paloalto NETWORKS

VM-Series & Azure Virtual WAN

Template Build Guide

Matt McLimans, Public Cloud CE

January 2021

Introduction

- This guide walks through several VM-Series deployment architectures for Azure Virtual WAN
- The build uses VM-Series scale sets deployed through Panorama Orchestration to secure:
  - Internet inbound traffic
  - Lateral traffic traversing through a virtual hub

- The build is broken down into 5 steps
  1. Create Virtual WAN & Virtual Hub
  2. Connect Security Inbound VNET to the Virtual Hub
  3. Connect Security Outbound VNET to the Virtual Hub
  4. Peer Local VNET to Security Outbound VNET
  5. Connect Spoke VNET to Virtual Hub

Prerequisites

1. A active Azure subscription with appropriate permissions and resource allocation quota

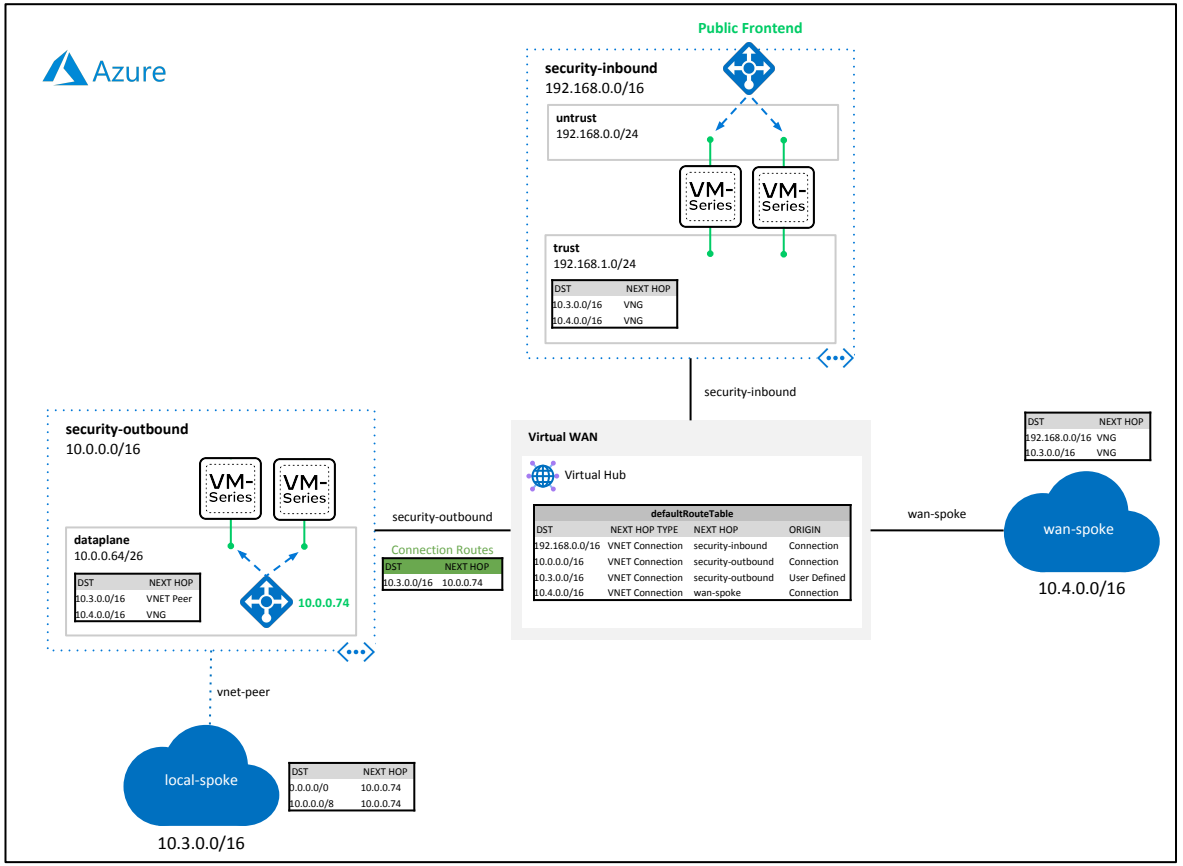
2. Panorama
   a. PAN-OS 10.0.0 or greater
   b. Panorama Azure Plugin 3.0 or greater

3. Access to this Github repo to automate the deployment of required Azure resources

   https://github.com/wwce/azure-arm-virtual-wan

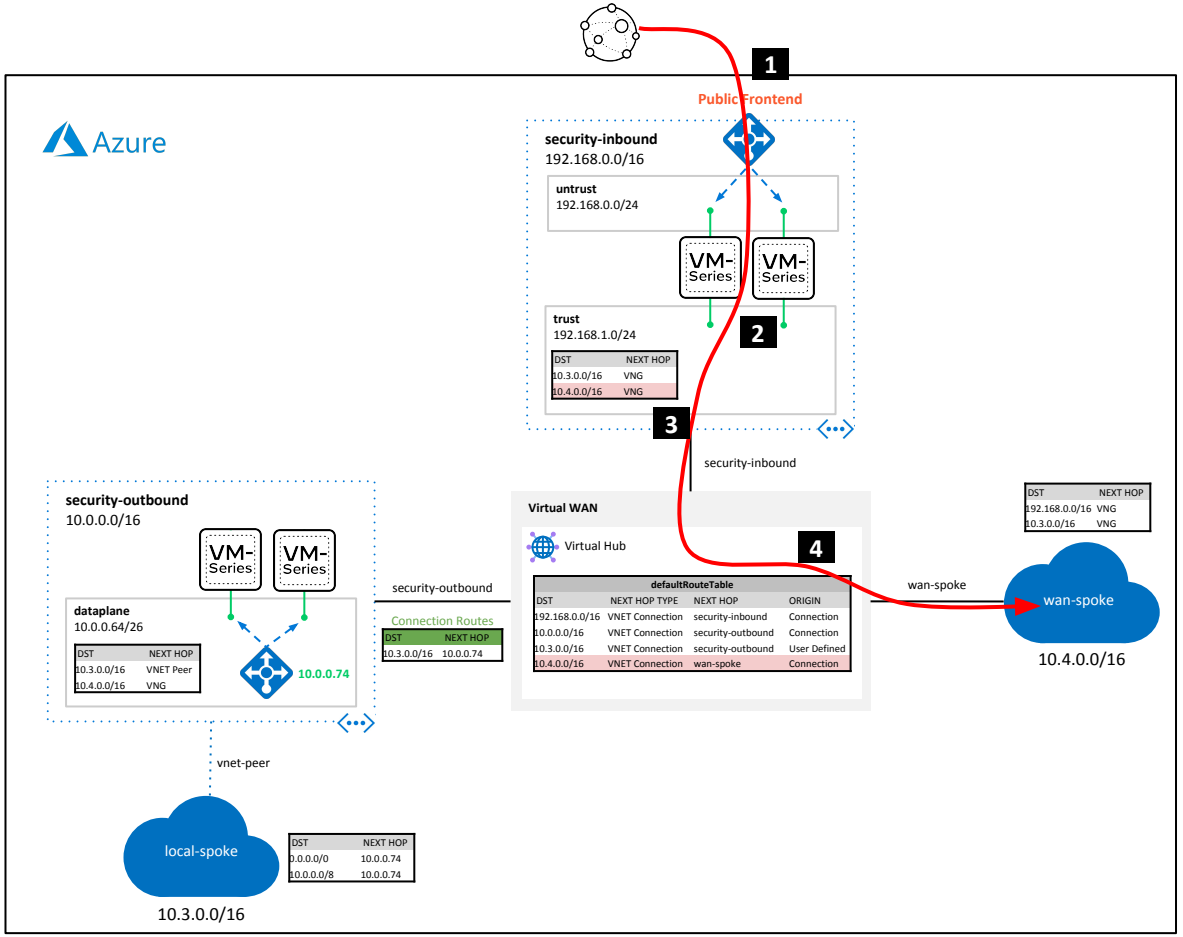
Architecture Diagram

- Security Inbound
  - VM-Series scale set secures inbound traffic to wan-spoke
- Security Outbound
  - VM-Series scale set secures vWAN traffic to local-spoke



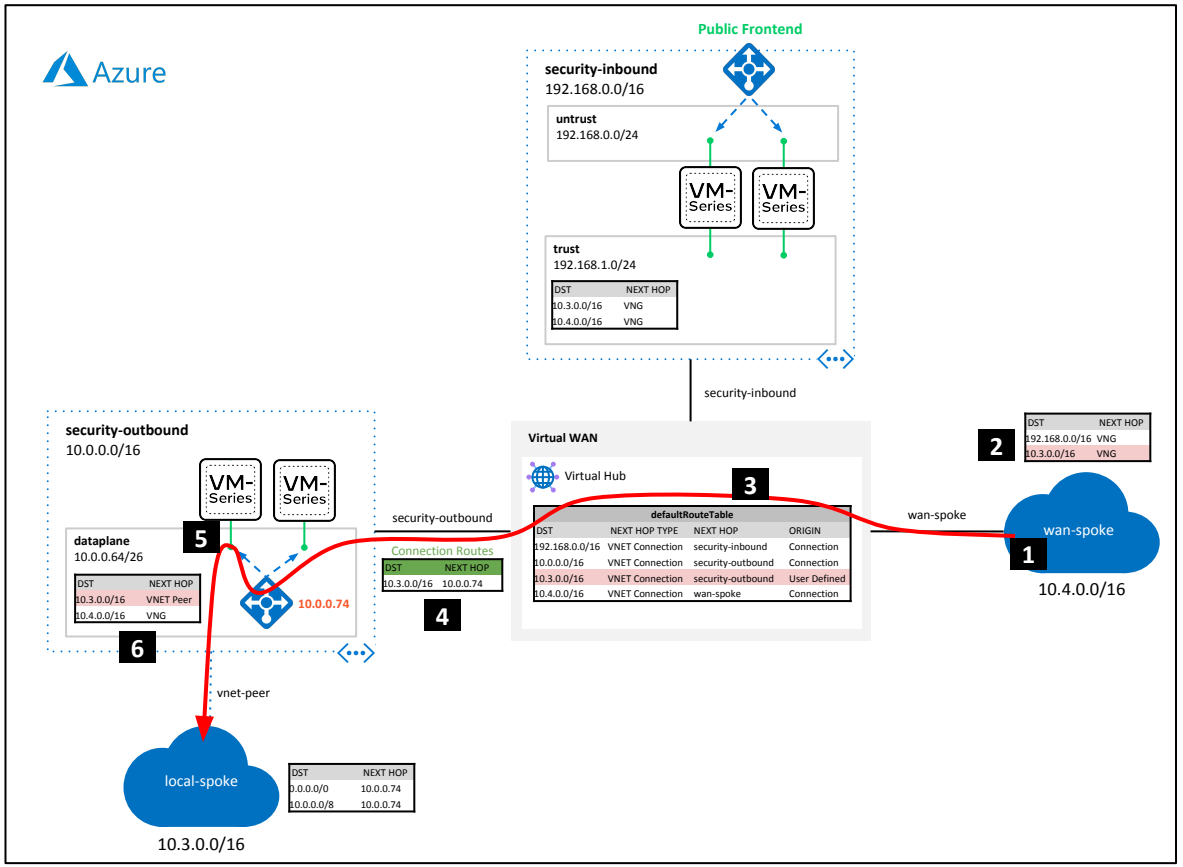
Inbound Flow

1. Inbound request hits public load balancer frontend IP.
2. VM-Series SNAT to trust interface and DNAT to destination (**10.4.0.x**).
3. Trust subnet sends traffic to vWAN hub via propagated route.
4. vWAN hub route table sends request to wan-spoke VNET connection.



Lateral Flow

1. wan-spoke VNET makes request to local-spoke (**10.3.0.x)**
2. Spoke subnet sends request to vWAN hub via propagated route.
3. vWAN hub route table has user defined route to direct request to the security-outbound VNET connection.
4. The security-outbound VNET connection has a “connection-route” to direct request to the internal load balancer of the VM-Series (**10.0.0.74**).
5. Internal load balancer forwards traffic to VM-Series
6. VM-Series dataplane subnet sends request to its local peer (**10.3.0.x**)

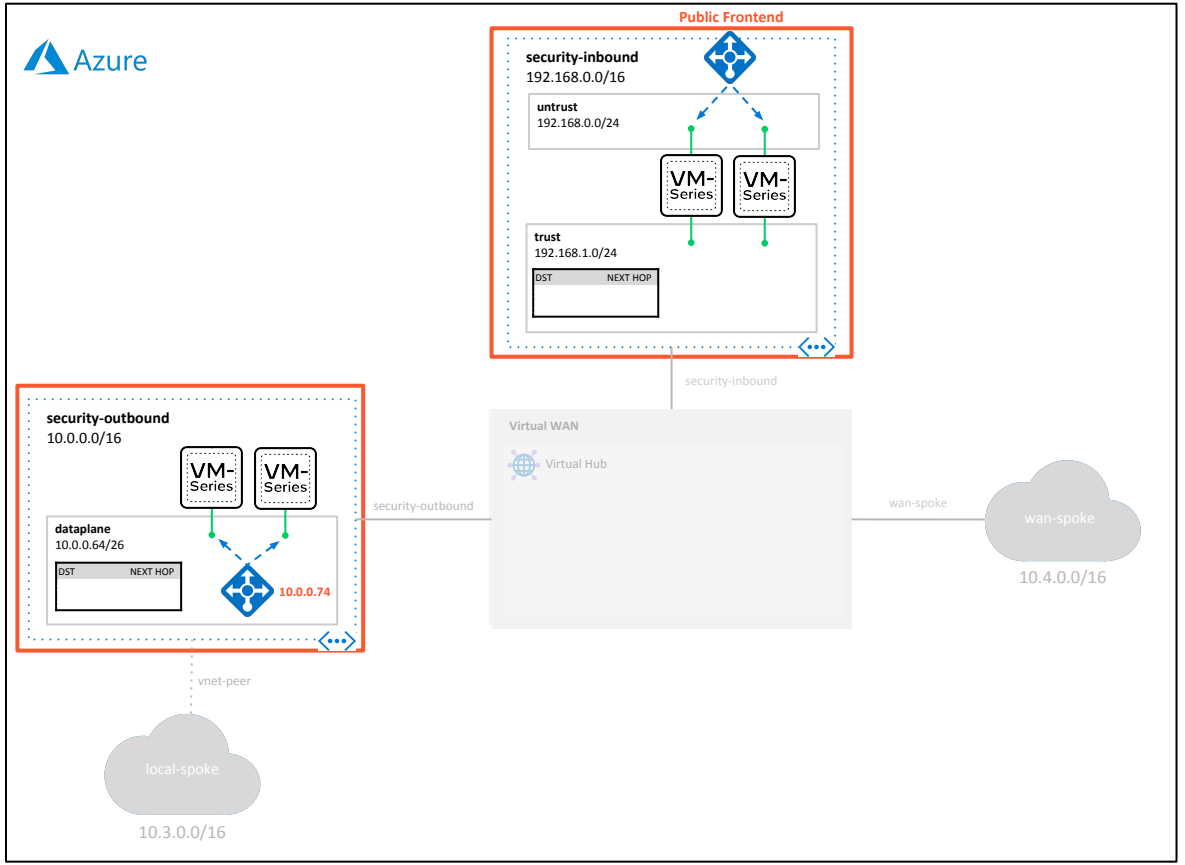


Prerequisite Setup

Create Inbound & Outbound Security VNETs

In this step, we will use Panorama Orchestration to deploy an inbound and outbound VM-Series scale set.

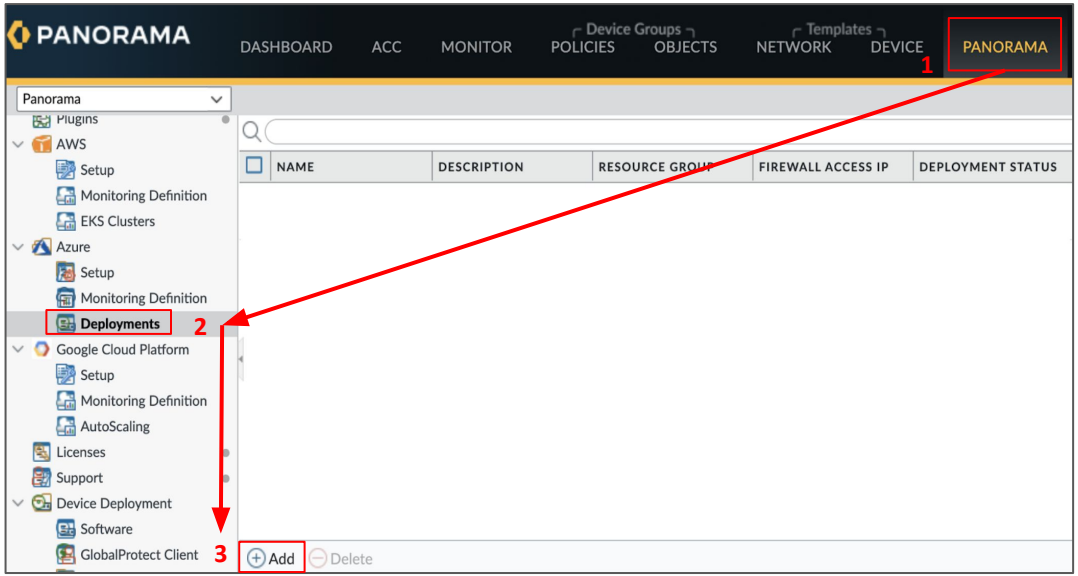
Each scale set will belong to its own VNET.



Deploy VM-Series Scale Sets

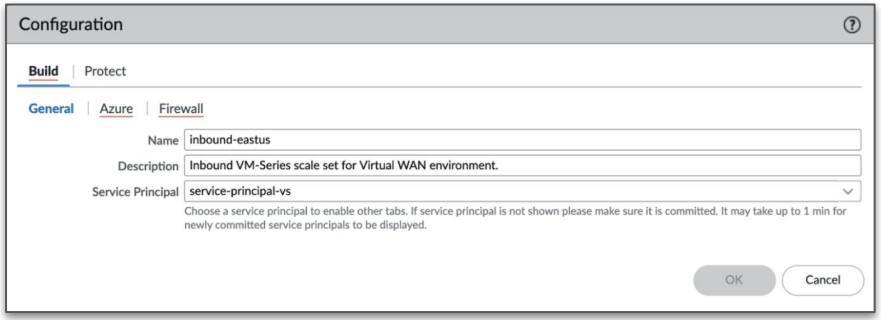
1. Login to Panorama

Open *Panorama* tab
→ Azure Plugin
→ Deployments
→ Add

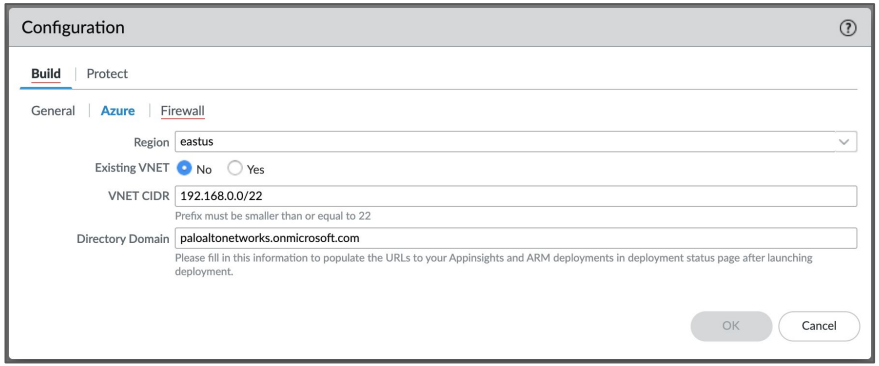


Create Inbound VM-Series Scale Set (1)

1. Enter a Name and select your Service Principal.

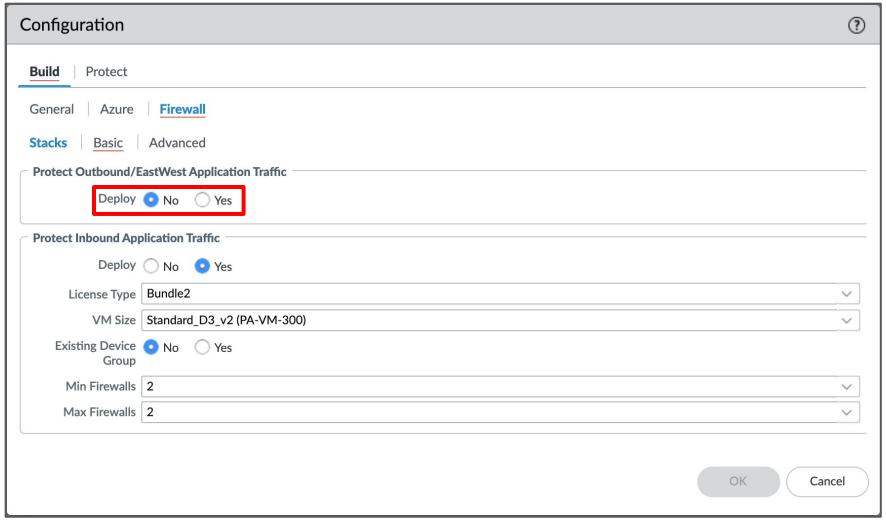


2. Select a region (must be same region as your Virtual Hub)
3. Enter a unique CIDR for the inbound VNET.

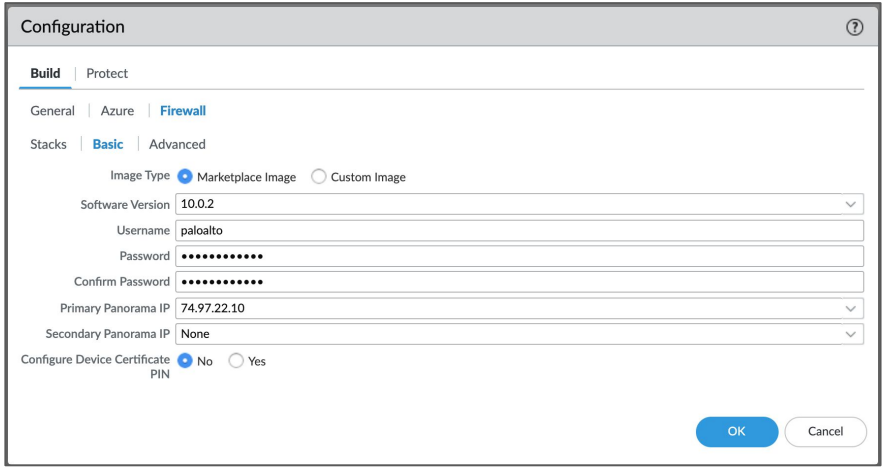


Create Inbound VM-Series Scale Set (2)

5. Virtual WAN best practices recommend separate VNETs for inbound and outbound security VNETs. Therefore, only select ***Yes*** to deploy the inbound VNET. *Select* ***No*** on the outbound/eastwest VNET.



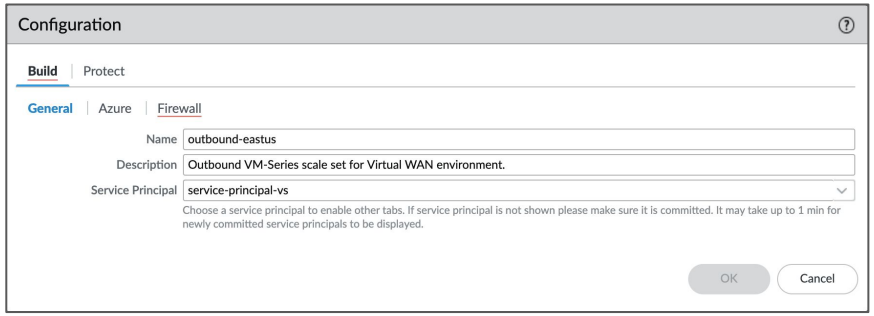
6. Select a Marketplace Image
7. Select Software Version (must <= Panorama version).
8. Enter a username and password.
9. Enter your Panorama IP



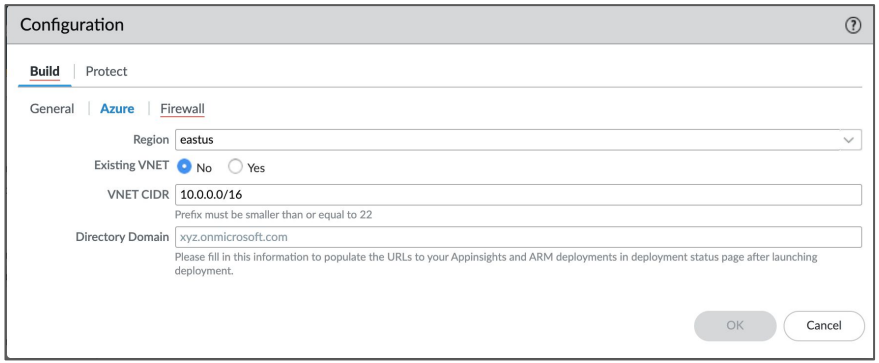
Create Outbound VM-Series Scale Set

Create Outbound VM-Series Scale Set (1)

1. Enter a Name and select your Service Principal.

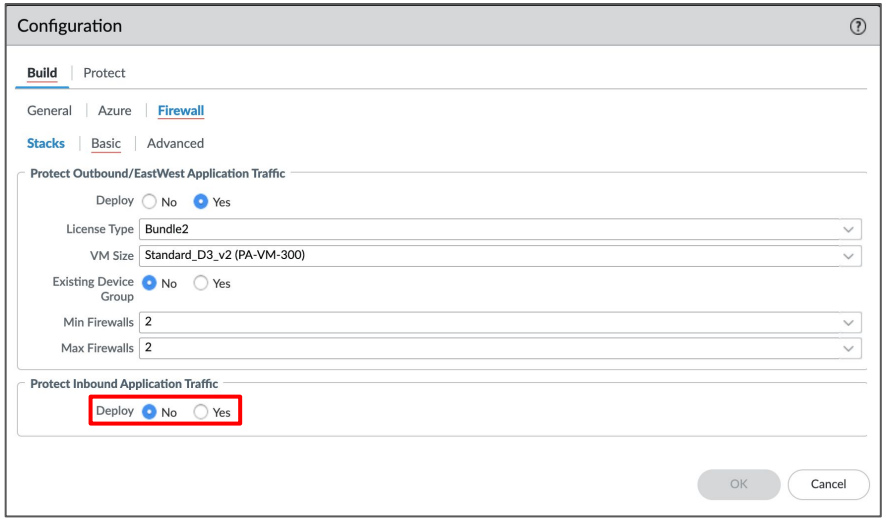


2. Select a region (must be same region as your Virtual Hub)
3. Enter a unique CIDR for the outbound VNET.

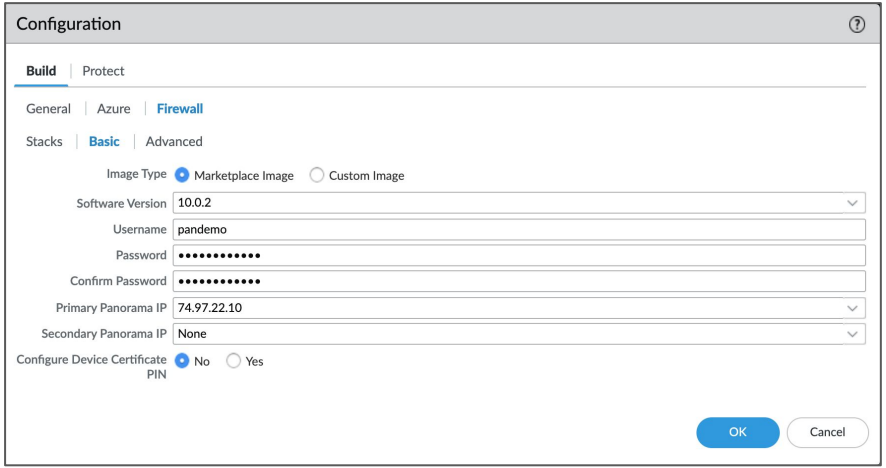


Create Outbound VM-Series Scale Set (2)

5. Virtual WAN best practices recommend separate VNETs for inbound and outbound security VNETs. Therefore, only select ***No*** to deploy the inbound VNET. *Select* ***Yes*** on the outbound/eastwest VNET.



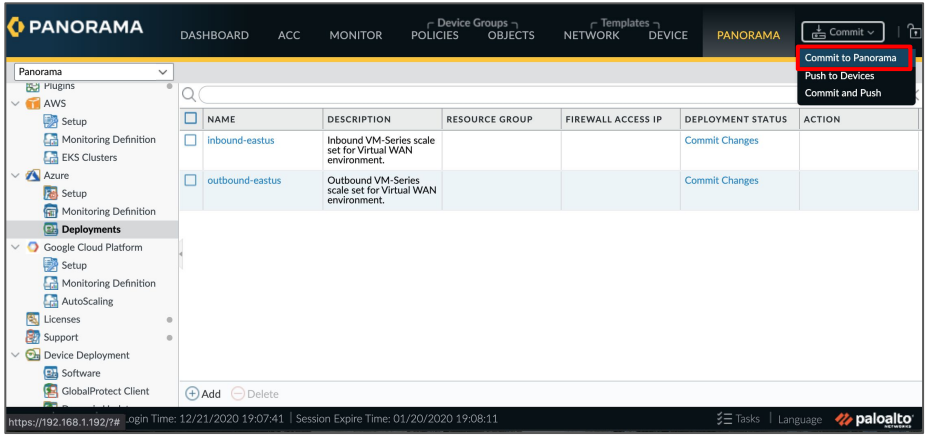
6. Select a Marketplace Image
7. Select Software Version (must <= Panorama version).
8. Enter a username and password.
9. Enter your Panorama IP

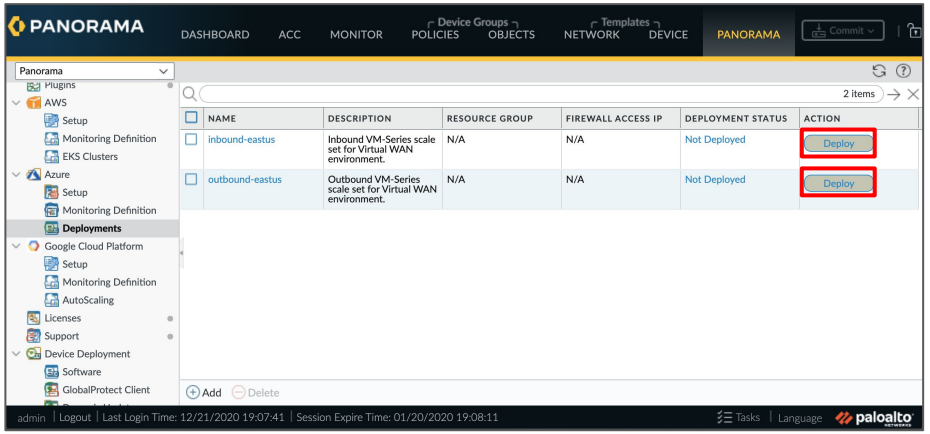


Commit to Panorama & Deploy

1. Commit the changes to Panorama.

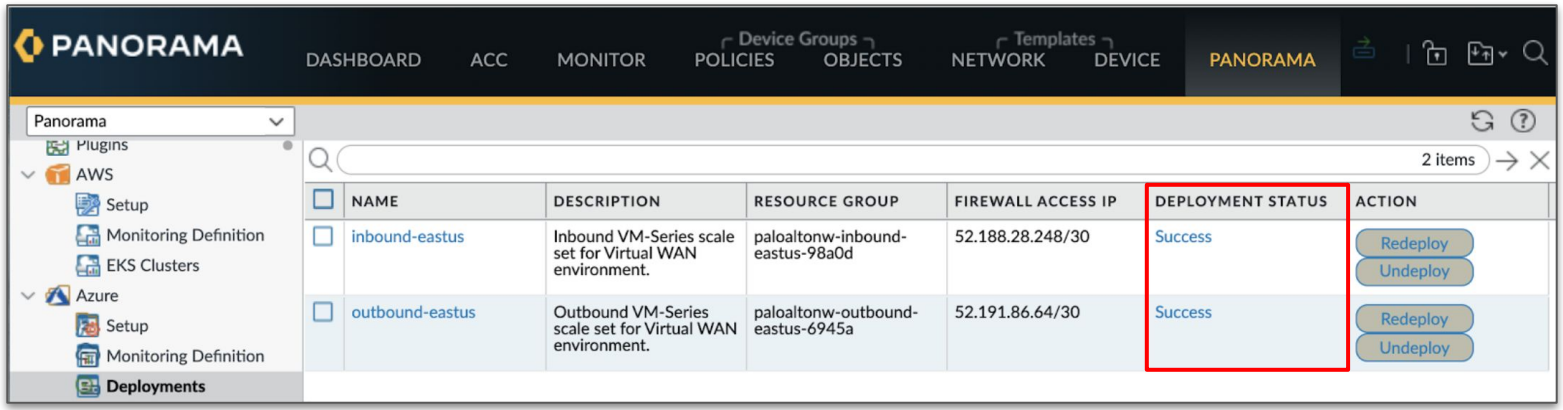
2. Click ***Deploy*** button for the inbound deployment
3. Click ***Deploy*** button for the outbound deployment





Completion Verification

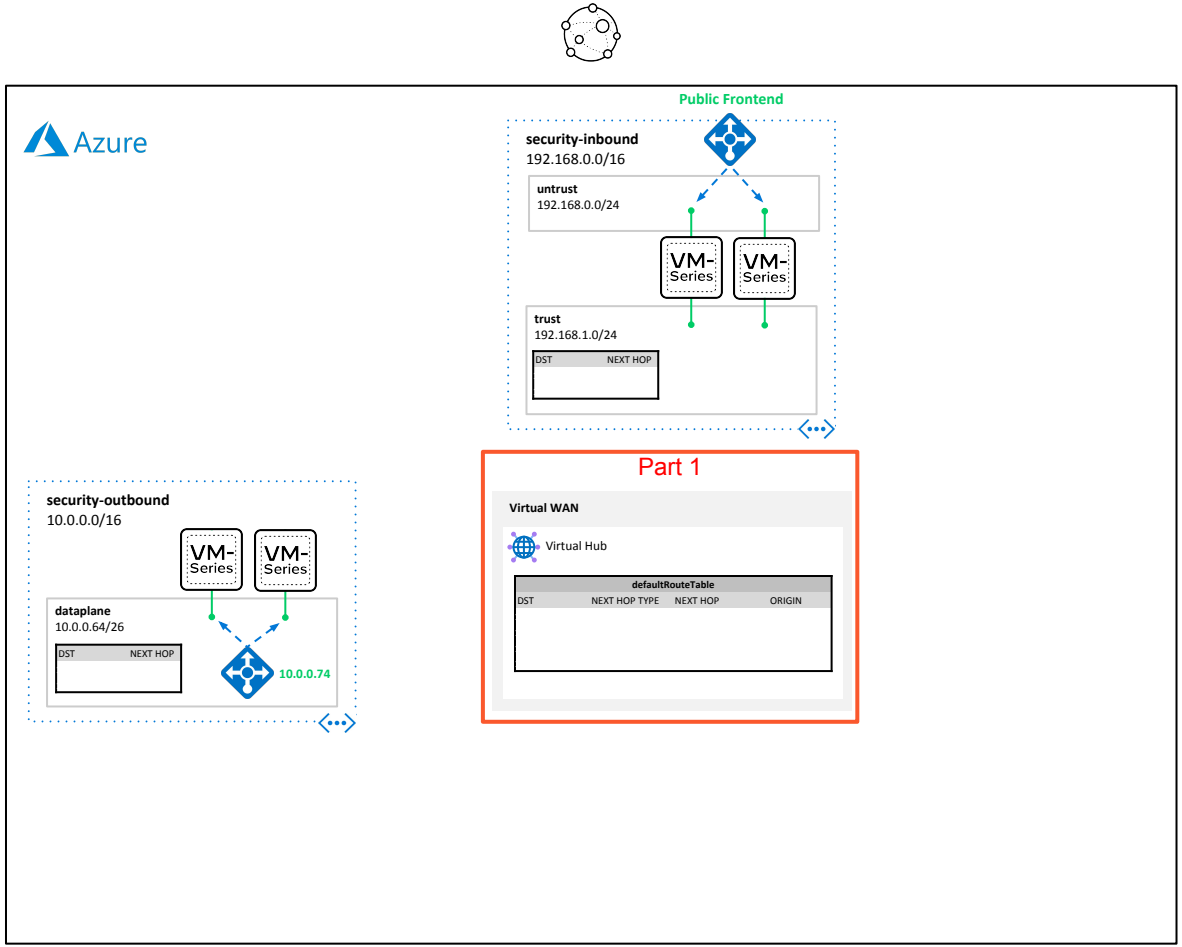
1. Continue to Part 1 once the deployment statuses' list as successful.



Part 1
Create Virtual WAN & Virtual Hub

Part 1: Overview

- Part 1 creates a Virtual WAN and a Virtual Hub with a default route table.



Part 1: Template Parameter Values

Parameter Values

Resource Group: Create a new resource group

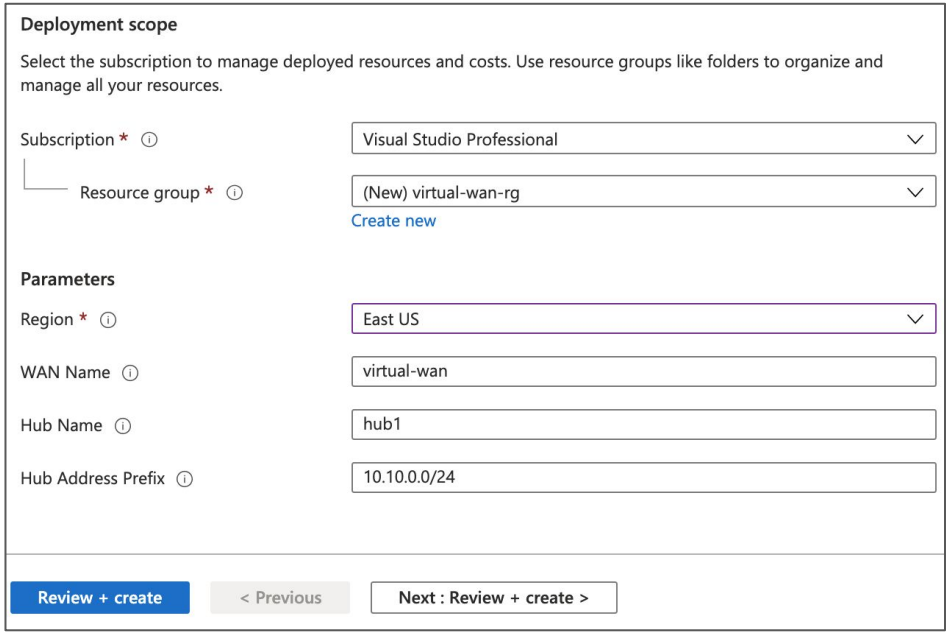
Region: Select a region

WAN Name: A name for the Virtual WAN

Hub Name: A name for the Virtual Hub

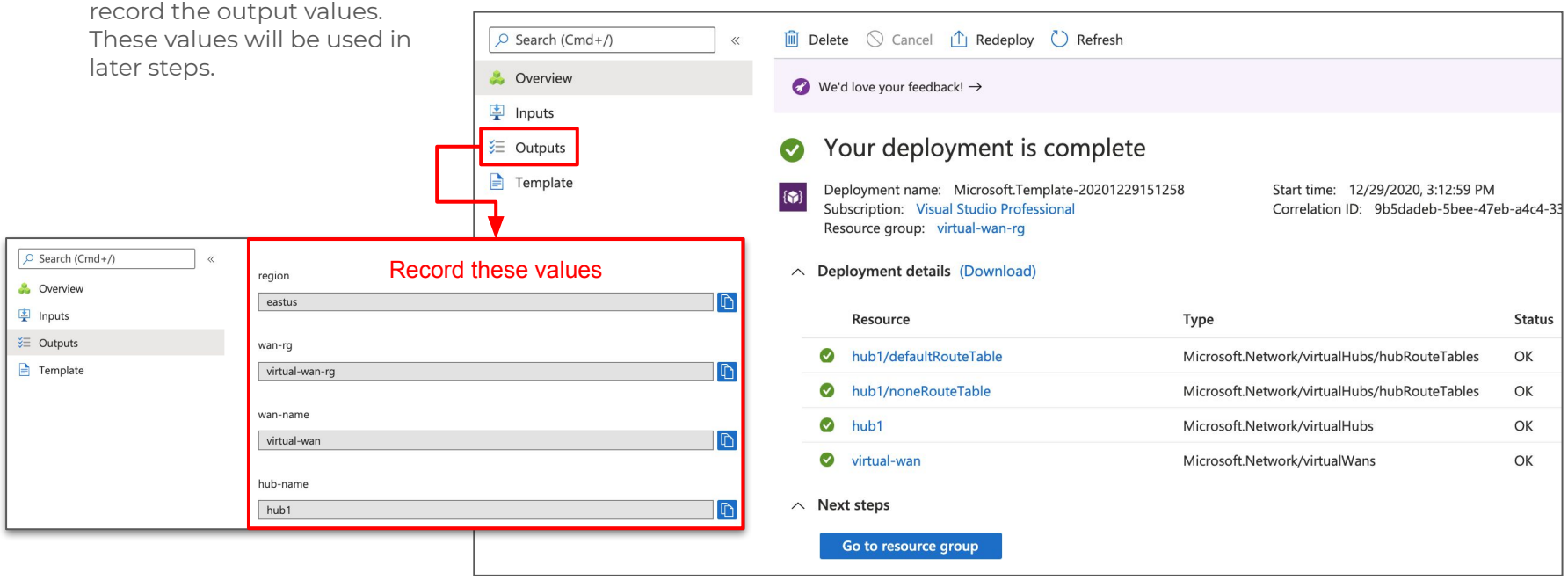
Hub Address Prefix: A unique CIDR range that does not overlap with other networks.

Click ***Review + Create***



Part 1: Completion Verification (1)

1. When the build completes, record the output values. These values will be used in later steps.

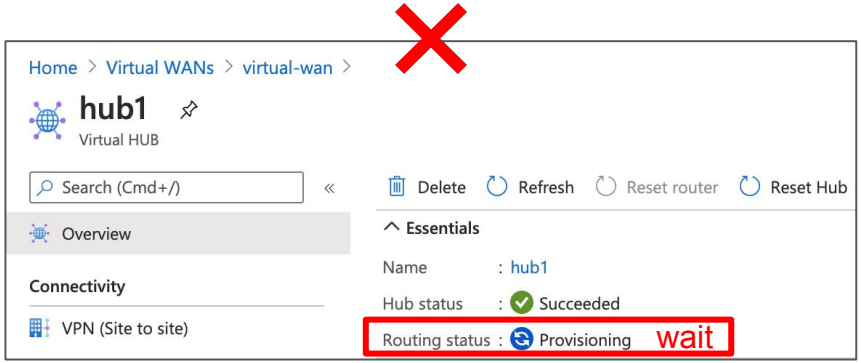


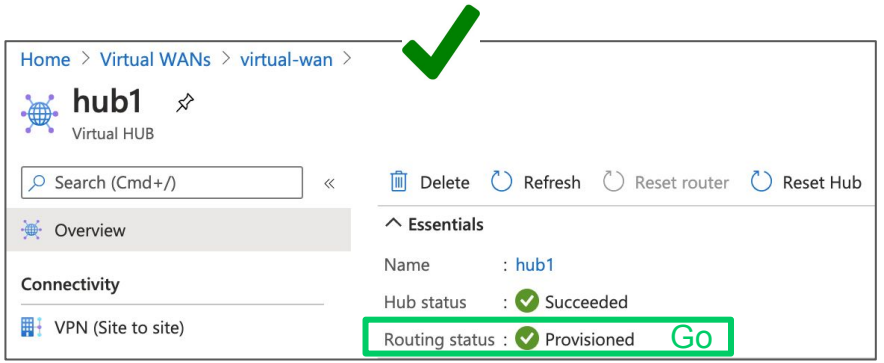
Part 1: Completion Verification (2)

Before proceeding, the Virtual Hub's router status *must* be in its "Succeeded" state.

This can take up to 30 minutes.

To check its status: *Azure Portal → Virtual WANs → Hub → Overview*

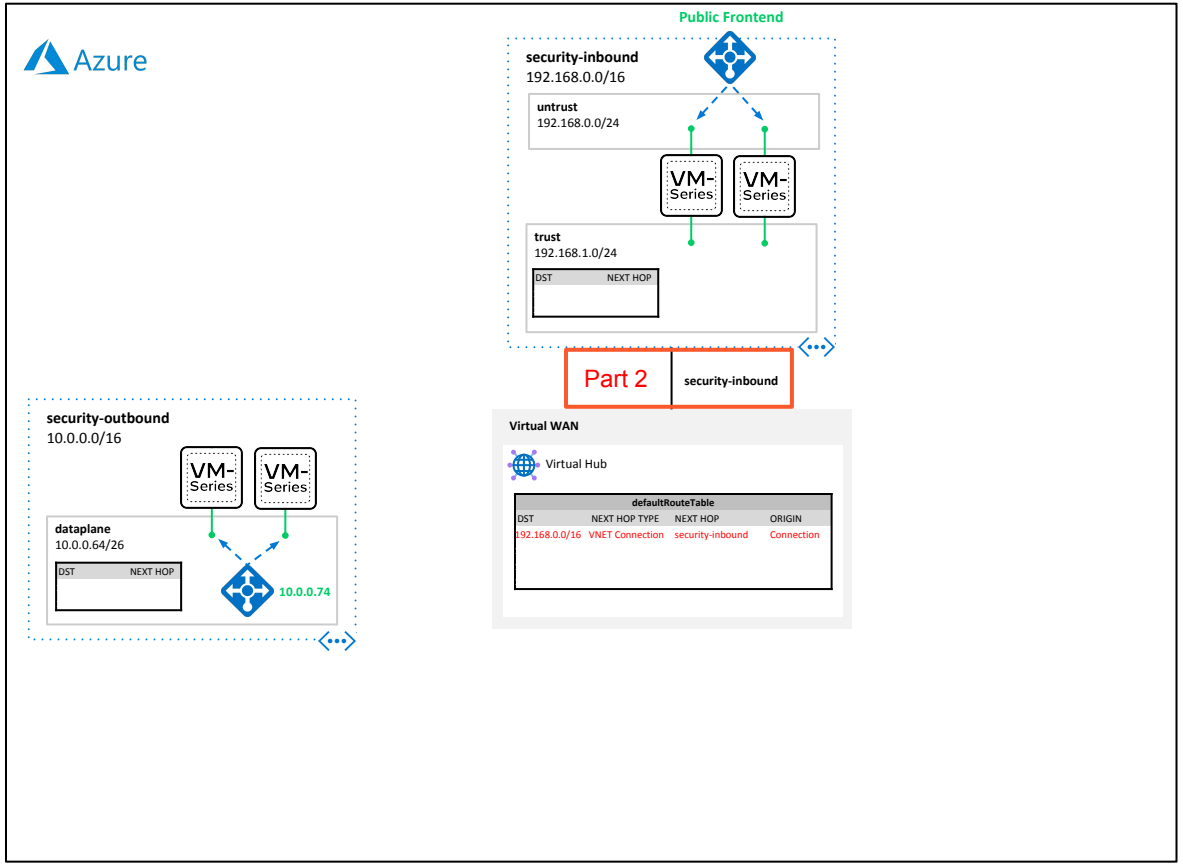




Part 2
Connect VM-Series Inbound Scale Set to the Virtual Hub

Part 2: Overview

- In this section, we will connect the inbound VM-Series VNET to the virtual hub that was created in Part 1.



Part 2: Template Parameter Values

Parameter Values

Resource Group: Select inbound VNET resource group

WAN Resource Group: WAN resource group (**Part 1 wan-rg output**)

Security VNET Name: Name of inbound VM-Series VNET

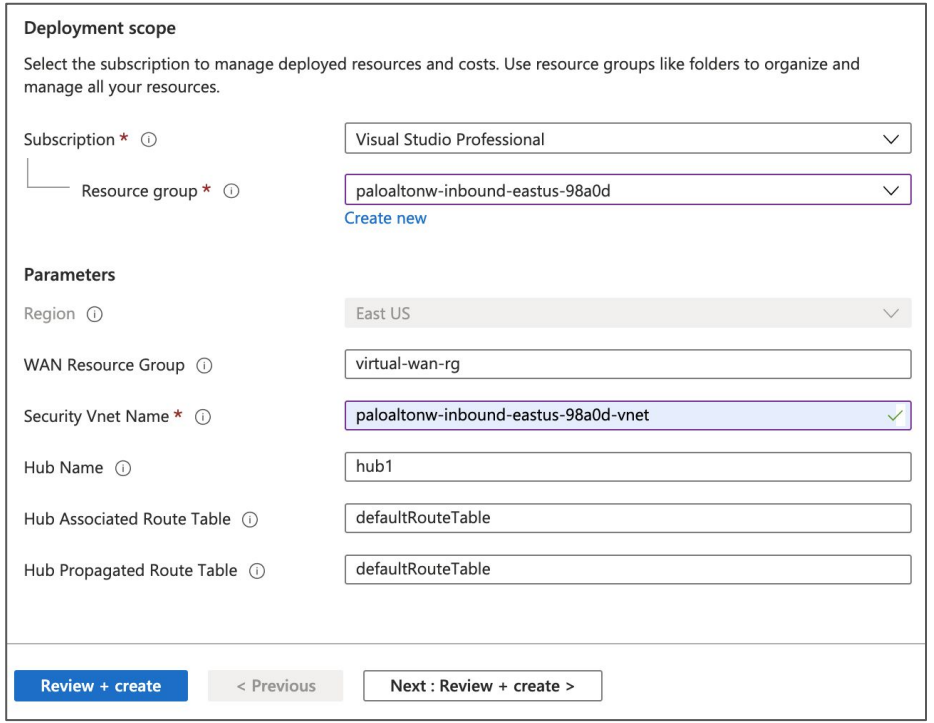
Security VNET Resource Group: Name of inbound VM-Series resource group

Hub Name: Virtual WAN hub name (**Part 1 hub-name output**)

Hub Associated Route Table: Leave as *defaultRouteTable*

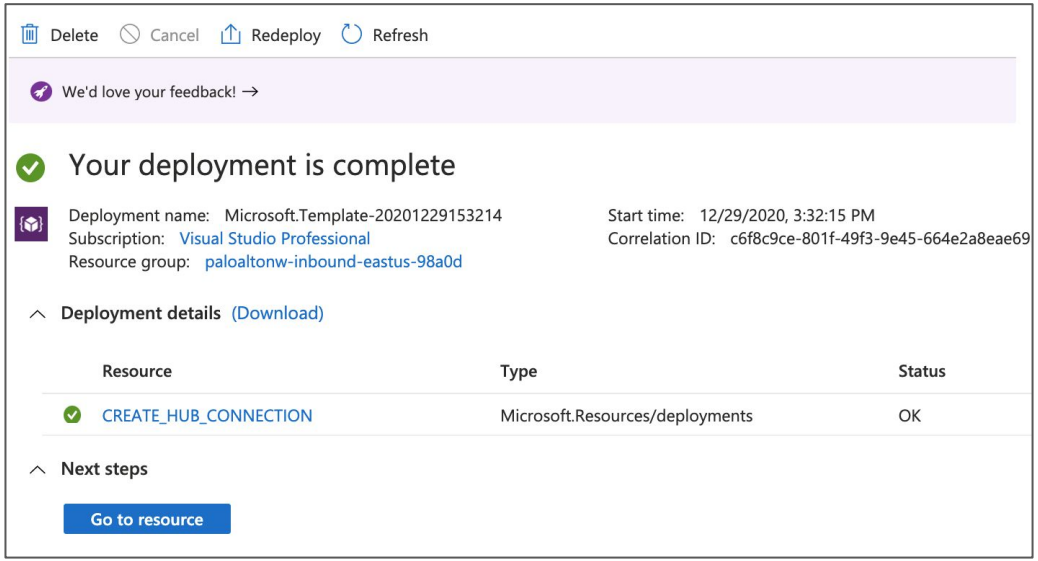
Hub Propagated Route Table: Leave as *defaultRouteTable*

Click ***Review and Create***



Part 2: Completion Verification

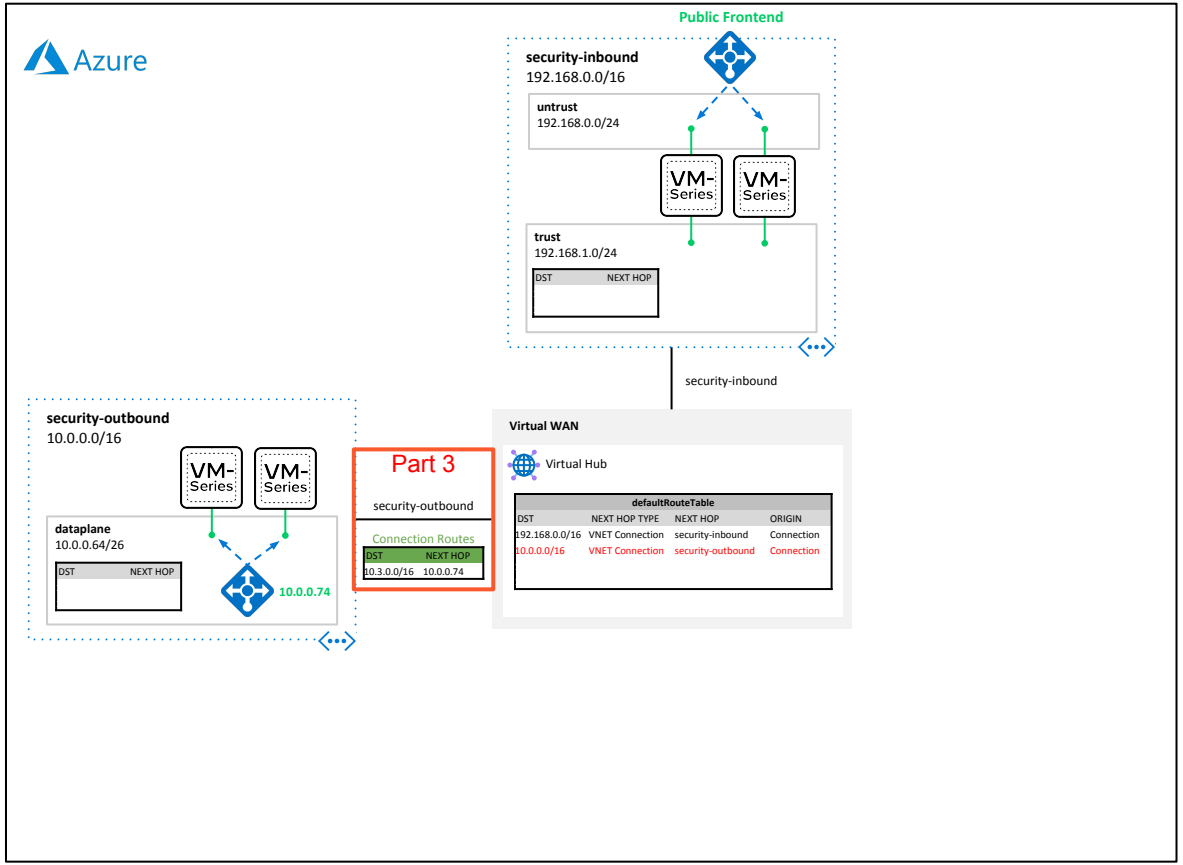
1. Proceed to Part 3 when the build completes.



Part 3
Connect VM-Series Outbound Scale Set to the Virtual Hub

Part 3: Overview

- In this section, we will connect the outbound VM-Series VNET to the virtual hub that was created in Part 1.

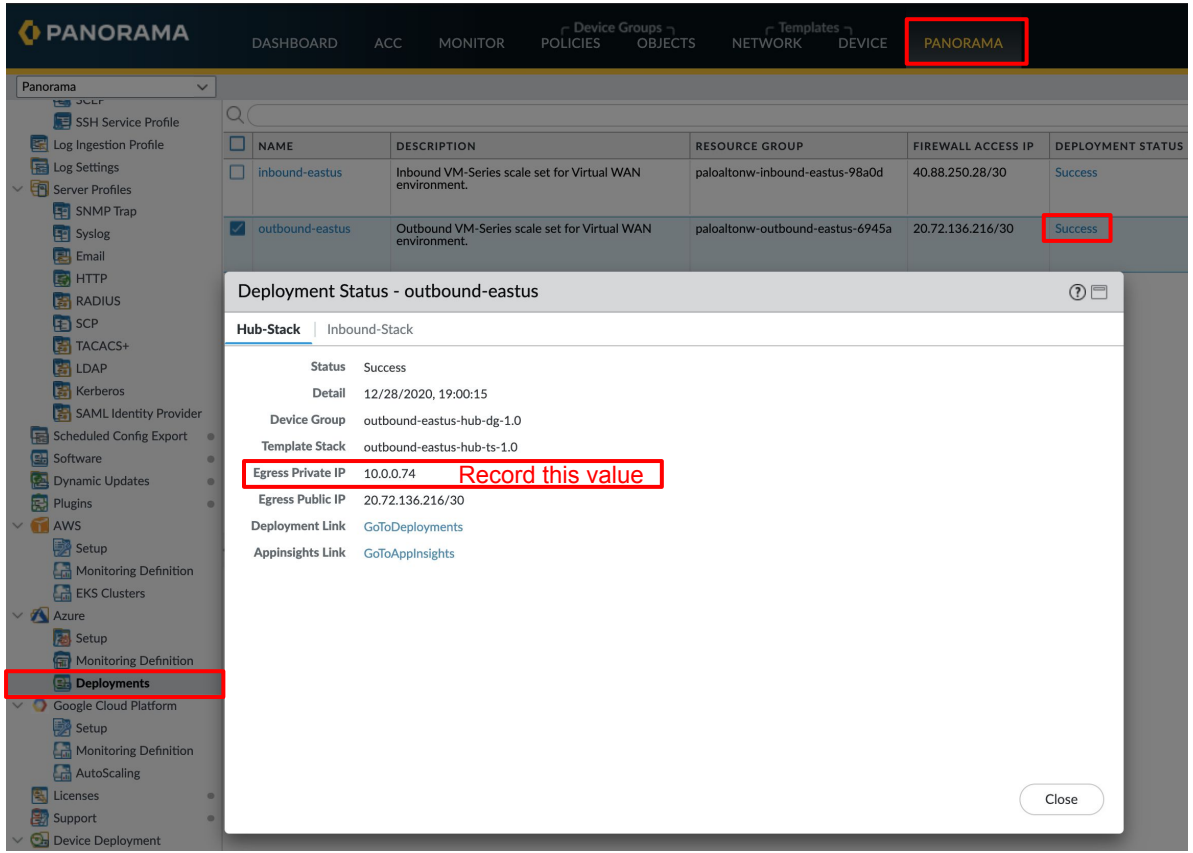


 |

Part 3: Record Outbound Deployment's Internal Load Balancer IP

1. Login to Panorama

Open *Panorama* tab
→ Azure Plugin
→ Deployments
→ Open Outbound Deployment
→ Copy ***Egress Private IP*** value



Part 3: Template Parameter Values

Parameter Values

Resource Group: Select outbound VM-Series resource group

WAN Resource Group: WAN resource group (**Part 1 wan-rg output**)

Security VNET Name: Name of outbound VM-Series VNET

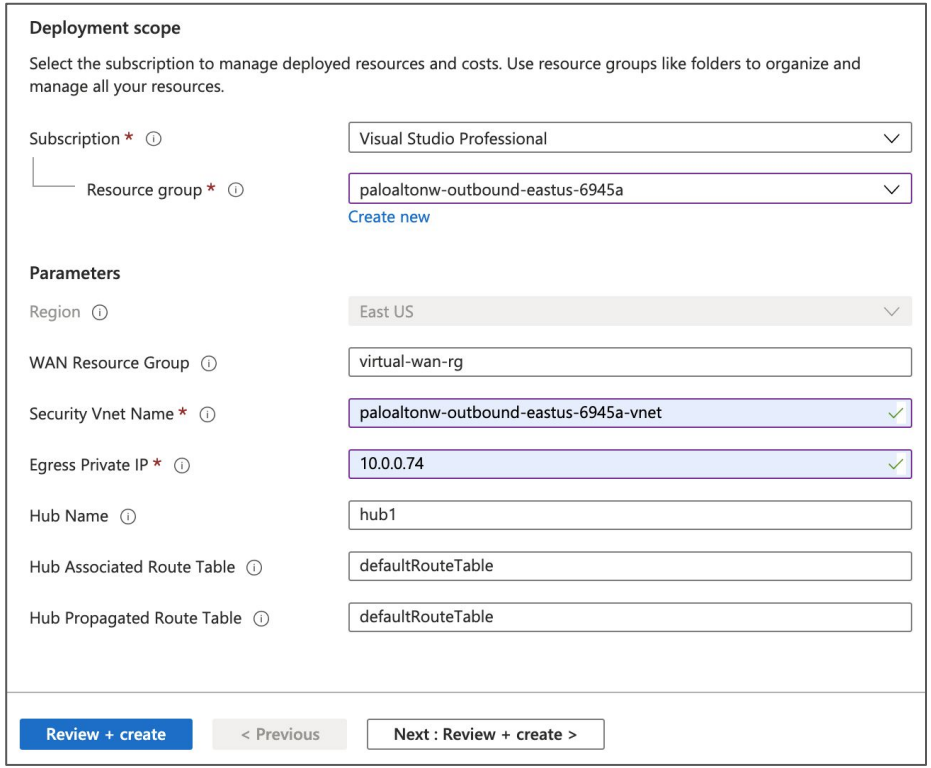
Egress Private IP: The IP Address of internal load balancer

Hub Name: Virtual WAN hub name (**Part 1 hub-name output**)

Hub Associated Route Table: Leave as *defaultRouteTable*

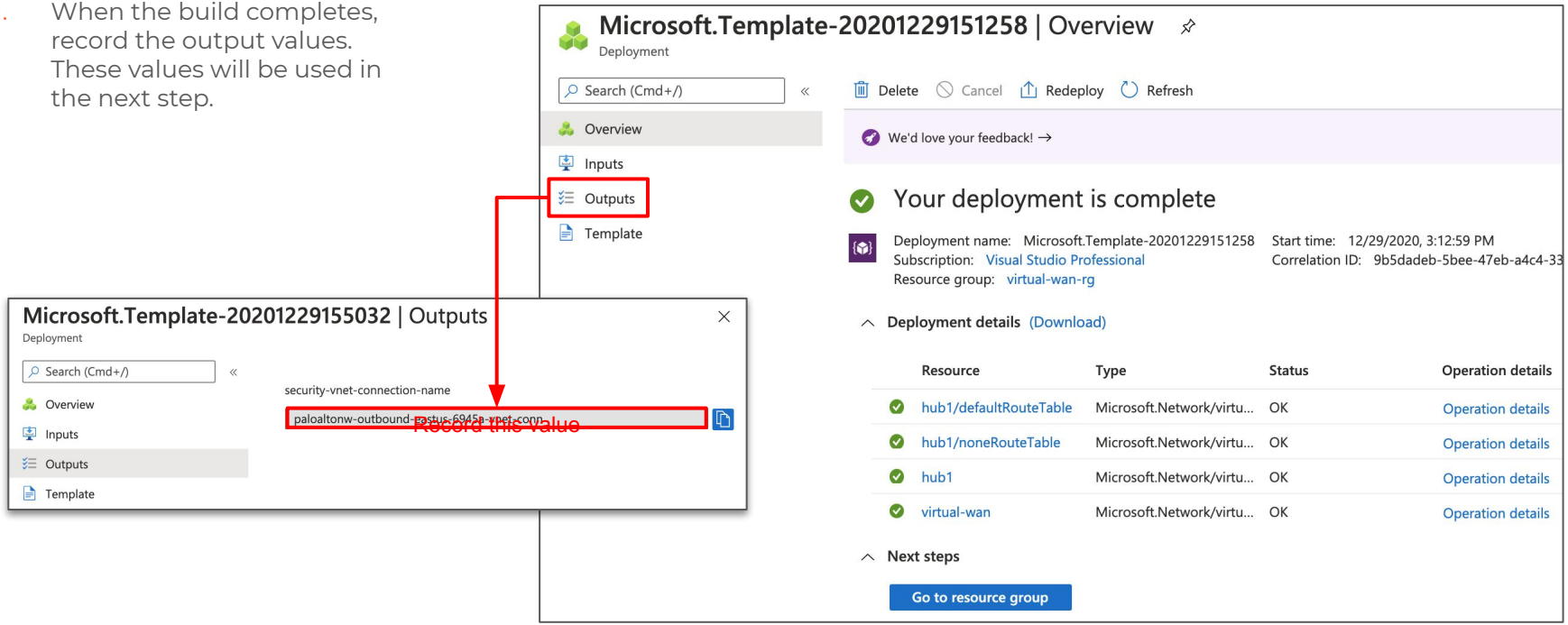
Hub Propagated Route Table: Leave as *defaultRouteTable*

Click ***Review + Create***



Part 3: Completion Verification (1)

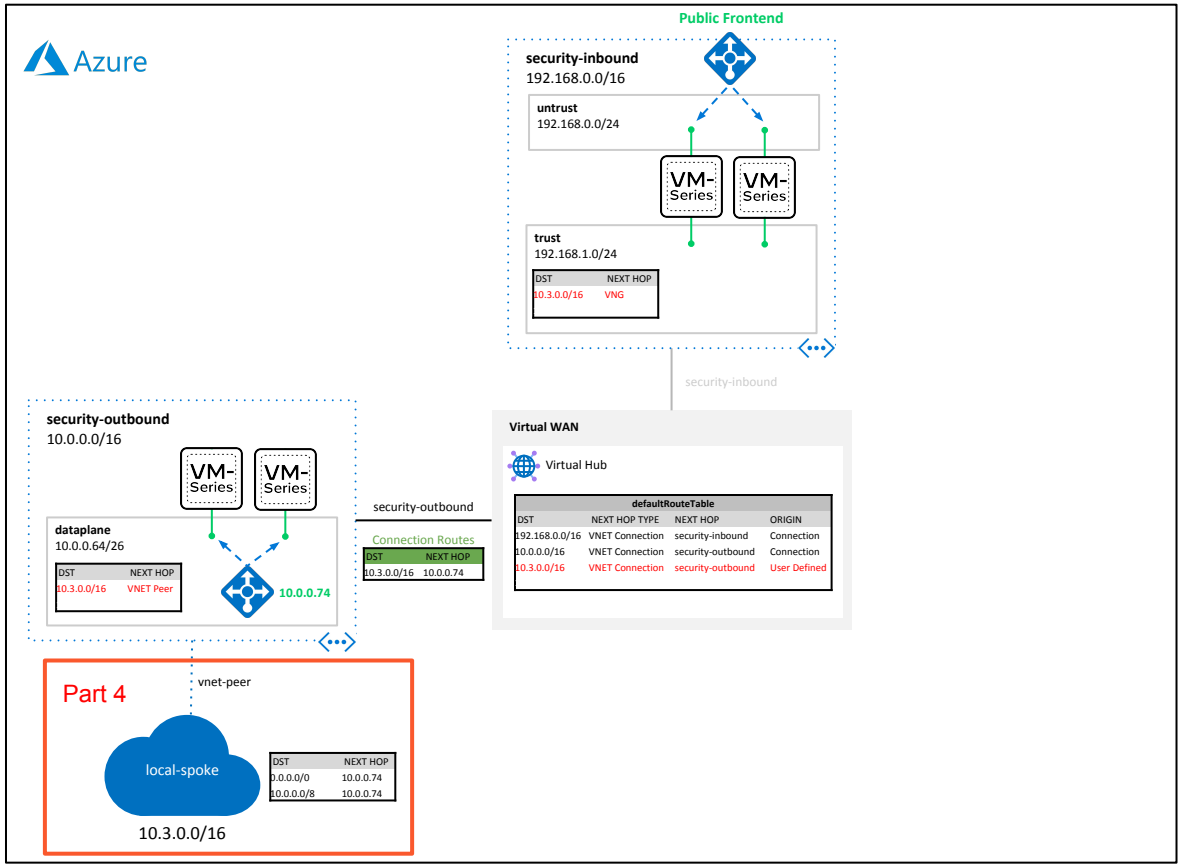
1. When the build completes, record the output values. These values will be used in the next step.



Part 4
Peer Local VNET to VM-Series Outbound VNET

Part 4: Overview

- In this section, we will create a spoke VNET that will be VNET peered to the outbound VNET.
- A route will be added to the virtual hub route table to direct Virtual WAN traffic destined to the local spoke VNET through the outbound firewalls.
- An Ubuntu VM will be deployed in the VNET that will be used to test traffic flows in later steps.



 |

Part 4: Template Parameter Values

Parameter Values

Resource Group: Create a new resource group

Region: Same region as previous deployments.

Security VNET Resource Group: Name of outbound VM-Series resource group

Security VNET Name: Name of outbound VM-Series VNET

Security VNET Connection Name: **Part 3 security-vnet-connection-name output**

Egress Private IP: The IP Address of internal load balancer

WAN Resource Group: WAN resource group (**Part 1 wan-rg output**)

Hub Name: Virtual WAN hub name (**Part 1 hub-name output**)

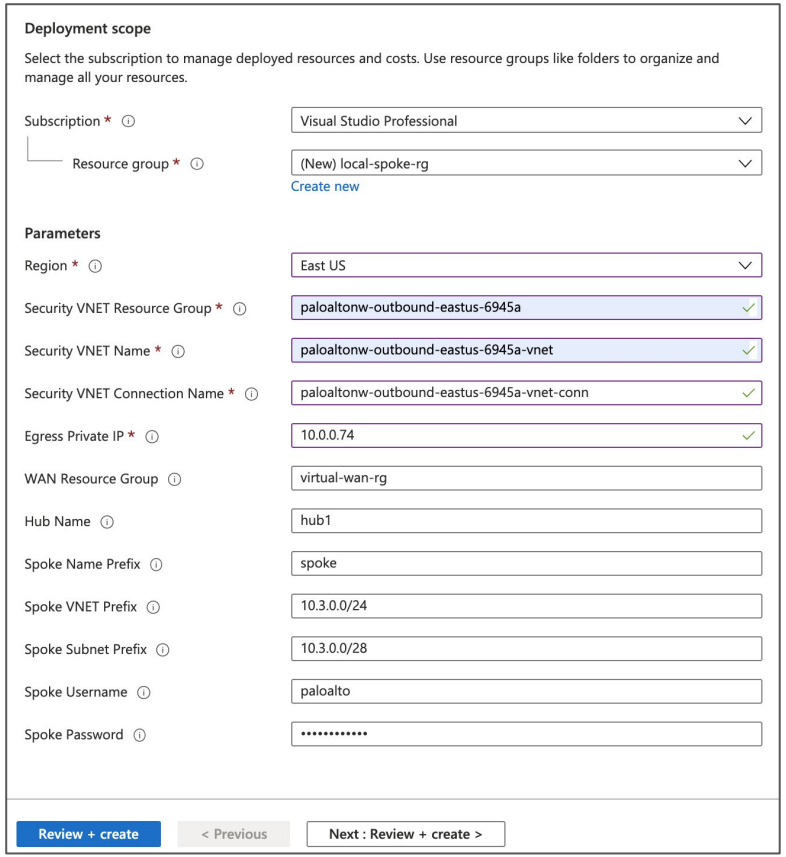
Spoke Name Prefix: Any name to append to created resources.

Spoke VNET Prefix: Unique CIDR Block for the spoke VNET.

Spoke Subnet Prefix: Unique subnet CIDR for the spoke subnet.

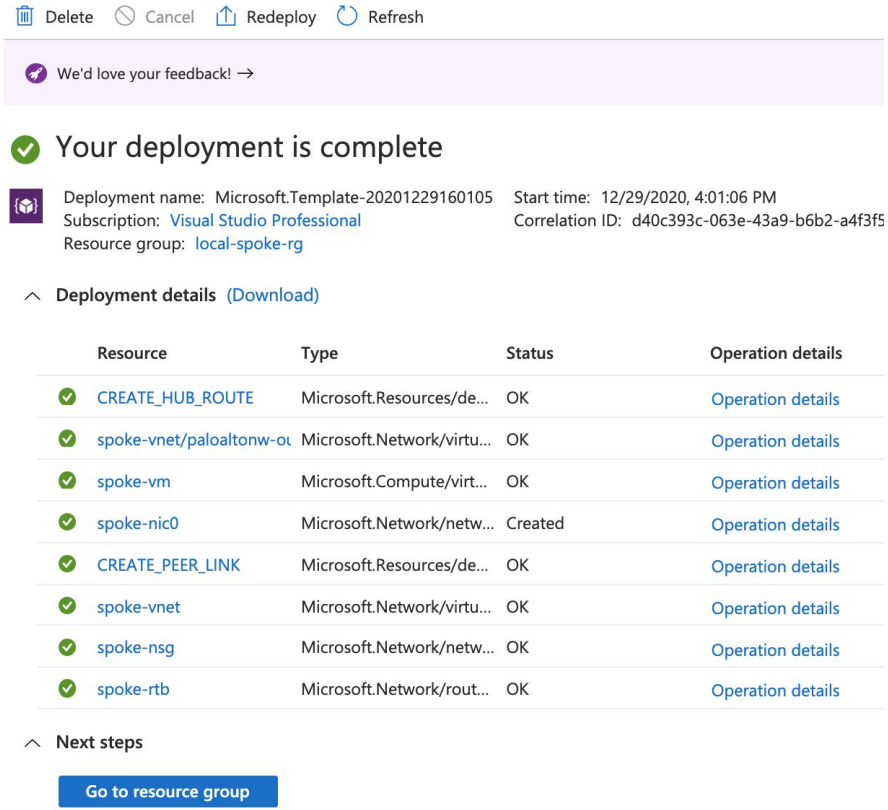
Spoke Username/Password: Credentials for Ubuntu VM.

Click ***Review + Create***



Part 4: Completion Verification

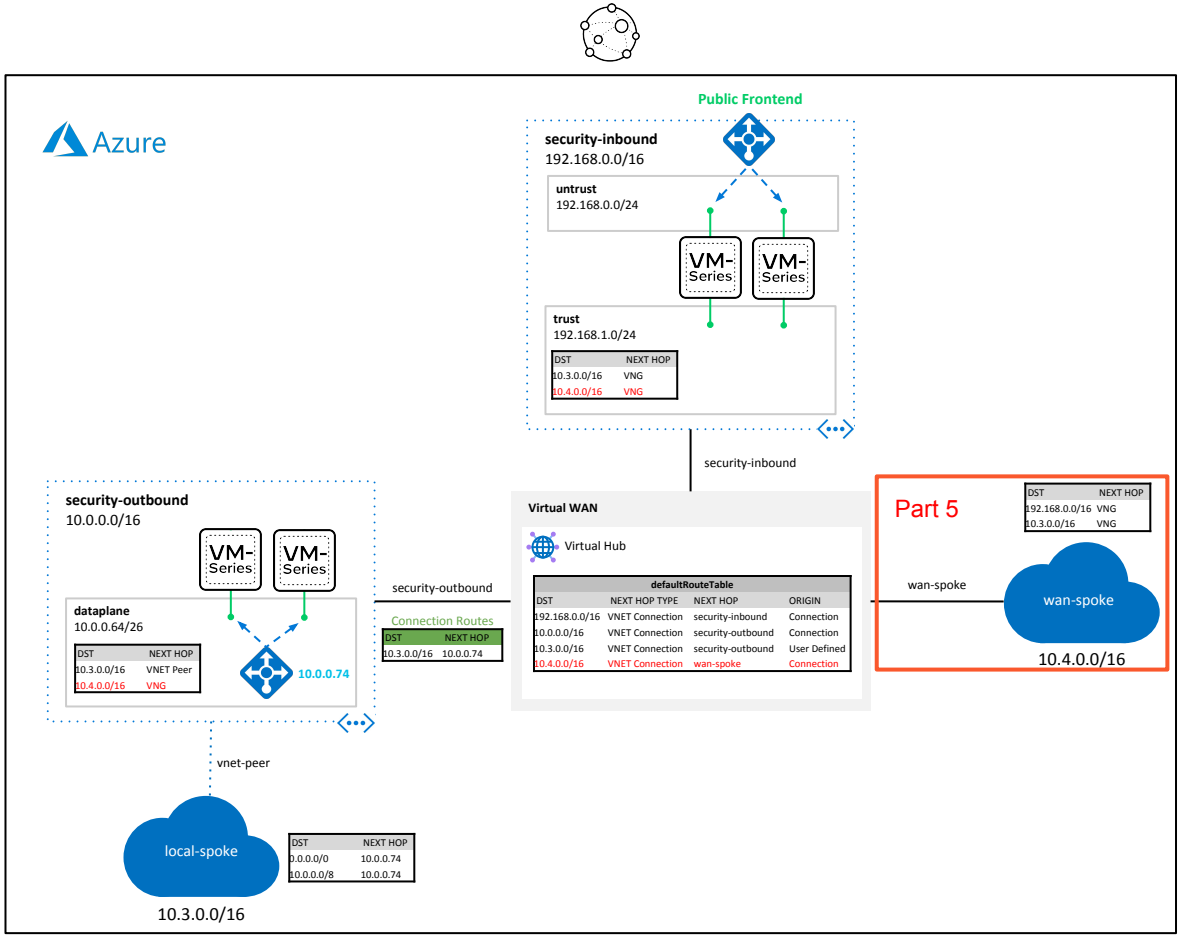
1. Proceed to Part 5 when the build completes.



Part 5
Connect WAN Spoke VNET to Virtual Hub

Part 5: Overview

- Part 5 creates a Virtual Network and connects it to the Virtual Hub via a VNET Connection.
- An Ubuntu VM is deployed in the VNET that will be used to test inbound and lateral flows.



Part 5: Template Parameter Values

Parameter Values

Resource Group: Create a new resource group

Region: Same region as previous deployments.

Spoke Name Prefix: A unique name to prepend to the created resources

Spoke VNET Prefix: Unique CIDR Block for the VNET.

Spoke Subnet Prefix: Unique subnet CIDR for the test subnet.

Spoke Username/Password: Credentials for Ubuntu VM.

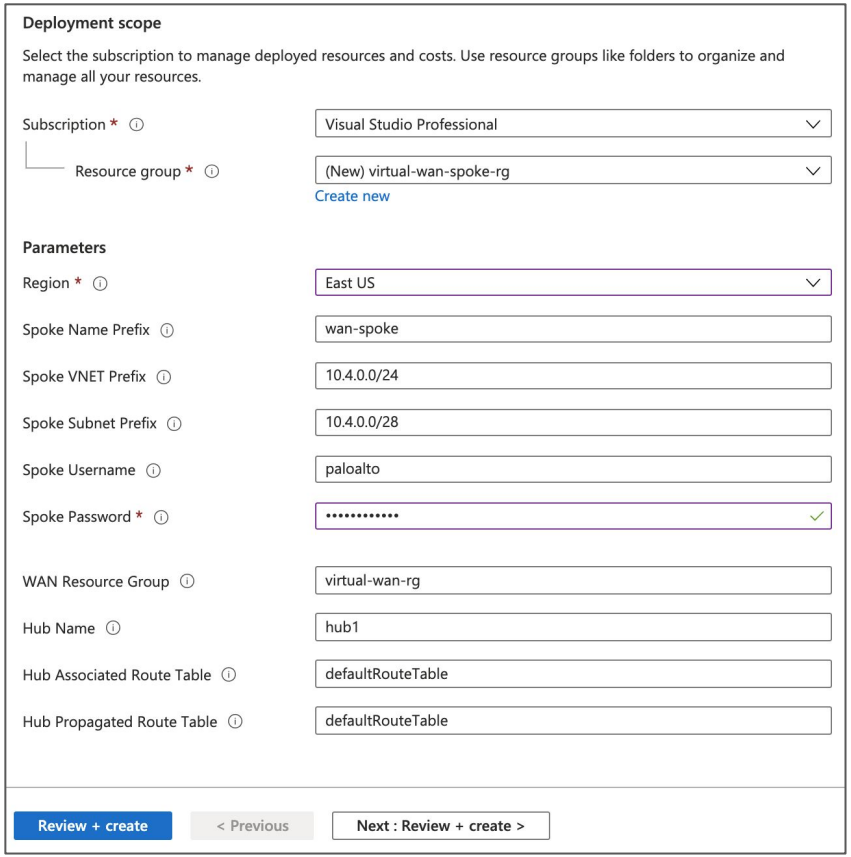
WAN Resource Group: WAN resource group (**Part 1 wan-rg output**)

Hub Name: Virtual WAN hub name (**Part 1 hub-name output**)

Hub Associated Route Table: Leave as *defaultRouteTable*

Hub Propagated Route Table: Leave as *defaultRouteTable*

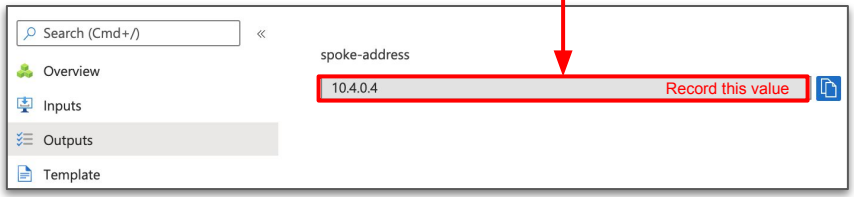
Click ***Review + Create***

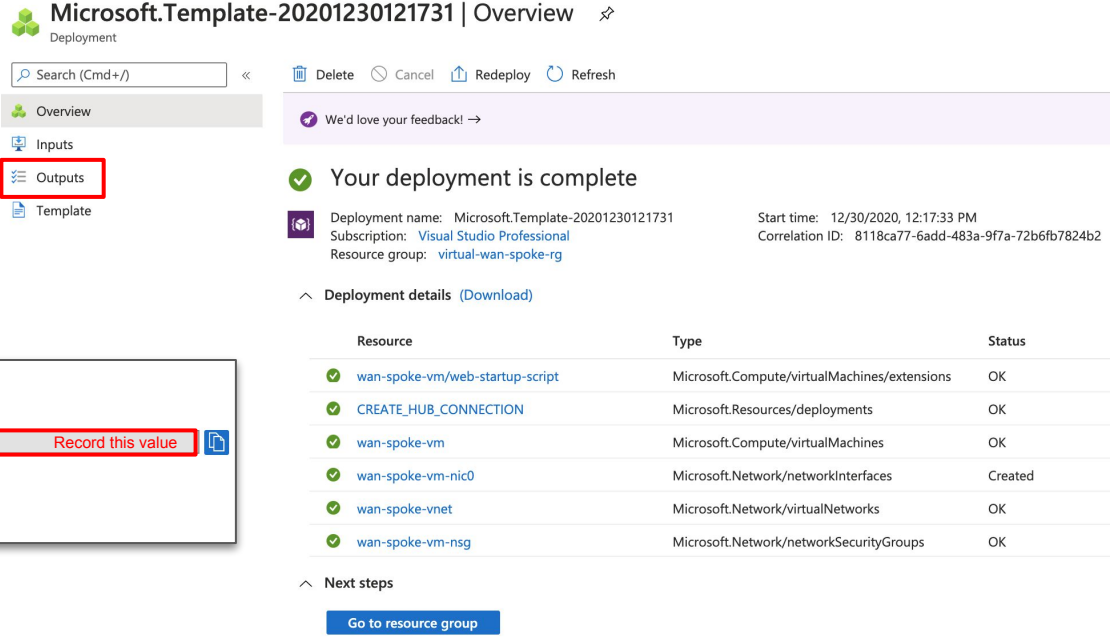


Part 5: Completion Verification

1. When the build completes, you will see this page from the portal.

2. Click ***Outputs*** and record the *spoke-address* value. It will be used in Part 3 for testing the inbound flow.





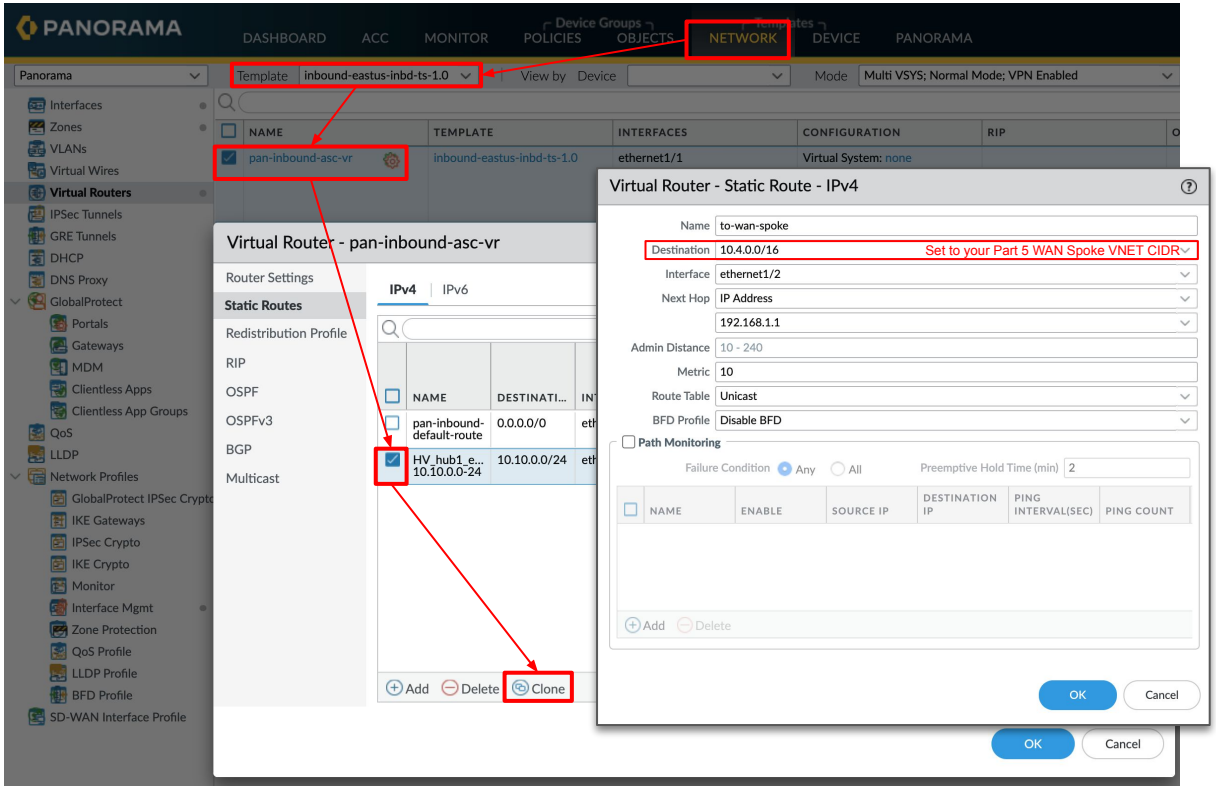
Part 6
Test Flows

Inbound Config - Add Route for vWAN Spoke VNET

Login to Panorama

Open *Network* tab
→ Select your inbound ***template stack***
→ Click *Virtual Routers*
→ Click *pan-inbound-asc-vr*
→ Click *Static Routes*

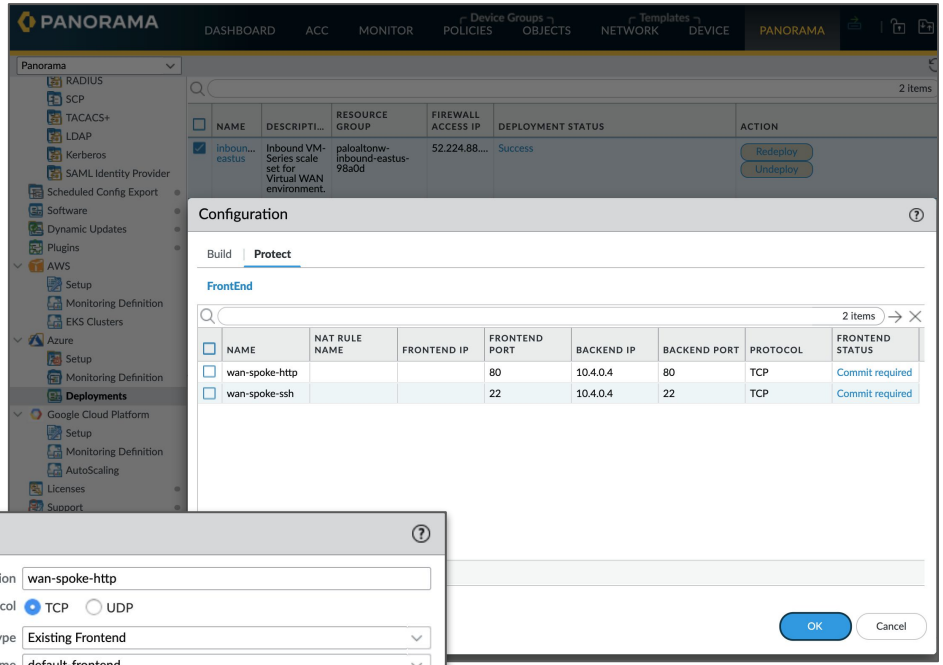
1. Clone the ***HV_hub_**** route
2. Change route destination to your WAN Spoke VNET prefix from Part 5.
3. Click OK

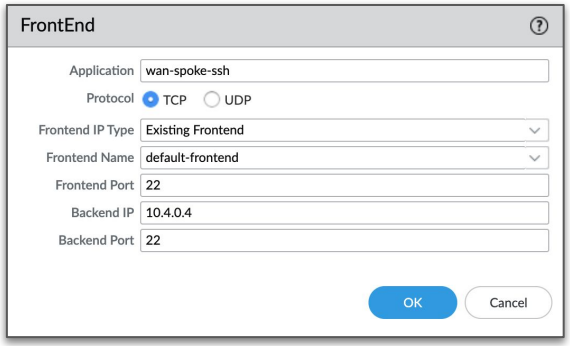


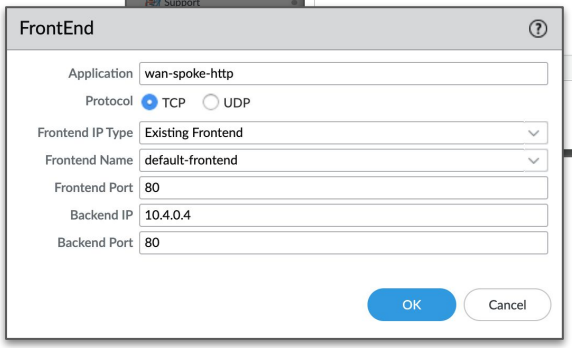
 |

Inbound Config - Add WAN Spoke Application

1. Open the *Panorama* Tab and go to Deployments within the Azure plugin
2. Open the inbound deployment and click to the *Protect* tab.
3. Create 2 protected applications.
    1. Inbound HTTP to the WAN Spoke VM.
    2. Inbound SSH to the WAN Spoke VM.
2. **Use Part 5 *spoke-address* output for the Backend IP value**

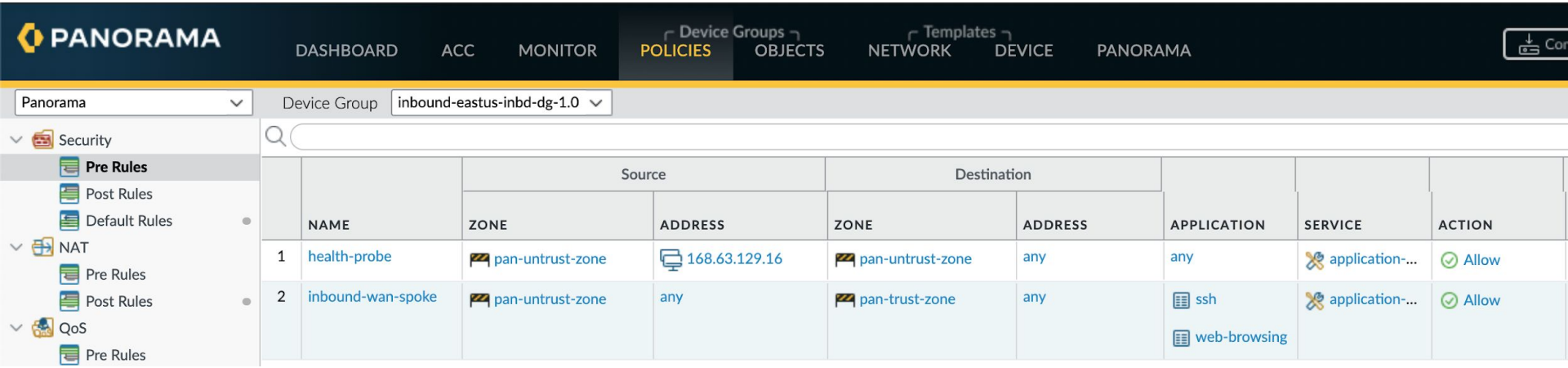






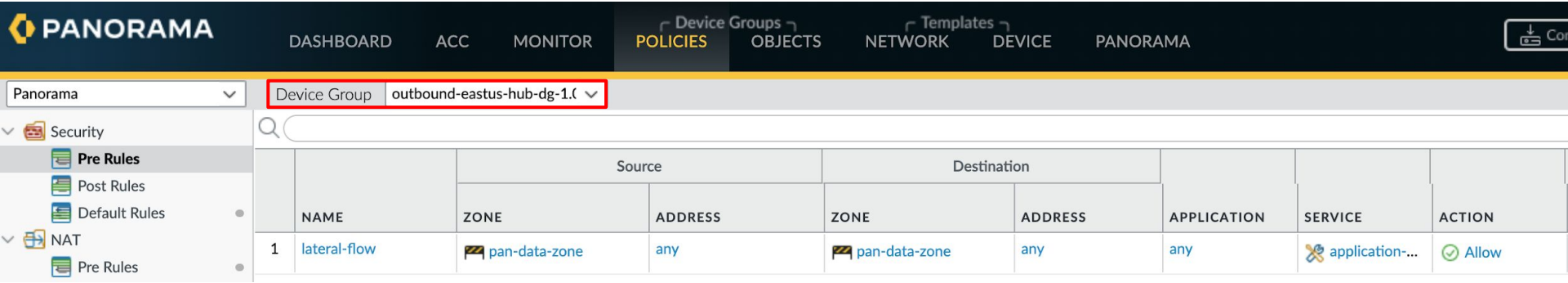
Inbound Config - Add Security Policies

1. Go to Policies tab and select the inbound Device Group.
2. Within security pre-rules create 2 policies as seen below.
   a. **health-probe**: allows the public load balancers health probes to traverse the firewalls.
   b. **inbound-wan-spoke**: allows inbound HTTP and SSH to the spoke network created in Part 5.



Outbound Config - Add Security Policies

1. Go to Policies tab and select the outbound Device Group.
2. Within security pre-rules create a policy to allow traffic to flow.

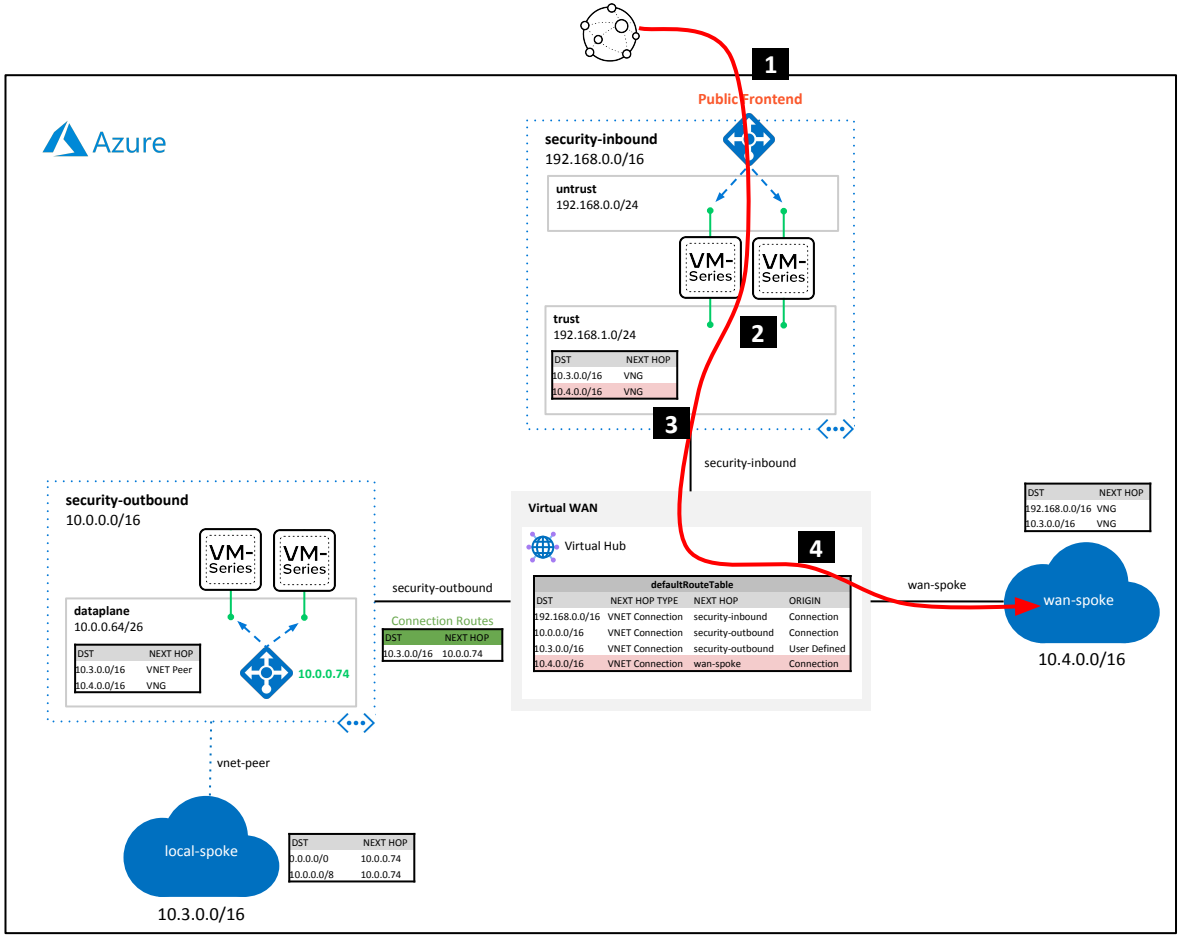


Push & Commit Changes

Test Inbound Flow

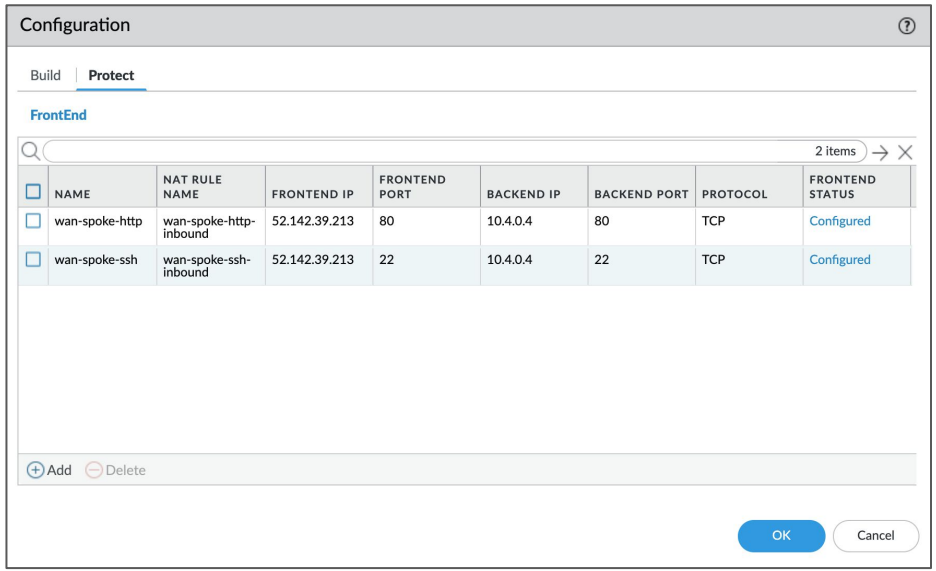
Inbound Flow

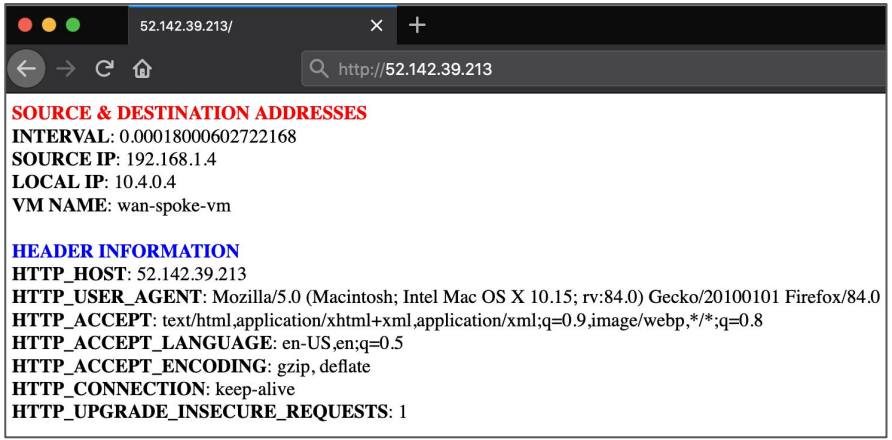
1. Inbound request hits public load balancer frontend IP.
2. VM-Series SNAT to trust interface and DNAT to destination (**10.4.0.4**).
3. Trust subnet sends traffic to vWAN hub via propagated route.
4. vWAN hub route table sends request to wan-spoke VNET connection.



Add WAN Spoke Application

1. Go to back to the Protect tab within the inbound deployment.
2. Copy the Frontend IP address.
3. Paste `http://<frontend-ip>` into your web browser.

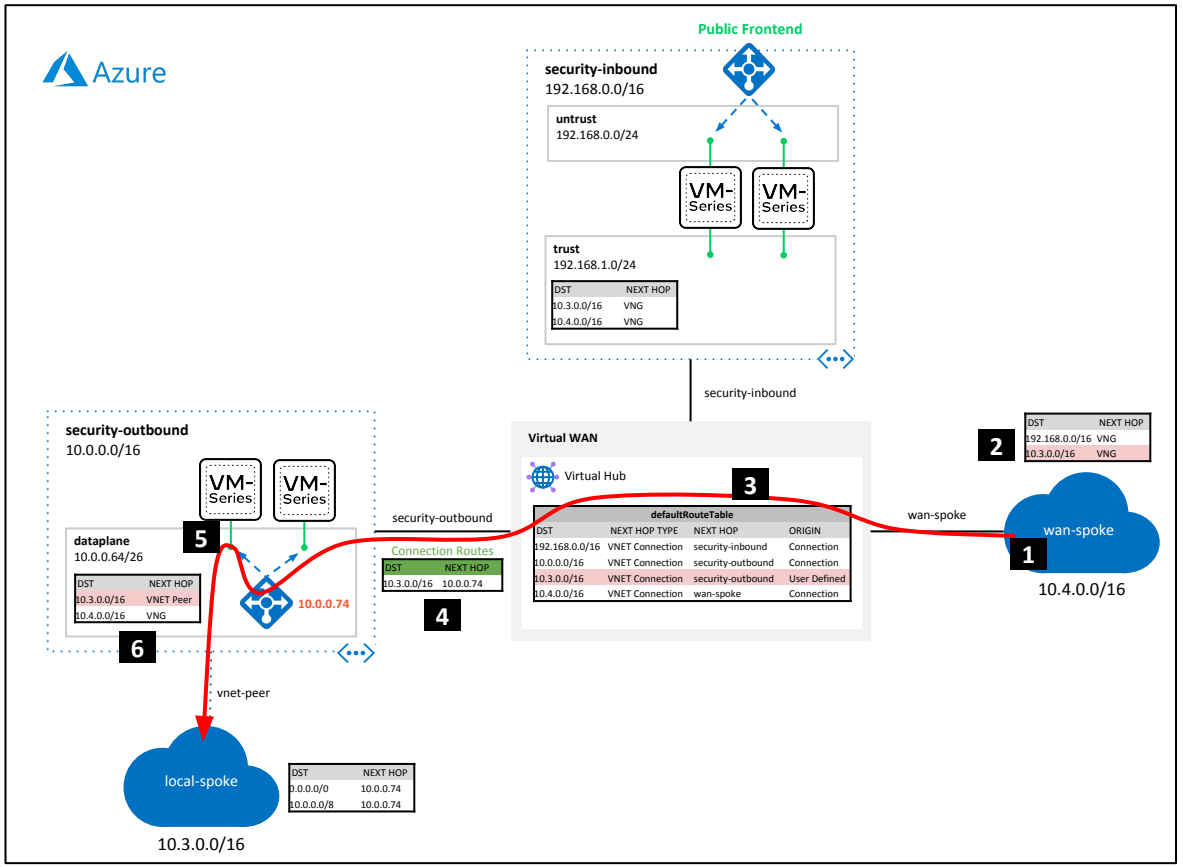




Test Outbound Flow

Lateral Flow

1. wan-spoke VNET makes request to local-spoke (**10.3.0.x)**
2. Spoke subnet sends request to vWAN hub via propagated route.
3. vWAN hub route table has user defined route to direct request to the security-outbound VNET connection.
4. The security-outbound VNET connection has a “connection-route” to direct request to the internal load balancer of the VM-Series (**10.0.0.74**).
5. Internal load balancer forwards traffic to VM-Series
6. VM-Series dataplane subnet sends request to its local peer (**10.3.0.x**)



Add WAN Spoke Application

1. Use the same IP to SSH to the WAN spoke VM. Use the username/password from Part 5.

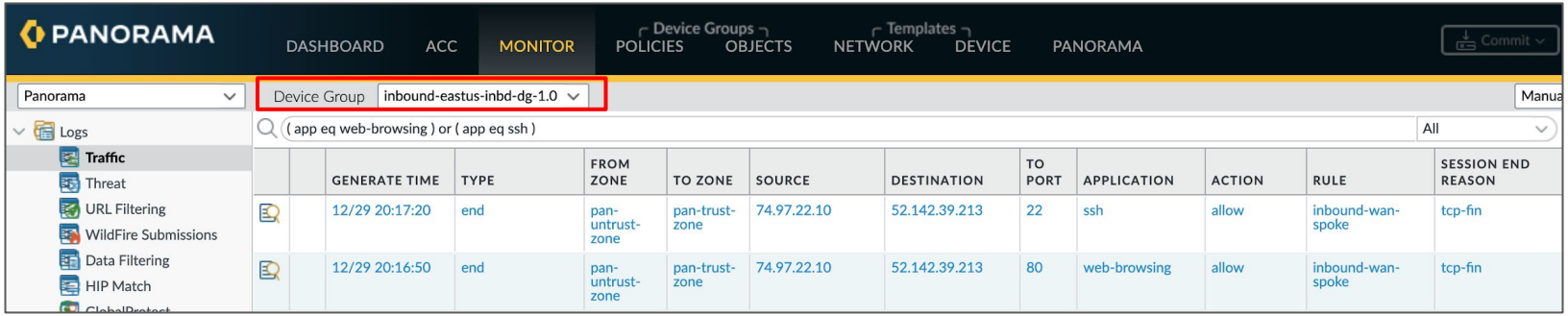
   ```
   ssh <user>@<your-frontend-ip>
   ```

2. Ping to the locally peered spoke (created in Part 4).

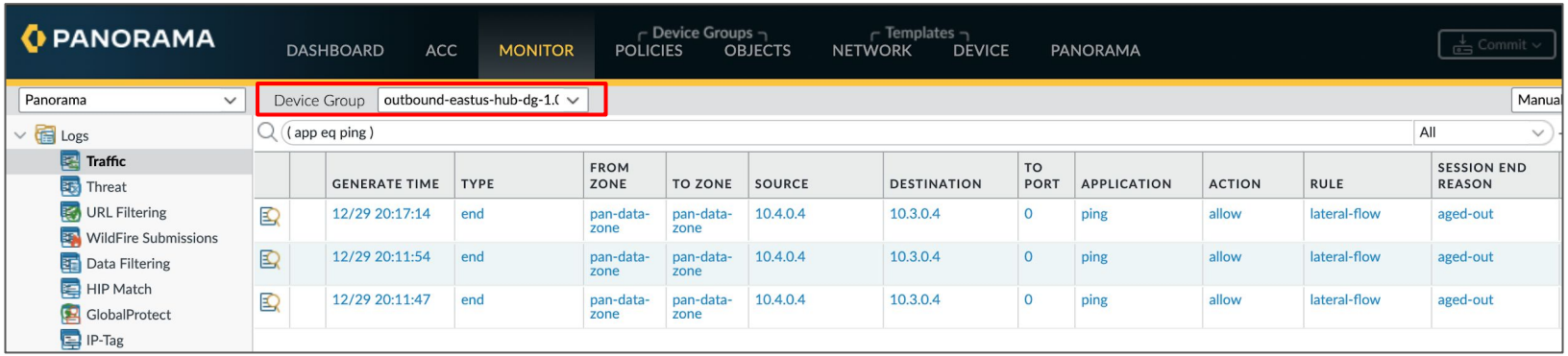
```
~ $ ssh paloalto@52.142.39.213
paloalto@52.142.39.213's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 5.4.0-1031-azure x86_64)

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

  System information as of Wed Dec 30 01:11:25 UTC 2020

  System load:  0.0               Processes:           118
  Usage of /:   5.2% of 28.90GB   Users logged in:     0
  Memory usage: 24%               IP address for eth0: 10.4.0.4
  Swap usage:   0%

20 packages can be updated.
16 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.

paloalto@wan-spoke-vm:~$ ping 10.3.0.4
PING 10.3.0.4 (10.3.0.4) 56(84) bytes of data.
64 bytes from 10.3.0.4: icmp_seq=1 ttl=62 time=6.12 ms
64 bytes from 10.3.0.4: icmp_seq=2 ttl=62 time=4.17 ms
64 bytes from 10.3.0.4: icmp_seq=3 ttl=62 time=3.70 ms
64 bytes from 10.3.0.4: icmp_seq=4 ttl=62 time=6.34 ms
64 bytes from 10.3.0.4: icmp_seq=5 ttl=62 time=4.49 ms
64 bytes from 10.3.0.4: icmp_seq=6 ttl=62 time=4.13 ms
64 bytes from 10.3.0.4: icmp_seq=7 ttl=62 time=3.62 ms
64 bytes from 10.3.0.4: icmp_seq=8 ttl=62 time=3.71 ms
64 bytes from 10.3.0.4: icmp_seq=9 ttl=62 time=3.92 ms
^C
--- 10.3.0.4 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8012ms
rtt min/avg/max/mdev = 3.626/4.471/6.344/0.978 ms
paloalto@wan-spoke-vm:~$
```


Add WAN Spoke Application

1. Go to the monitor tab
2. Filter for the HTTP & SSH logs within the inbound device group.



3. Filter for the ping logs within the outbound device group.



paloalto NETWORKS

Thank you