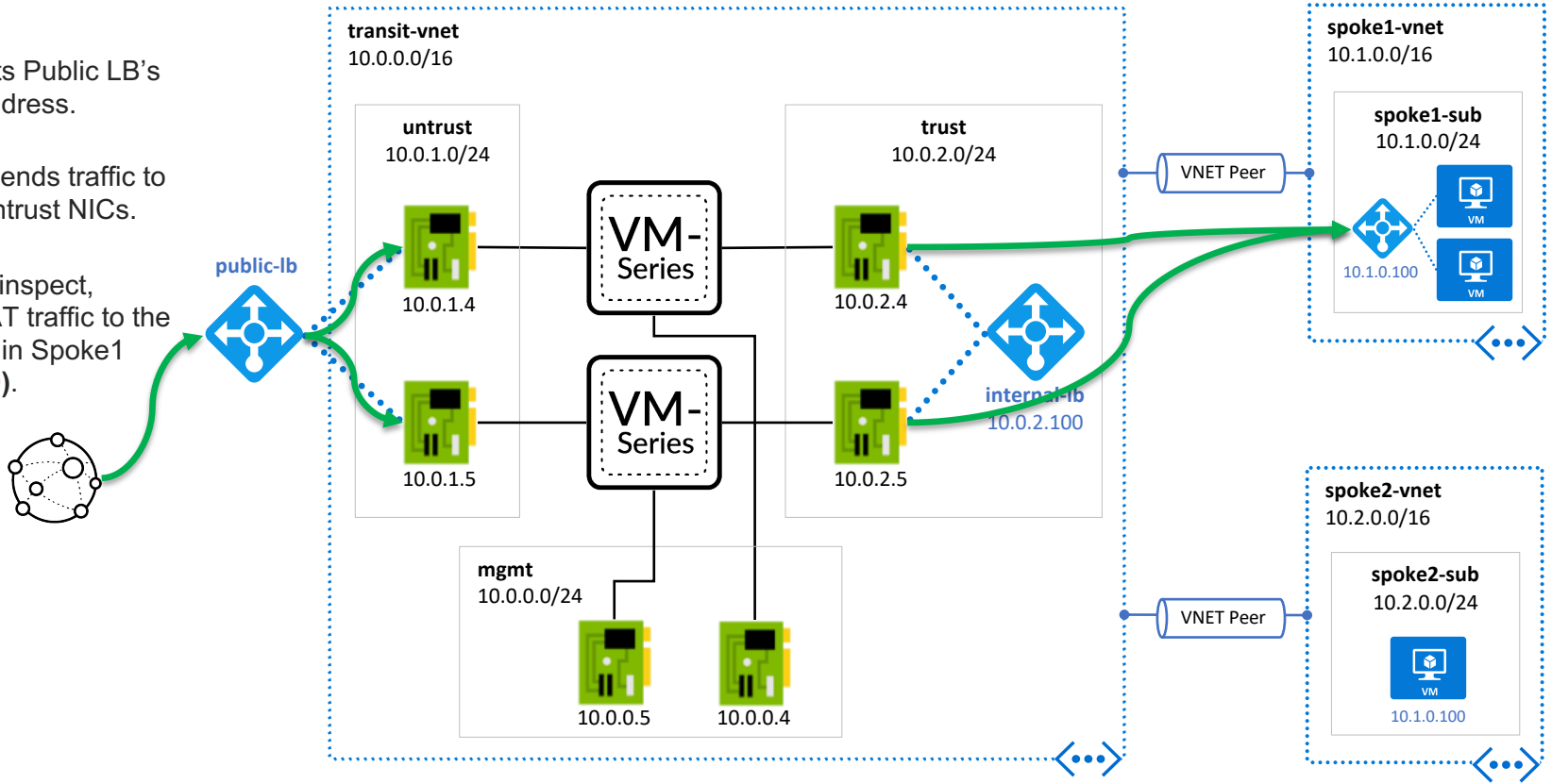


- Copy the **MGMT-FW1** & **MGMT-FW2** outputs into a web-browser.
  - **UN:** `paloalto`
  - **PW:** `Pal0Alt0@123`

PART 2
TEST ENVIRONMENT

Inbound Flow

1. Request hits Public LB's frontend address.
2. Public LB sends traffic to firewall's untrust NICs.
3. VM-Series inspect, SNAT/DNAT traffic to the internal LB in Spoke1 (**10.1.0.100**).



Inbound Flow Test

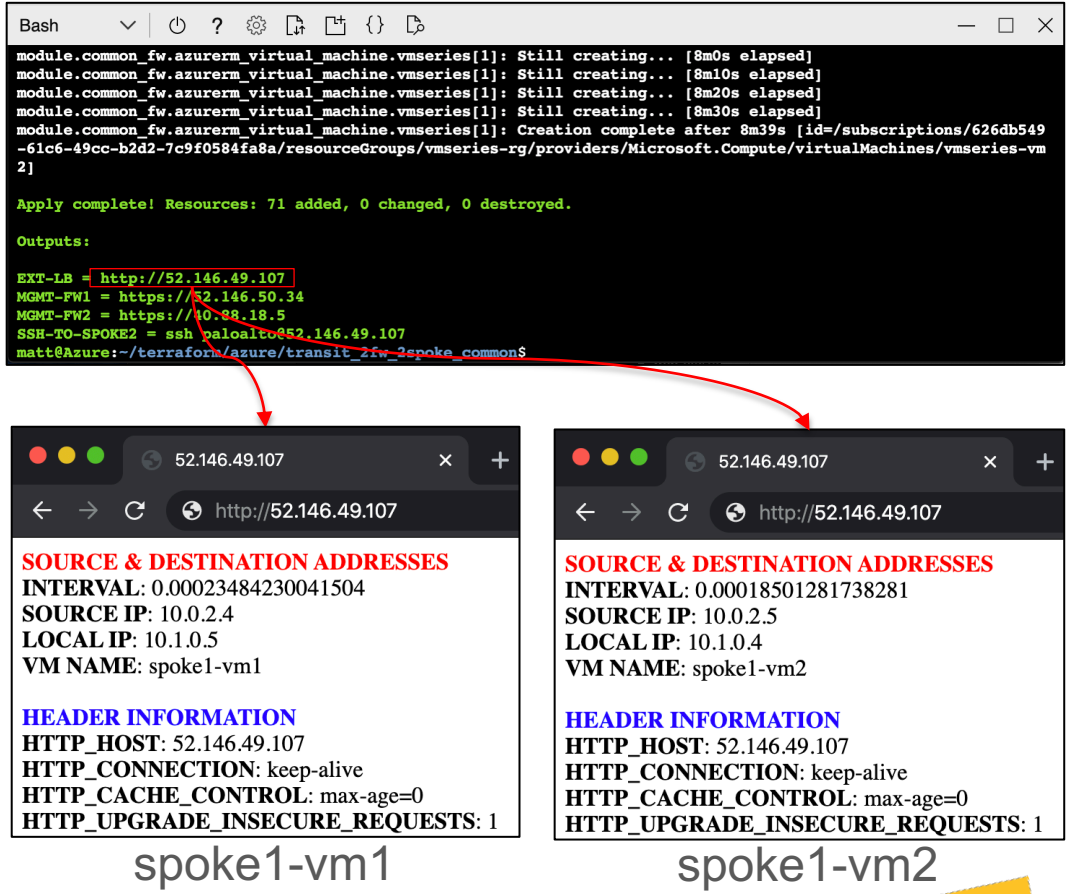
1. Copy the **EXT-LB** output into a web browser.
2. Observe inbound load balancing by refreshing the page. The following values will change:

SOURCE IP

- FW trust IP that is handling request.

LOCAL IP

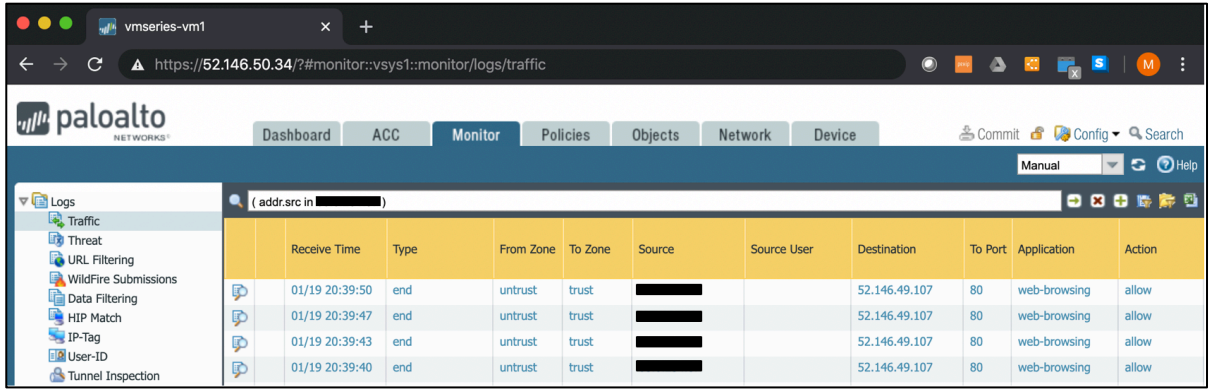
- IP Address of the VM in Spoke1 that received the request.



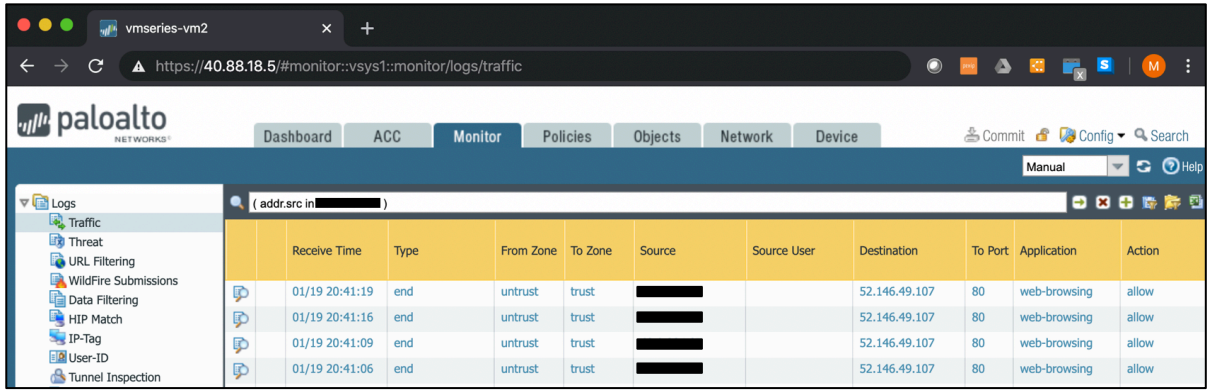
Inbound Flow Test

- View the inbound traffic on both firewalls (Monitor → Traffic)

vmseries01

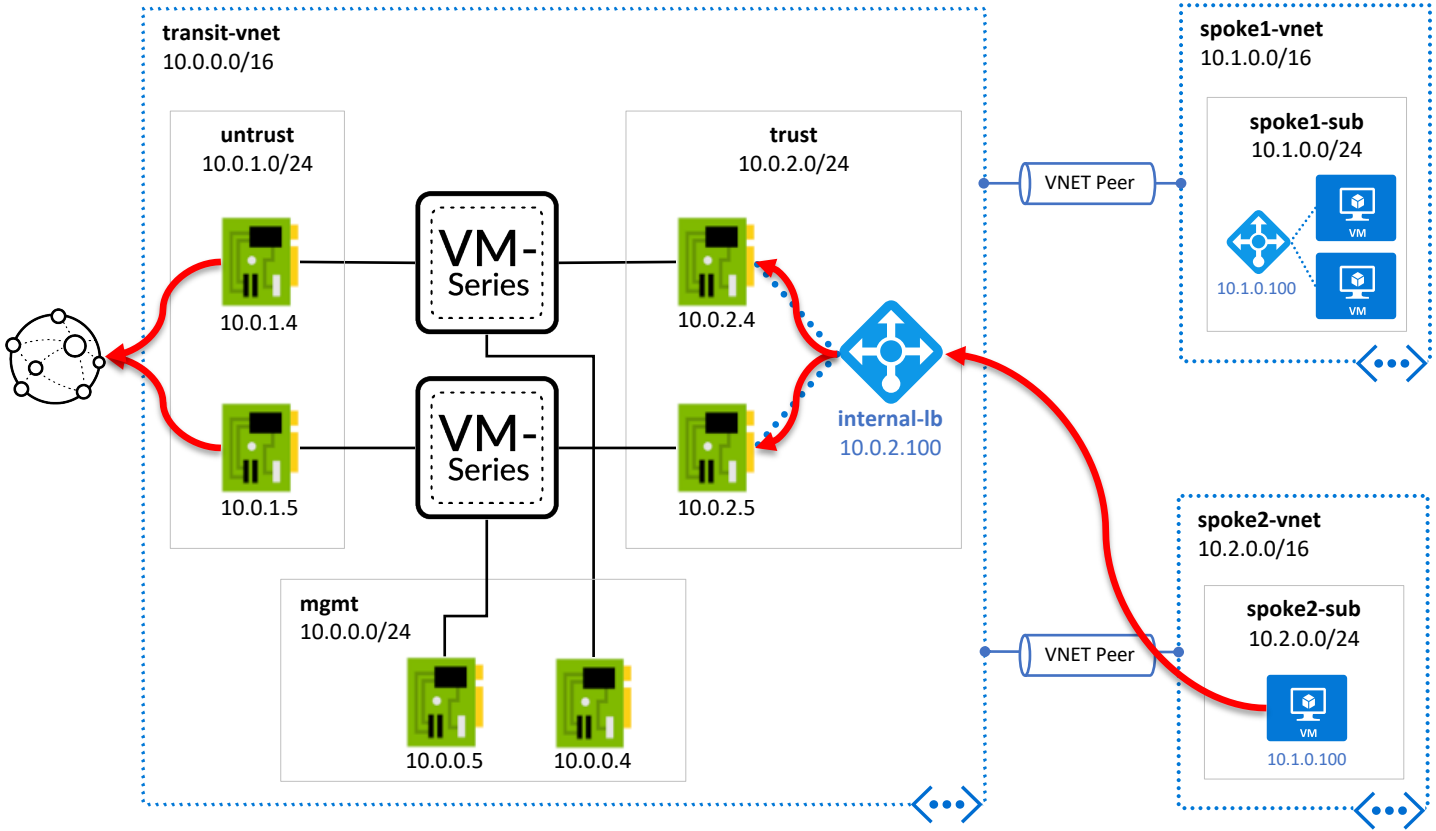


vmseries02



Outbound Flow

1. VM in spoke2-subnet sends a request to internet.
2. UDR within spoke2's route table sends traffic to internal LB 10.0.2.100.
3. LB forwards traffic to firewall's trust interfaces.
4. VM-Series inspects, SNATs to untrust interface.
5. Untrust NIC sends traffic out attached public IP to internet.



Outbound Flow Test

1. Log into spoke2-vm. Copy the **SSH-TO-SPOKE2** output and paste it into your terminal

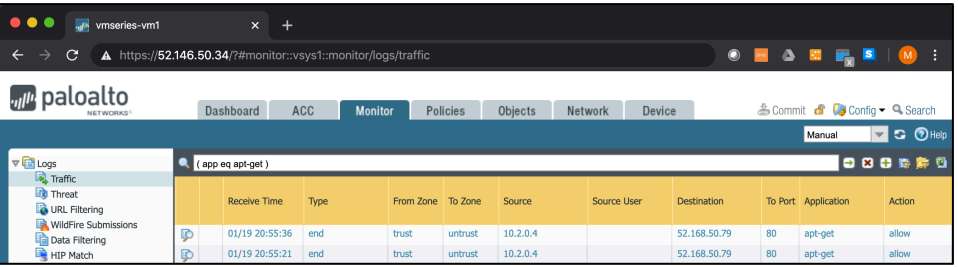
2. From the spoke2-vm, test outbound connectivity:

   ```
   $ sudo apt-get update
   ```

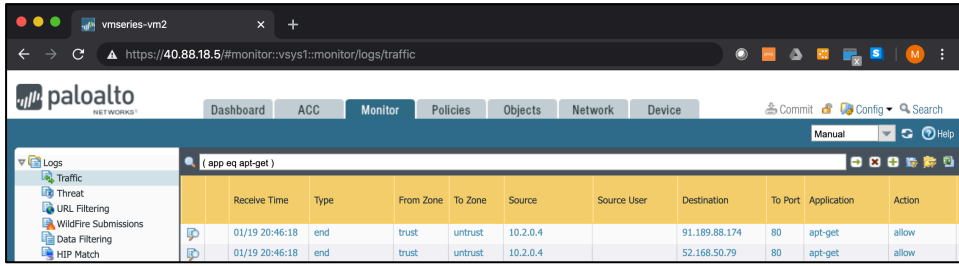
3. View the outbound traffic on both firewalls (Monitor → Traffic)

```
Bash
Outputs:

EXT-LB = http://52.146.49.107
MGMT-FW1 = https://52.146.50.34
MGMT-FW2 = https://40.88.18.5
SSH-TO-SPOKE2 = ssh paloalto@52.146.49.107
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$ ssh paloalto@52.146.49.107
paloalto@52.146.49.107's password:
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.15.0-1066-azure x86_64)

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


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

New release '18.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Jan 20 04:45:50 2020 from 10.0.2.4
paloalto@spoke2-vm1:~$ sudo apt-get update
```

vmseries01

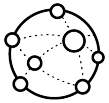


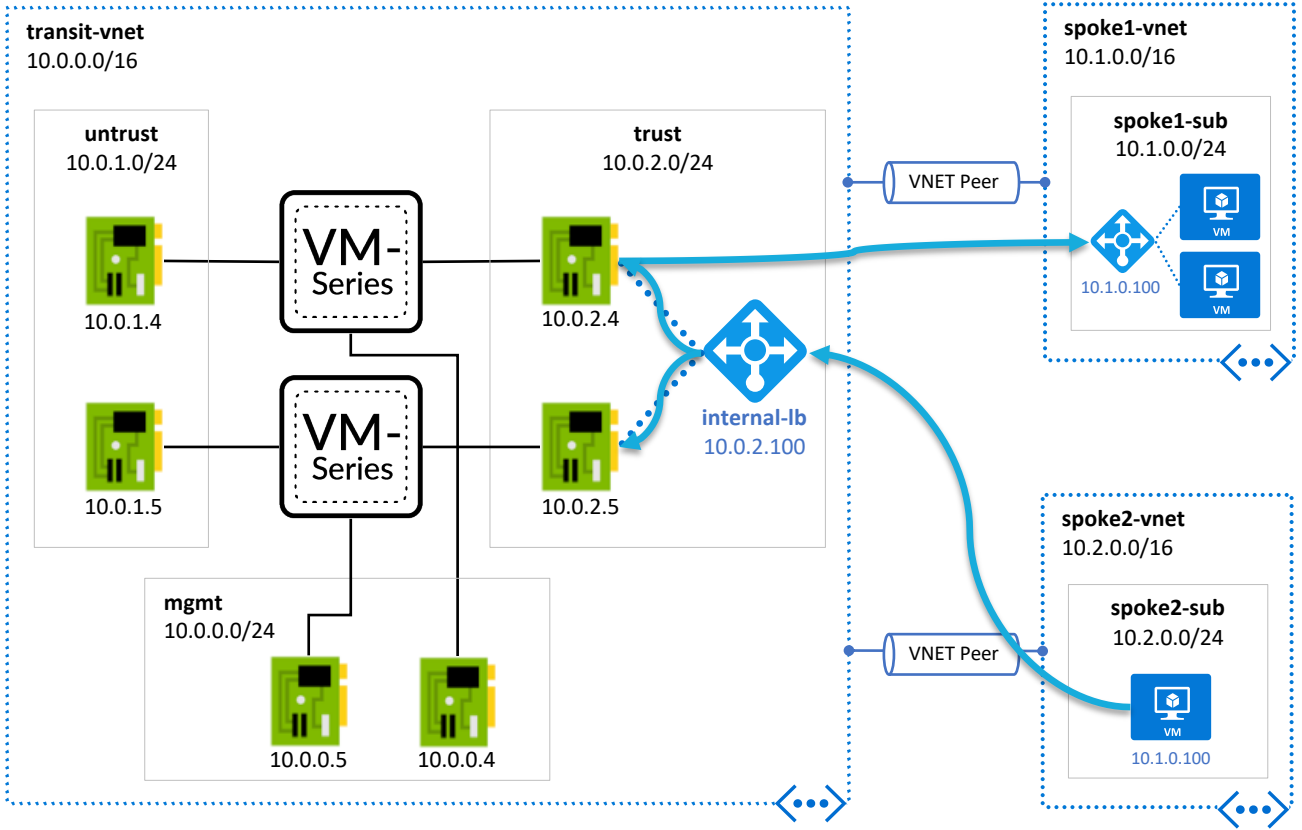
vmseries02



East-West Flow

1. VM in spoke2-subnet sends a request to VM in spoke1-subnet.
2. UDR within spoke2's route table sends traffic to internal LB 10.0.2.100.
3. LB forwards traffic to firewall's trust interfaces.
4. VM-Series inspects and routes traffic to spoke1-subnet.
5. Response back from spoke1-subnet takes same path back.





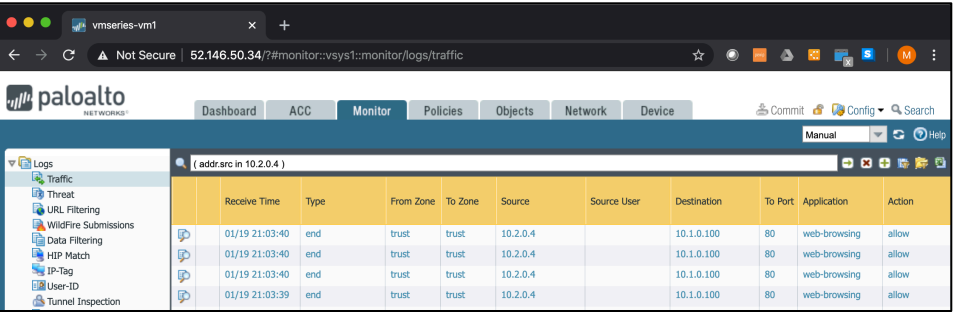
East-West Flow Test

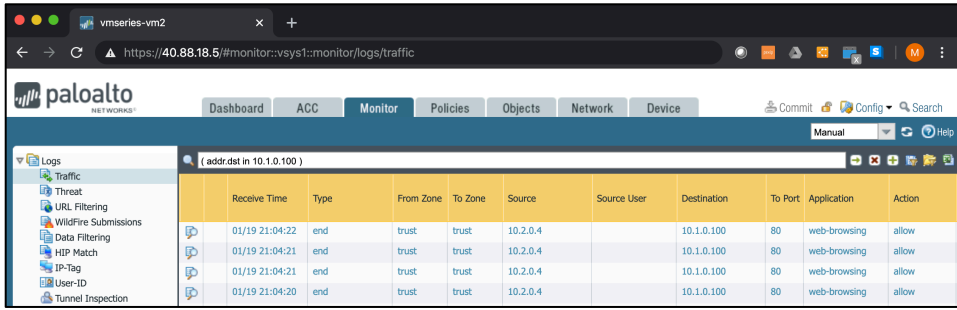
1. Stay inside spoke2-vm1 (**10.2.0.4**)
2. Run a curl command to the web server's internal load balancer in spoke1 (**10.1.0.100**)

```
$ curl http://10.1.0.100/?[1-1000]
```

```
paloalto@spoke2-vm1:~$ curl http://10.1.0.100/?[1-1000]
```

3. View the east-west traffic on the firewalls (Monitor → Traffic)





DESTROY ENVIRONMENT

DESTROY BUILD

Perform the following to destroy the deployment.

1. Verify you are in the `adv_peering_4fw_2spoke` directory and run the following. Enter `yes` to confirm.

```
$ terraform destroy
```

```
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$ terraform destroy

Plan: 0 to add, 0 to change, 71 to destroy.

Do you really want to destroy all resources?
  Terraform will destroy all your managed infrastructure, as shown above.
  There is no undo. Only 'yes' will be accepted to confirm.

  Enter a value: yes
```

2. Once the destroy is complete, you will receive the following output:

```
azurerm_resource_group.common_fw: Destruction complete after 3m16s

Destroy complete! Resources: 71 destroyed.
matt@Azure:~/terraform/azure/transit_2fw_2spoke_common$
```

DEPLOYMENT CHEAT SHEET

Part 1: Setup Build

```
$ az vm image terms accept --urn paloaltonetworks:vmseries1:<byol/bundle1/bundle2>:9.0.1
$ git clone https://github.com/wwce/terraform; cd terraform/azure/transit_vnet_2fw_2spoke_common
```

Part 2: Run Build

Edit terraform.tfvars to match PAN-OS license.

```
$ terraform init
$ terraform apply
```

Part 3: Destroy Build

```
$ terraform destroy
```

