Hands-on Lab

Outbound HA in Azure

http://www.paloaltonetworks.com

Last Update: 20180101

Table of Contents

Lab Prerequisites – What you need

To complete this lab, you will need:

- A functioning computer
- An accessible Azure Portal account with a paid subscription
  - A free trial Azure subscription is not sufficient for this lab
- Permission to deploy and delete resources in your Azure Subscription.
- A functioning web browser
- SSH client to port 22.
- For the SSH client you can use the MAC Terminal. For Windows you can use Putty, SecureCRT etc.
- A text editor to keep notes

Please go through the checklist and verify that you have all the prerequisites necessary to participate in this lab. If you do not have all that is required, you can use this lab guide to go through the steps on your own once you have the prerequisites.

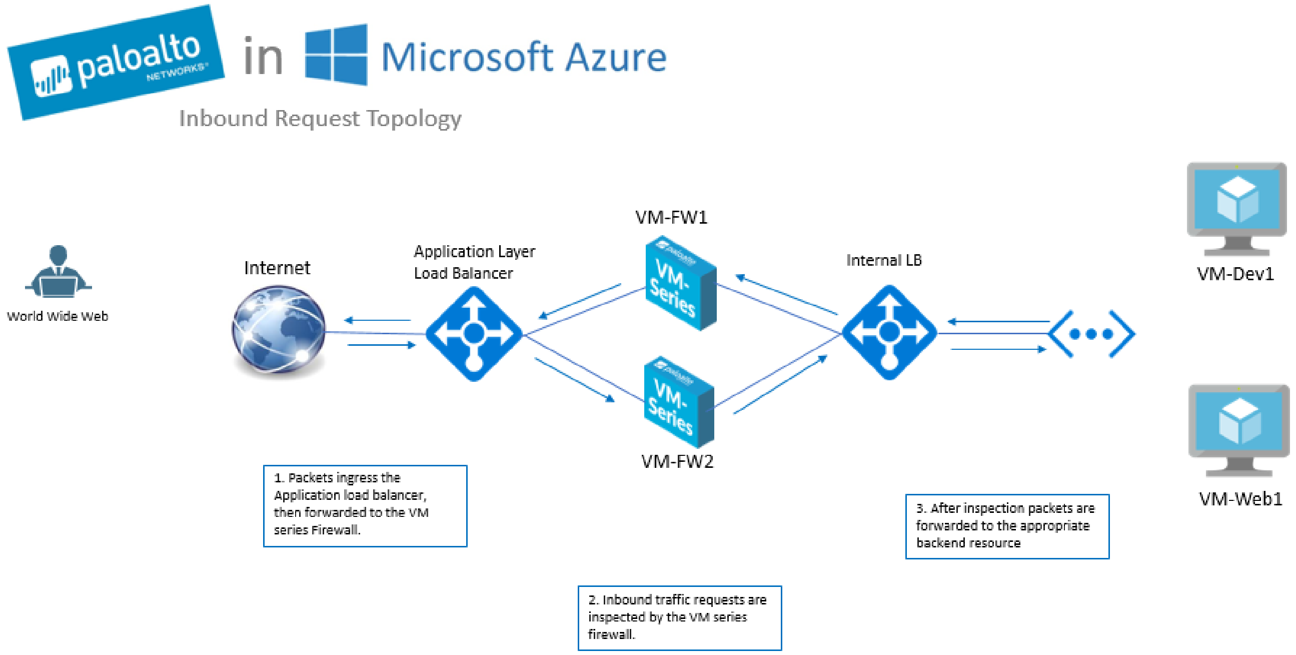
Lab Introduction – Outbound HA Overview

In this introduction, you will:

- Understand the benefit of using outbound high availability in Azure
- Understand the topological architecture used for this lab

Use Case – Why you need outbound HA in Azure

In cloud architecture when protecting public facing services, security planning is normally done for inbound requests. Palo Alto Networks Layer-7 inspection capabilities can be leveraged in these situations. Palo Alto Networks layer-7 capabilities allows for the securing of publicly exposed resources from threats masquerading as web traffic.

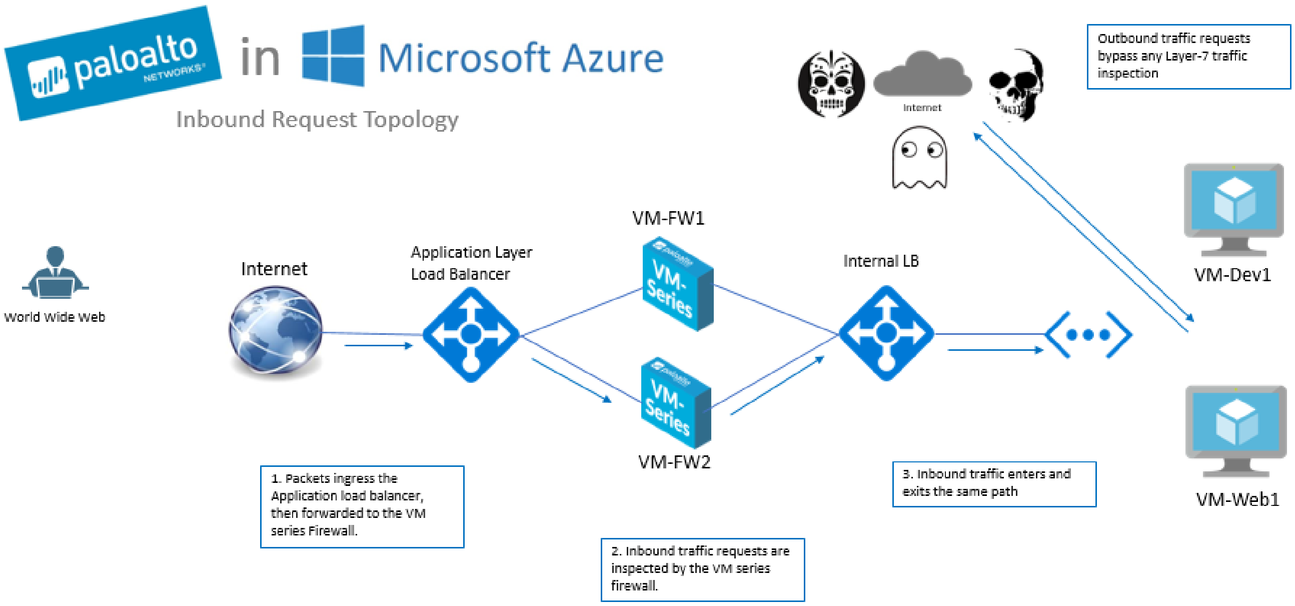


1. Packets ingress the Application load balancer, then forwarded to the VM series Firewall.

2. Inbound traffic requests are inspected by the VM series firewall.

3. After inspection packets are forwarded to the appropriate backend resource

Although the above topology is secure from an inbound traffic perspective, it's not a complete solution and here's why. Azure allows traffic to the internet without defining a user defined route (**UDR**) in the route table. What are **User Defined Routes**? UDR's are used to send traffic to a desired next hop and this will be demonstrated in the lab. Azure also allows outbound traffic requests to the internet without a public IP address. Although, this may be of benefit to some, this can implicitly cause vulnerabilities. In the diagram below, you can see that even without a public IP address the two web servers can still make requests to the internet through Azure networking. Due to the lack of Layer-7 visibility for outbound traffic, there is concern regarding malware exploits being pulled down into the environment undetected. As companies also look to protect their intellectual property, data exfiltration is a growing threat to any organization. Palo Alto Networks Data Loss Prevention(DLP) can be leveraged to protect from data exfiltration. In today's cyber security landscape, the nature in which websites can now be compromised has grown exponentially. You can't be guaranteed that websites are delivering safe content. Because of this high availability of protected outbound requests is becoming a pre-requisite for most organizations.


paloalto NETWORKS
in
Microsoft Azure
Inbound Request Topology
Internet
Outbound traffic requests bypass any Layer-7 traffic inspection
World Wide Web
Internet
Application Layer Load Balancer
VM-FW1
VM-Series
VM-Series
VM-FW2
Internal LB
VM-Dev1
VM-Web1
1. Packets ingress the Application load balancer, then forwarded to the VM series Firewall.
2. Inbound traffic requests are inspected by the VM series firewall.
3. Inbound traffic enters and exits the same path

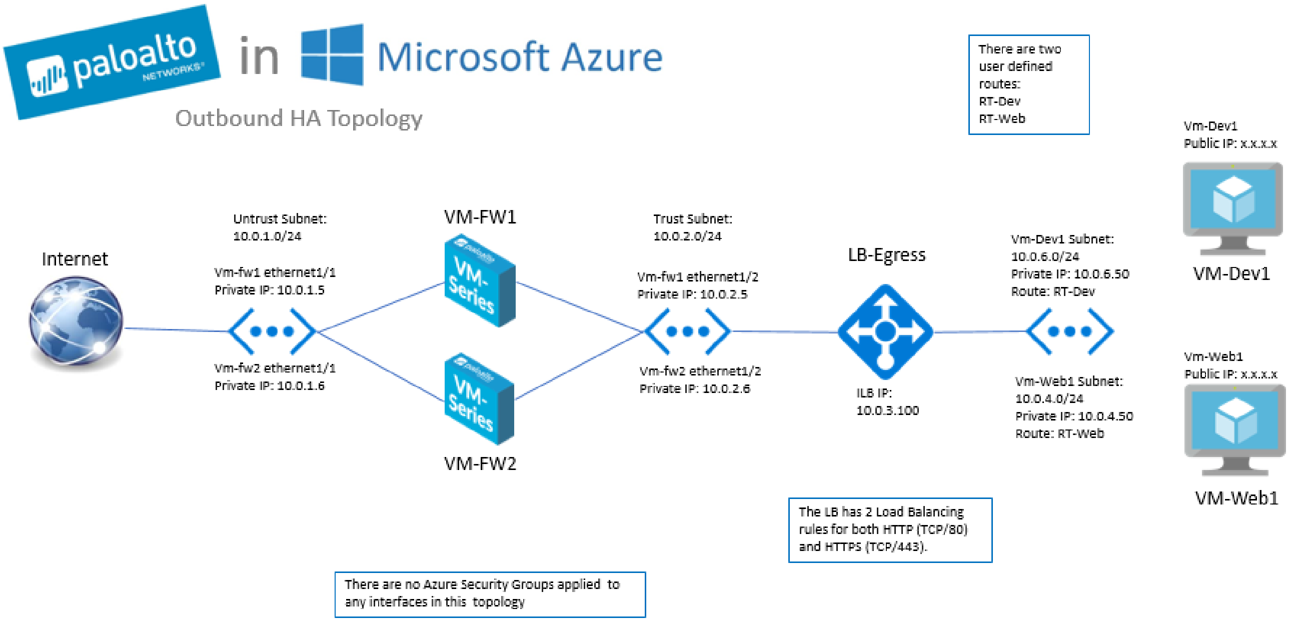
Topology Review – Lab Architecture Overview

In this topology review you will:

- Have a visual representation of the topology you will deploy
- See where every resource you configure lives in the architecture

In the next section, you will review the lab topology that addresses outbound traffic vulnerabilities while maintaining high availability.

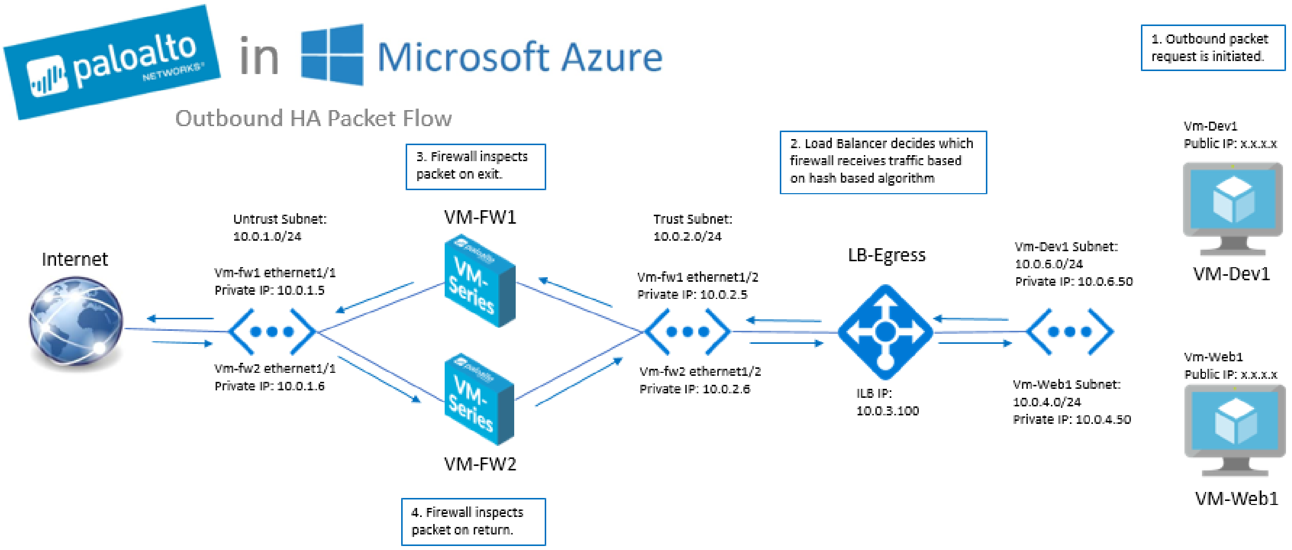
Topology Overview – Lab architecture overview



There are two user defined routes:
RT-Dev
RT-Web

The LB has 2 Load Balancing rules for both HTTP (TCP/80) and HTTPS (TCP/443).

There are no Azure Security Groups applied to any interfaces in this topology



Outbound HA Packet Flow

1. Outbound packet request is initiated.

2. Load Balancer decides which firewall receives traffic based on hash based algorithm

3. Firewall inspects packet on exit.

4. Firewall inspects packet on return.

Activity 1 – Azure Outbound HA Deployment

In this activity, you will:

- Deploy Azure Template from GitHub
- Confirm successful template deployment
- Verify creation of resources in portal.azure.com

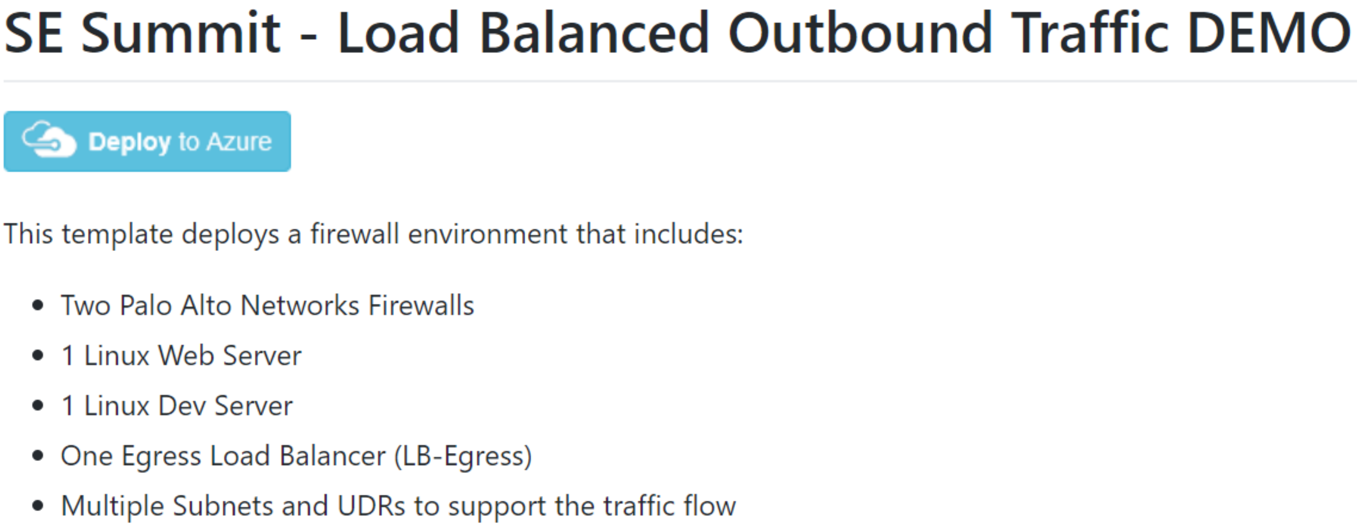
Introduction: Now that you are familiar with the topology, you will deploy the Outbound HA template directly from the GitHub repository. The template you will launch creates all the resources needed for a complete topology. You will need to use your student ID number we provide to you to properly fill in the Resource Group and vNet parameters required to deploy the template. Once the template is launched, follow the steps in the task to view the deployment progress as well as verify that all resources are created.

Task 1 – Deploy Azure Template from GitHub

Step 1: First, confirm that you have internet access and that you have a functioning web browser. Make sure you have access to a Microsoft Azure portal account. In this task, you will be deploying the Azure Outbound HA template directly from GitHub to the Azure Portal. Due to time sensitivity, Azure portal access is a prerequisite of this training so please have your azure portal account setup and ready to go.

Step 2: Open your web browser and navigate to the following GitHub URL https://github.com/jpeezus/SE-Summit

Once you've navigated to the GitHub URL, you should see a web page that looks like the screenshot below. Briefly review the infrastructure that will be deployed in the template, then click the "Deploy to Azure" icon.



SE Summit - Load Balanced Outbound Traffic DEMO

This template deploys a firewall environment that includes:

- Two Palo Alto Networks Firewalls
- 1 Linux Web Server
- 1 Linux Dev Server
- One Egress Load Balancer (LB-Egress)
- Multiple Subnets and UDRs to support the traffic flow

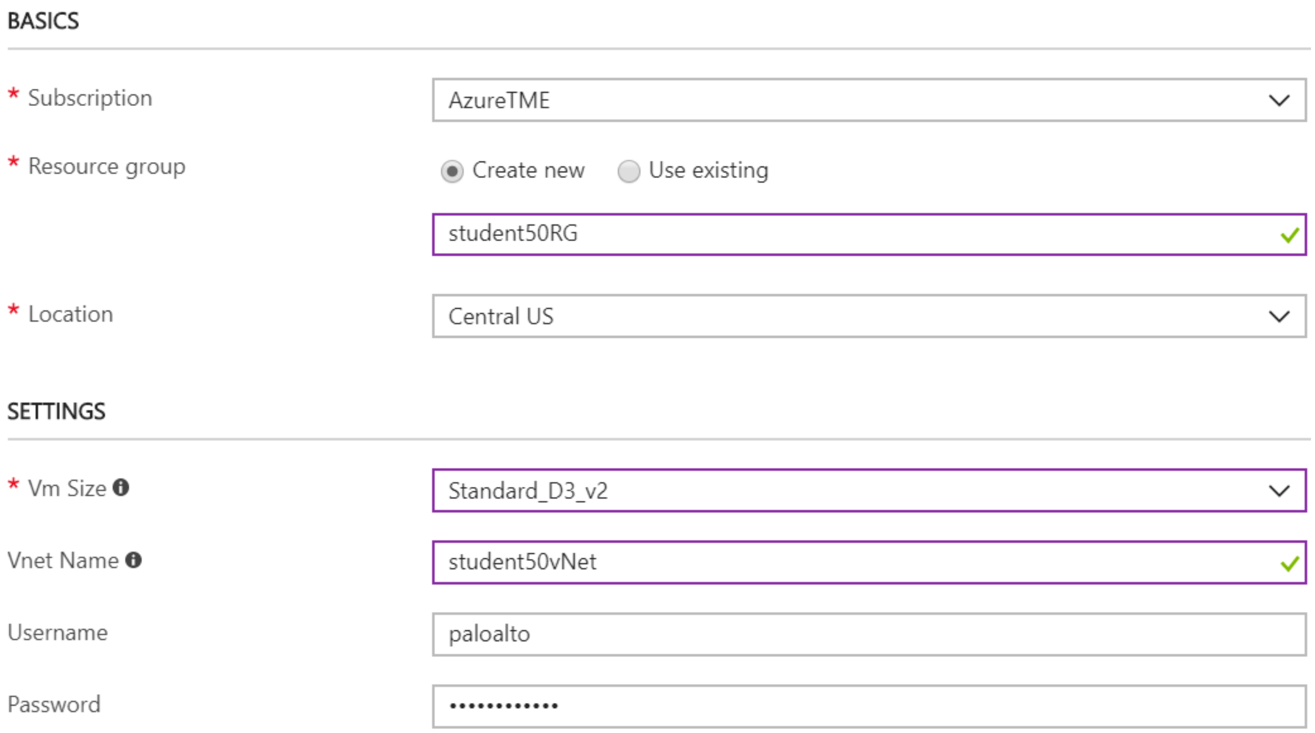
Step 3: Next, you will input your parameters.

- Select the azure subscription you will be using. If you only have one subscription, then leave it at the default.
- Each desk will be assigned a **student number**. Some of you will be using shared Azure subscriptions and this will help to differentiate your deployment from your colleagues.
- Create your Resource group. Use the format of student<**studentid**>RG to name your Resource Group. For example, if your student number is 50, it should look like **student50RG**. Select "**Create new"** and type in your resource group name. Save the resource group name in your text editor
- Select your region. The region you deploy to will vary based on your organization. For internal Palo Alto Networks Employees select **Central US**. For partners please select the region you have designated for this deployment. Below is a link which lists the supported regions.

Supported Azure Public Regions

https://www.paloaltonetworks.com/documentation/global/compatibility-matrix/vm-series-firewalls/azure-regions.html

- Select your VM size as "**Standard_D3_v2**"
- Next create a vNet name using the format student<studentid>vNet. **For example**, if the number on your desk is **50** it should look like this **student50vNet.** Be sure to use your own number!
- Save the **following** username and password in your text editor. This will be used to access the virtual machines deployed by the template. Username:**paloalto** Password:**Paloalto123!**



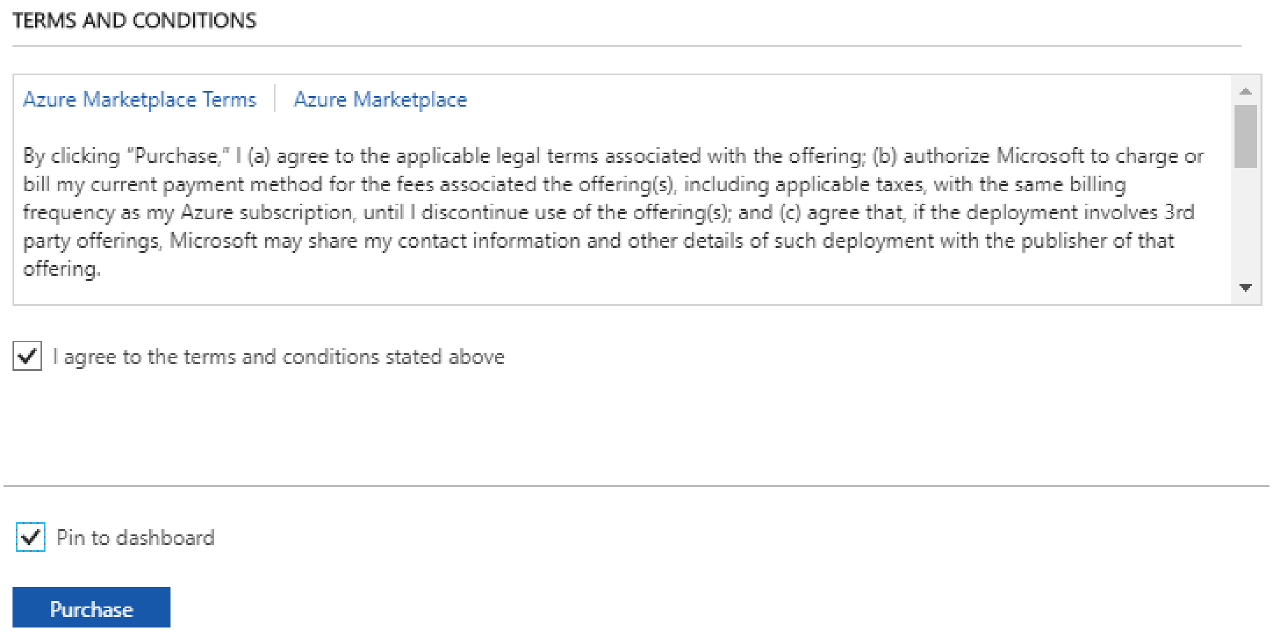
BASICS

* Subscription	AzureTME
* Resource group	● Create new ○ Use existing student50RG ✓
* Location	Central US

SETTINGS

* Vm Size	Standard_D3_v2
Vnet Name	student50vNet ✓
Username	paloalto
Password	••••••••••••

Step 4: Be sure to agree to the **terms and conditions**. Check the box next to **Pin to dashboard**, then click **Purchase** to launch the template. The template will take up to 4 minutes to complete. Again, a free trial Azure account will not be sufficient for this lab.



TERMS AND CONDITIONS

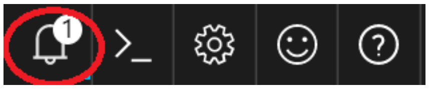
Azure Marketplace Terms | Azure Marketplace

By clicking "Purchase," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

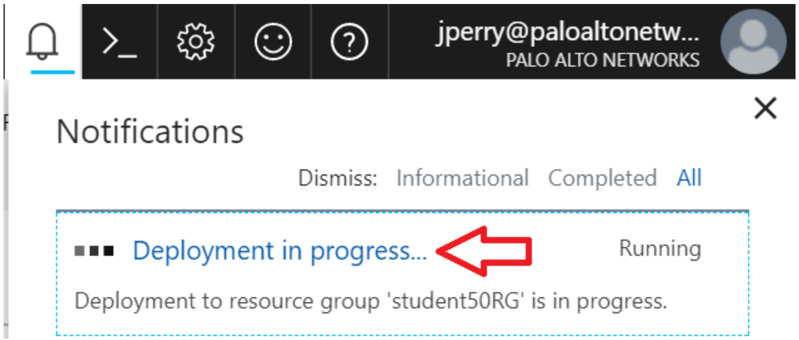
☑ I agree to the terms and conditions stated above

☑ Pin to dashboard

Step 5: To check the progress of the deployment, navigate to the top right of the screen and click the bell icon. This will open the notifications tab.



Step 6: In the notifications tab click **Deployment in progress**. While the template is being launched, you can check the status from this view.

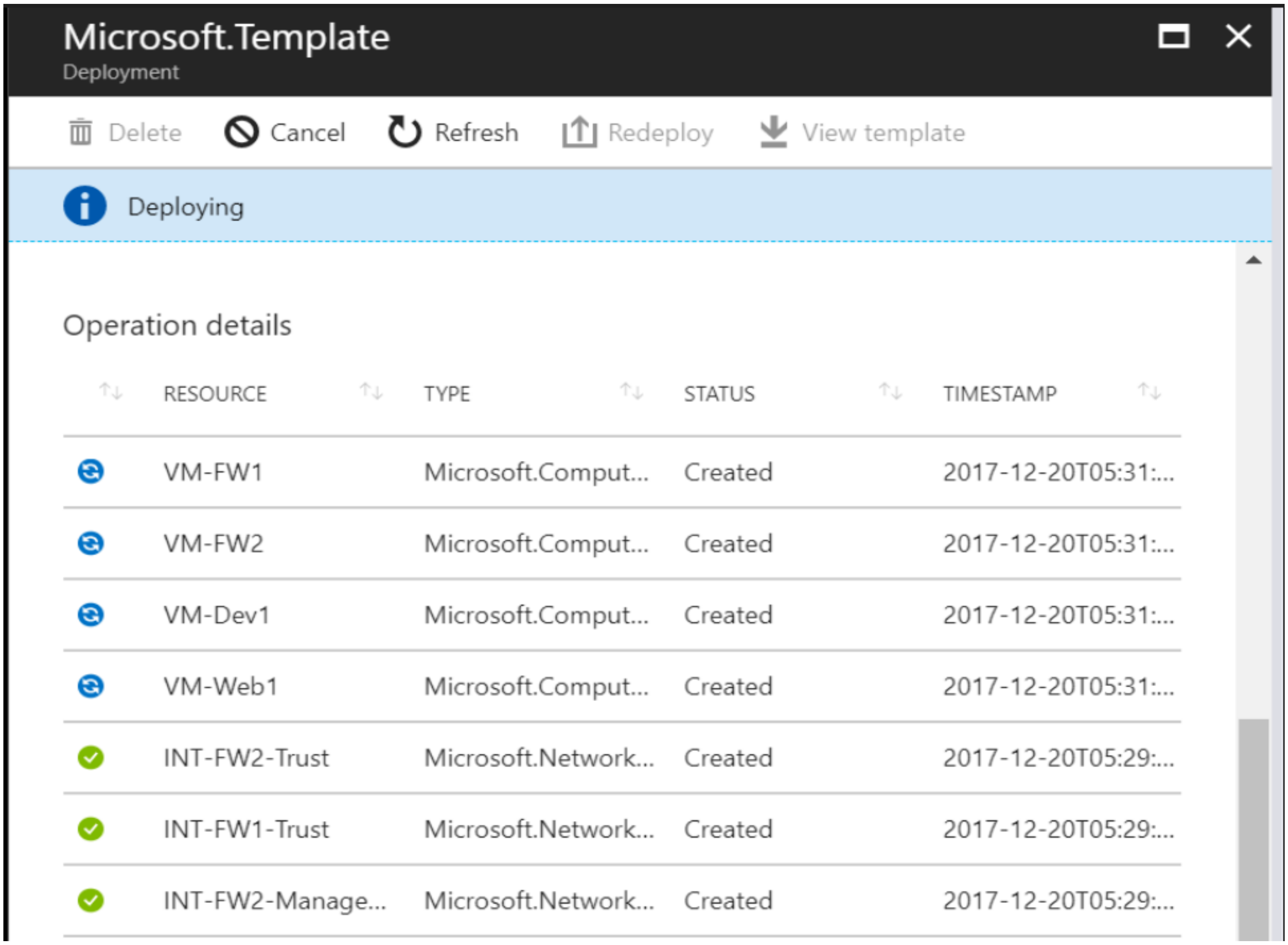


Notifications

Dismiss: Informational Completed All

Deployment in progress... Running

Deployment to resource group 'student50RG' is in progress.

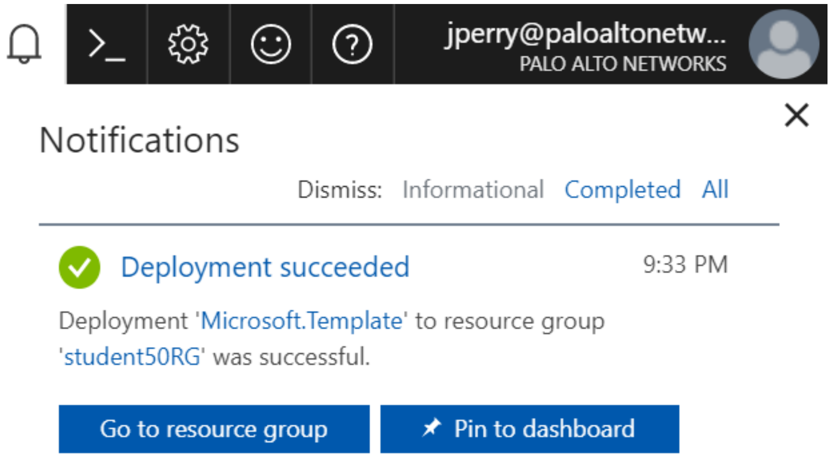


Deploying

Operation details

	RESOURCE	TYPE	STATUS	TIMESTAMP
⟳	VM-FW1	Microsoft.Comput...	Created	2017-12-20T05:31:...
⟳	VM-FW2	Microsoft.Comput...	Created	2017-12-20T05:31:...
⟳	VM-Dev1	Microsoft.Comput...	Created	2017-12-20T05:31:...
⟳	VM-Web1	Microsoft.Comput...	Created	2017-12-20T05:31:...
✓	INT-FW2-Trust	Microsoft.Network...	Created	2017-12-20T05:29:...
✓	INT-FW1-Trust	Microsoft.Network...	Created	2017-12-20T05:29:...
✓	INT-FW2-Manage...	Microsoft.Network...	Created	2017-12-20T05:29:...

Step 7: When the deployment is complete, the notifications tab will say "Deployment succeeded" with a green check mark.

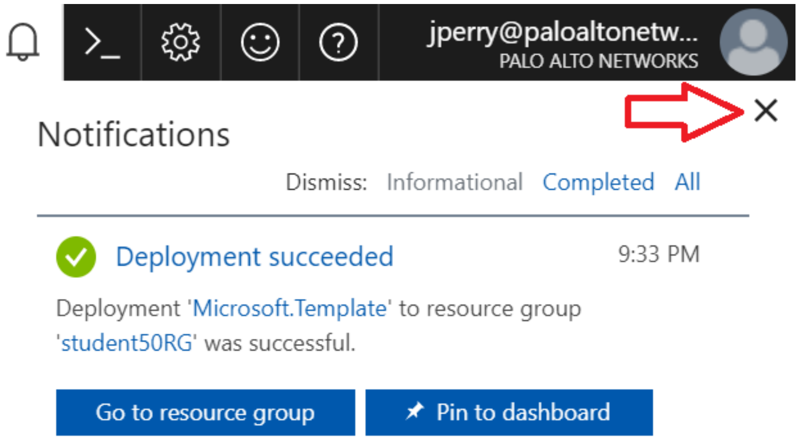


Notifications

✓ Deployment succeeded 9:33 PM

Deployment 'Microsoft.Template' to resource group 'student50RG' was successful.

Step 8: Click the **X** on the top right to close this tab and any tab moving forward.



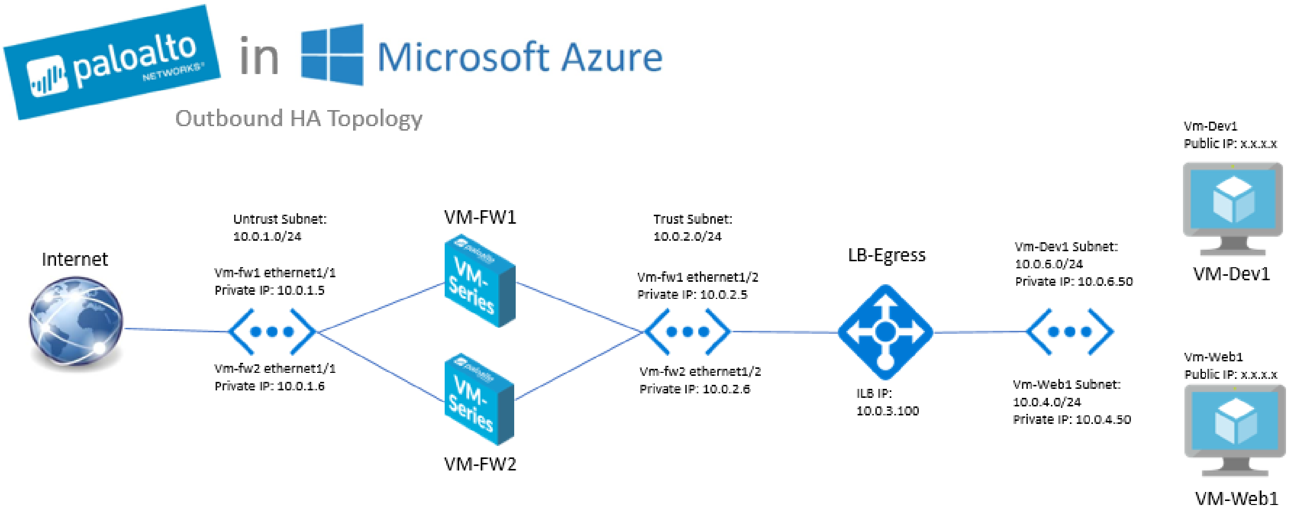
Notifications

Dismiss: Informational Completed All

Deployment succeeded 9:33 PM

Deployment 'Microsoft.Template' to resource group 'student50RG' was successful.

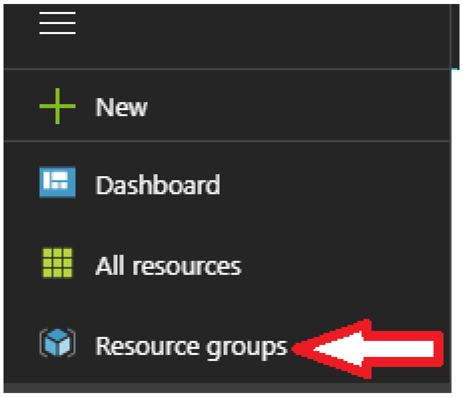
Step 9: You have just deployed the following topology in your Azure portal account.



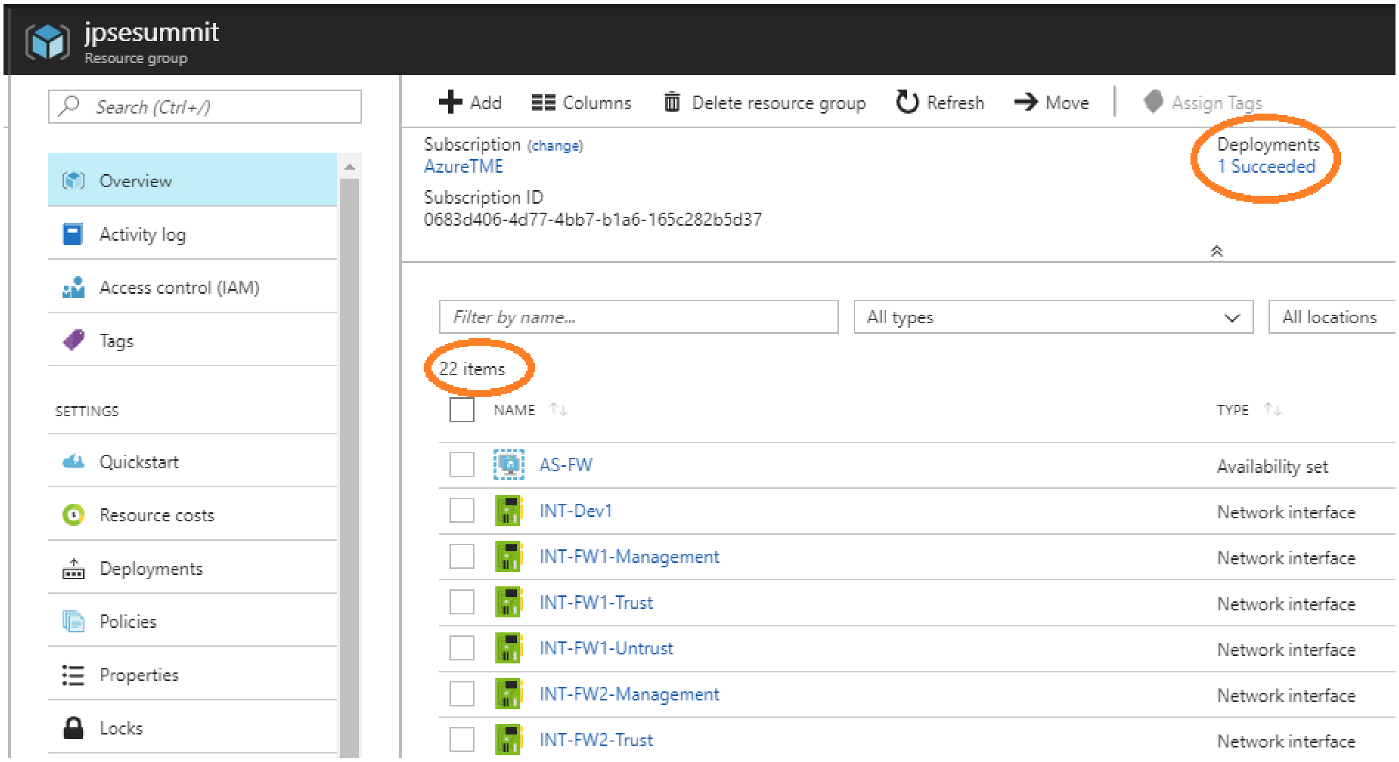
Task 2 – Verify Resources in your Azure Deployment

Now that your Azure template has been deployed, there are a few things you need to verify.

Step 1: Verify that you have all the resources needed in your deployment. In the left side pane click **Resource Groups.** Select your resource group named **studentXXRG**. You can filter by name.



Step 2: In the resource group options select "Overview," you should see **22 items**. For deployments, you should see **1 Succeeded**.



Subscription (change)
AzureTME

Subscription ID
0683d406-4d77-4bb7-b1a6-165c282b5d37

Deployments
1 Succeeded

22 items

NAME	TYPE
AS-FW	Availability set
INT-Dev1	Network interface
INT-FW1-Management	Network interface
INT-FW1-Trust	Network interface
INT-FW1-Untrust	Network interface
INT-FW2-Management	Network interface
INT-FW2-Trust	Network interface

End of Activity: In Activity 1, you successfully launched an ARM template into the Azure Portal. To instantiate the launch, you had to input parameters such as resource group name, vNet name, username, password, and instance size. You also verified the completion of the template deployment and you also verified that all resources were created in the resource group. Now that you have finished this activity, please move forward to the next activity.

Activity 2 – Customize your Azure Deployment

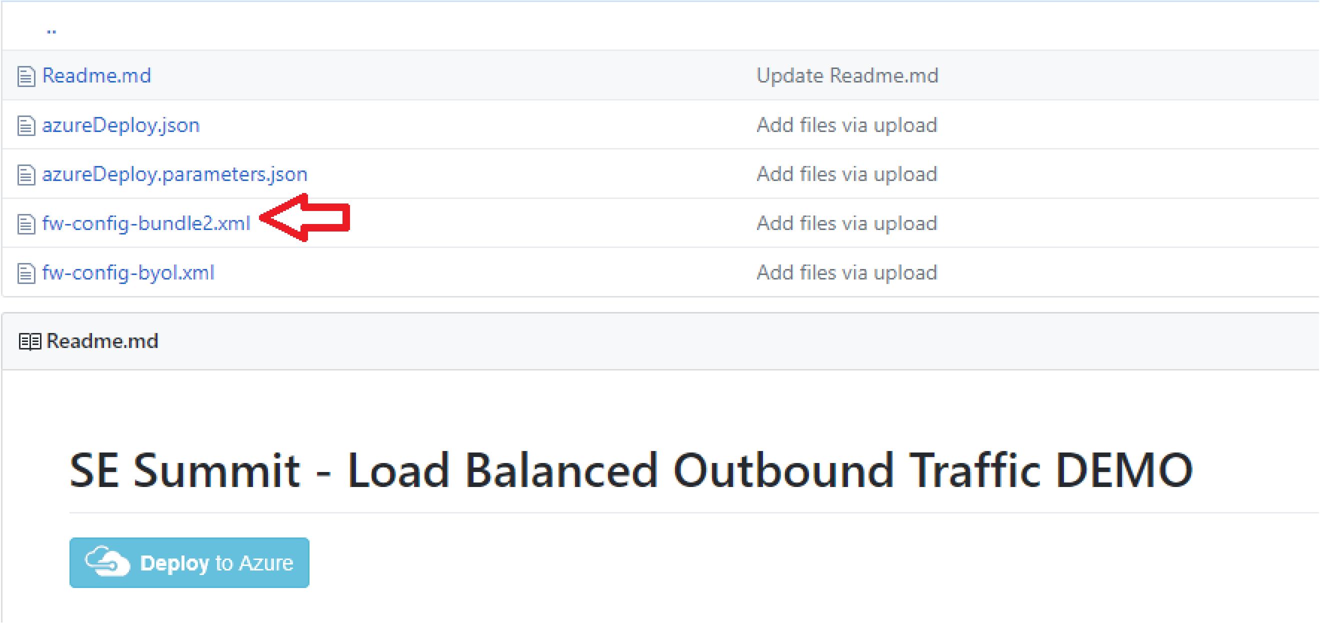
In this activity you will:

- Download the firewall configuration from GitHub and import to firewall
- Add route to an existing UDR in Azure for management access to VM-Dev1 and VM-Web1
- Create and add security group for firewall access

Introduction: Now you have successfully deployed your template, you will need to take a few steps to customize your deployment. The steps you perform will apply the firewall configuration, and create routes within the Azure network. The routes you create in Azure will allow you SSH access to both VM-Dev1 and VM-Web1. Username:**paloalto** Password:**Paloalto123!**

Task 1 – Download and Import Firewall Configuration

Step 1: Navigate to GitHub URL https://github.com/jpeezus/SE-Summit and click the **fw-config-bundle2.xml** file.

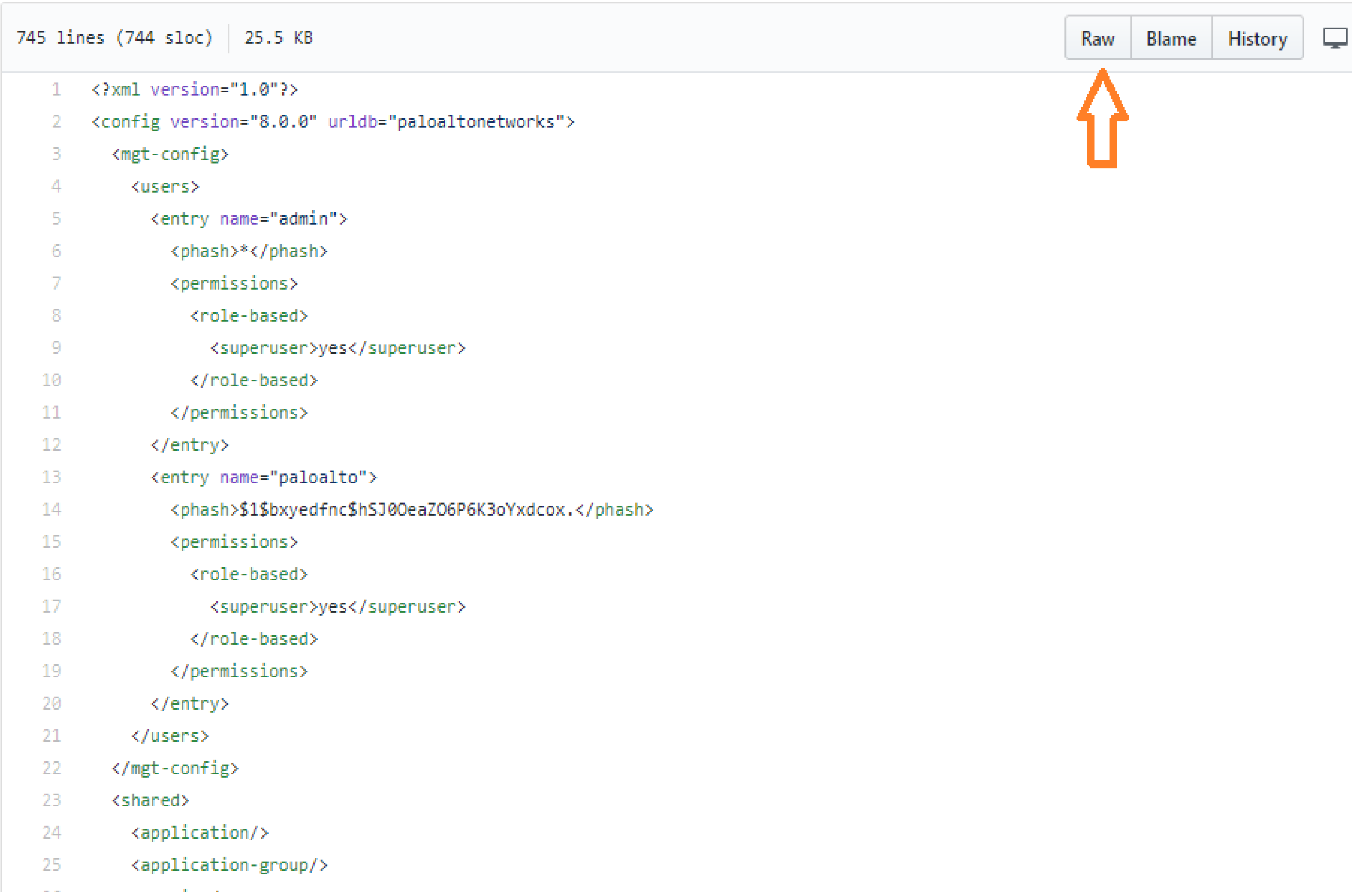


..	
Readme.md	Update Readme.md
azureDeploy.json	Add files via upload
azureDeploy.parameters.json	Add files via upload
fw-config-bundle2.xml	Add files via upload
fw-config-byol.xml	Add files via upload

Readme.md

SE Summit - Load Balanced Outbound Traffic DEMO

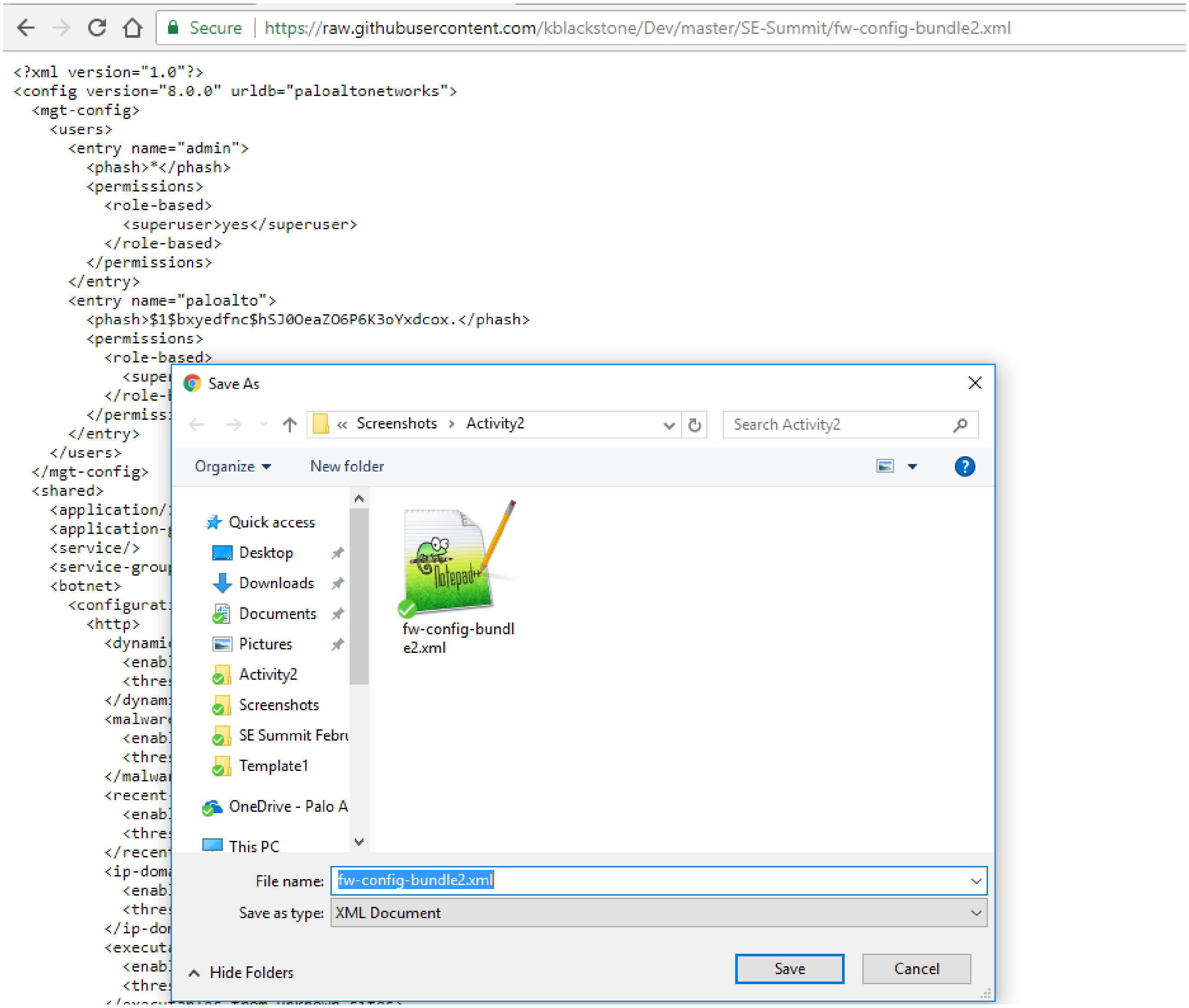
Step 2: Once you see the xml data, look to the top right and select "Raw". The xml data will now be directly in the web browser instead of GitHub.



745 lines (744 sloc) | 25.5 KB

```
<?xml version="1.0"?>
<config version="8.0.0" urldb="paloaltonetworks">
  <mgt-config>
    <users>
      <entry name="admin">
        <phash>*</phash>
        <permissions>
          <role-based>
            <superuser>yes</superuser>
          </role-based>
        </permissions>
      </entry>
      <entry name="paloalto">
        <phash>$1$bxyedfnc$hSJ0OeaZO6P6K3oYxdcox.</phash>
        <permissions>
          <role-based>
            <superuser>yes</superuser>
          </role-based>
        </permissions>
      </entry>
    </users>
  </mgt-config>
  <shared>
    <application/>
    <application-group/>
```

Step 3: From there "right click" and select "Save as" to download the fw-config-bundle2.xml to your laptop.



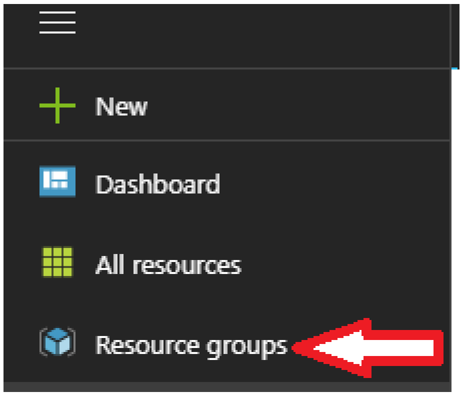
Secure | https://raw.githubusercontent.com/kblackstone/Dev/master/SE-Summit/fw-config-bundle2.xml

```
<?xml version="1.0"?>
<config version="8.0.0" urldb="paloaltonetworks">
  <mgt-config>
    <users>
      <entry name="admin">
        <phash>*</phash>
        <permissions>
          <role-based>
            <superuser>yes</superuser>
          </role-based>
        </permissions>
      </entry>
      <entry name="paloalto">
        <phash>$1$bxyedfnc$hSJ0OeaZ06P6K3oYxdcox.</phash>
        <permissions>
          <role-based>
            <supe
          </role-
        </permiss
      </entry>
    </users>
  </mgt-config>
  <shared>
    <application/
    <application-
    <service/>
    <service-grou
    <botnet>
      <configurat
        <http>
          <dynami
            <enab
            <thre
          </dynam
          <malwar
            <enab
            <thre
          </malwa
          <recent
            <enab
            <thre
          </recen
          <ip-dom
            <enab
            <thre
          </ip-do
          <execut
            <enab
            <thre
          </executables-from-unknown-sites>
```

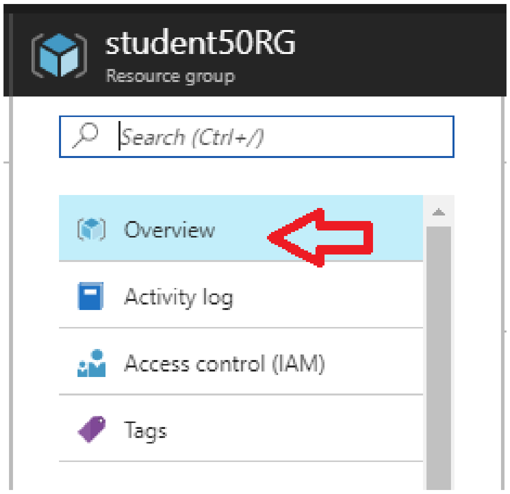
File name: fw-config-bundle2.xml

Save as type: XML Document

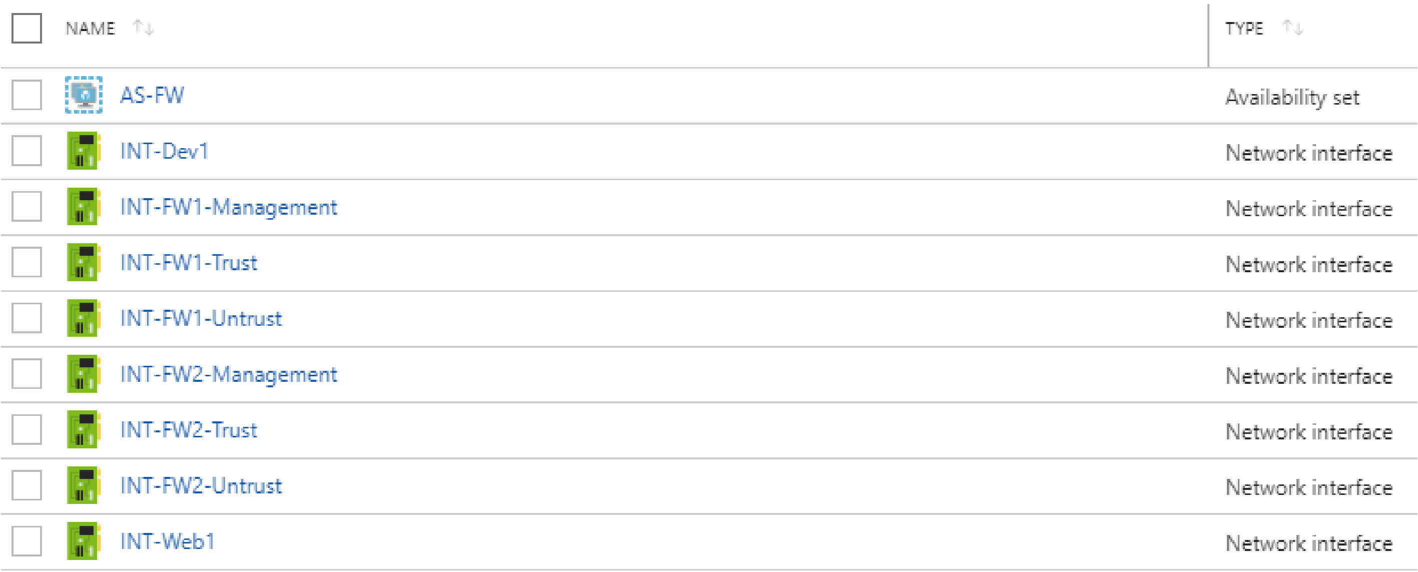
Step 4: Open your web browser and type in portal.azure.com to bring up the azure portal. Within the Azure Portal, in the left side pane select **Resource Groups**.



Step 5: Select the resource group that you created earlier, then click **Overview**. You can filter by resource group name on the top.

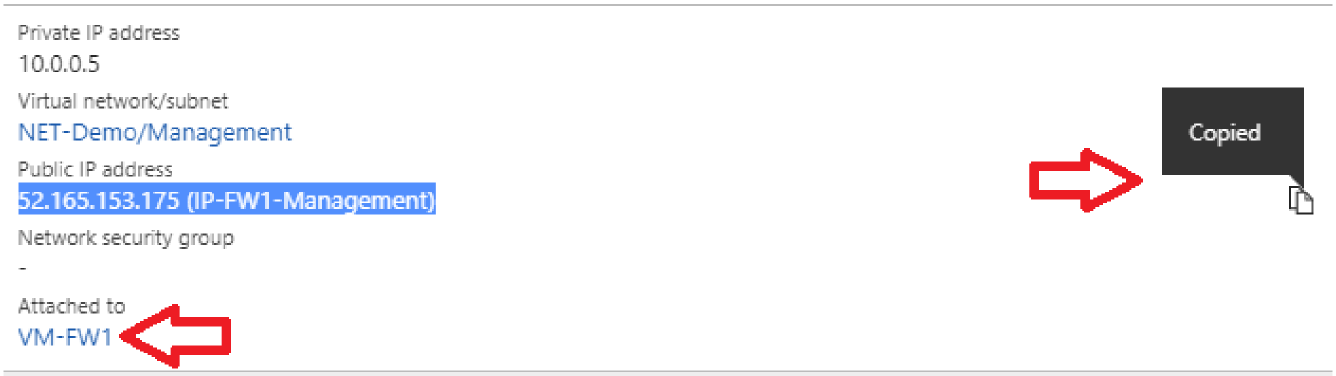


Step 6: On the list of resources select **INT-FW1-Management**.



□ NAME	TYPE
□ AS-FW	Availability set
□ INT-Dev1	Network interface
□ INT-FW1-Management	Network interface
□ INT-FW1-Trust	Network interface
□ INT-FW1-Untrust	Network interface
□ INT-FW2-Management	Network interface
□ INT-FW2-Trust	Network interface
□ INT-FW2-Untrust	Network interface
□ INT-Web1	Network interface

Step 7: On the **INT-FW1-Management** screen locate the public IP address and save that in your text editor. Locate the "**Attached to**" wording and add the VM name to your text editor as well.



Private IP address
10.0.0.5

Virtual network/subnet
NET-Demo/Management

Public IP address
52.165.153.175 (IP-FW1-Management)

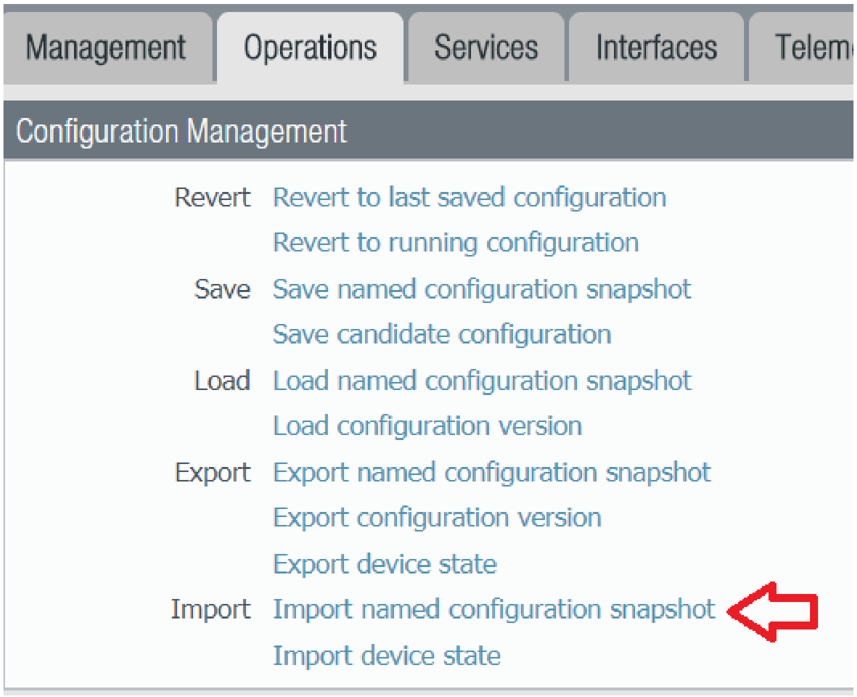
Network security group
-

Attached to
VM-FW1

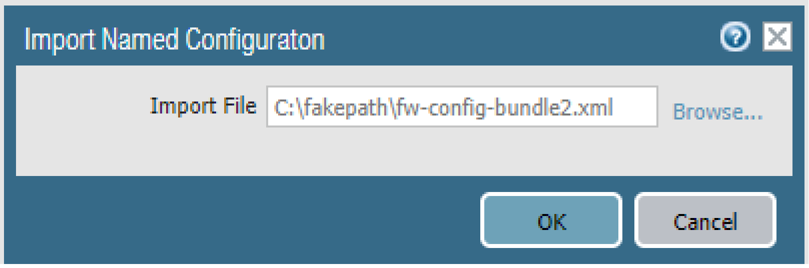
Step 8: Repeat Steps 4-7 of this task for **INT-FW2-Management**, **INT-Dev1**, and **INT-Web1**. You should end up with four public IP addresses and four virtual machine names in your text editor.

```
52.165.153.175 (IP-FW1-Management)
52.165.162.188 (IP-FW2-Management)
52.165.160.73 (IP-Dev1)
52.165.159.44 (IP-Web1)
```

Step 9: With the username and password you saved earlier, log into **VM-FW1** using https://<publicIP>. Navigate to the **Devices** tab, **Setup, Operations,** then **Import named configuration snapshot**. Username:**paloalto** Password:**Paloalto123!**


Management
Operations
Services
Interfaces
Telem
Configuration Management
Revert
Revert to last saved configuration
Revert to running configuration
Save
Save named configuration snapshot
Save candidate configuration
Load
Load named configuration snapshot
Load configuration version
Export
Export named configuration snapshot
Export configuration version
Export device state
Import
Import named configuration snapshot
Import device state

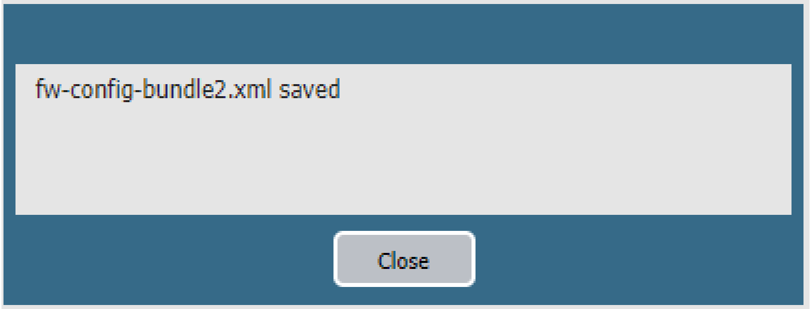
Step 10: Browse to the location you saved the **fw-config-bundle2.xml** and click OK.



Import Named Configuraton

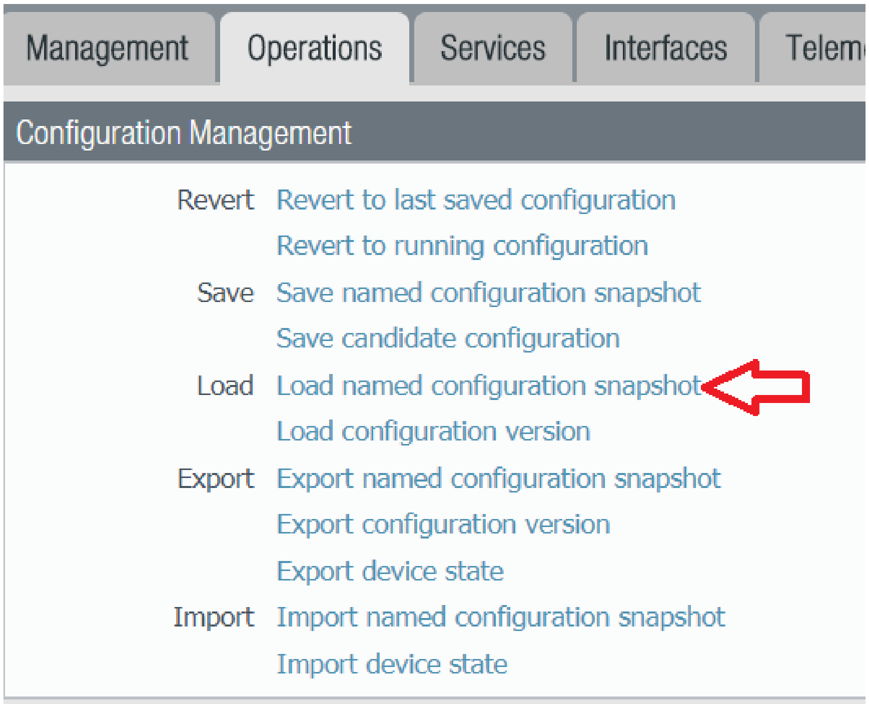
Import File C:\fakepath\fw-config-bundle2.xml

Step 11: Once imported, the output will display **fw-config-bundle2.xml saved**. Click **Close.**



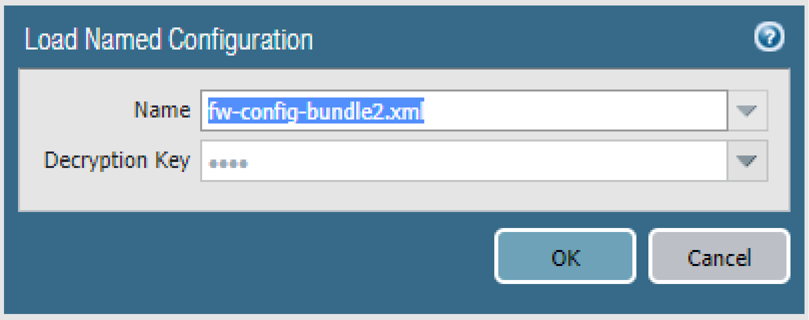
fw-config-bundle2.xml saved

Step 12: Next, from the same location select **Load named configuration snapshot.**



Management
Operations
Services
Interfaces
Telem
Configuration Management
Revert
Revert to last saved configuration
Revert to running configuration
Save
Save named configuration snapshot
Save candidate configuration
Load
Load named configuration snapshot
Load configuration version
Export
Export named configuration snapshot
Export configuration version
Export device state
Import
Import named configuration snapshot
Import device state

Step 13: From the drop down, list select **fw-config-bundle2.xml**, then click **OK**.

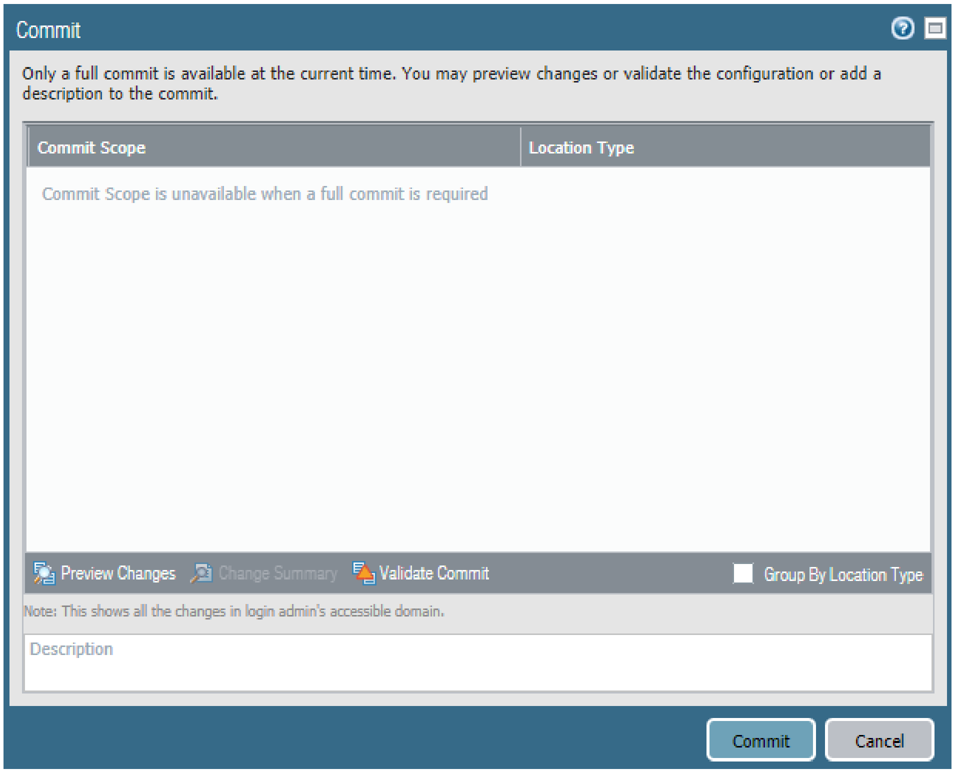


Load Named Configuration

Name: fw-config-bundle2.xml

Decryption Key: ••••

Step 14: From the top right click **Commit** to save the changes. Disregard any commit warnings. When the commit is complete, click **Close**.



Commit

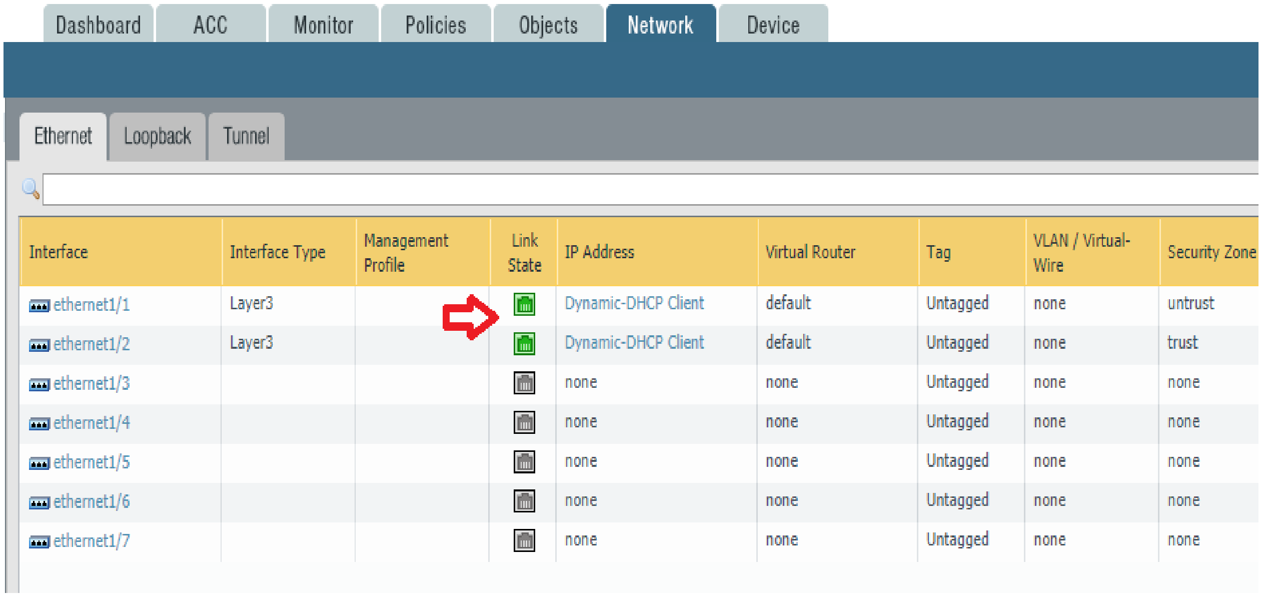
Only a full commit is available at the current time. You may preview changes or validate the configuration or add a description to the commit.

Commit Scope	Location Type
Commit Scope is unavailable when a full commit is required	

Note: This shows all the changes in login admin's accessible domain.

Description

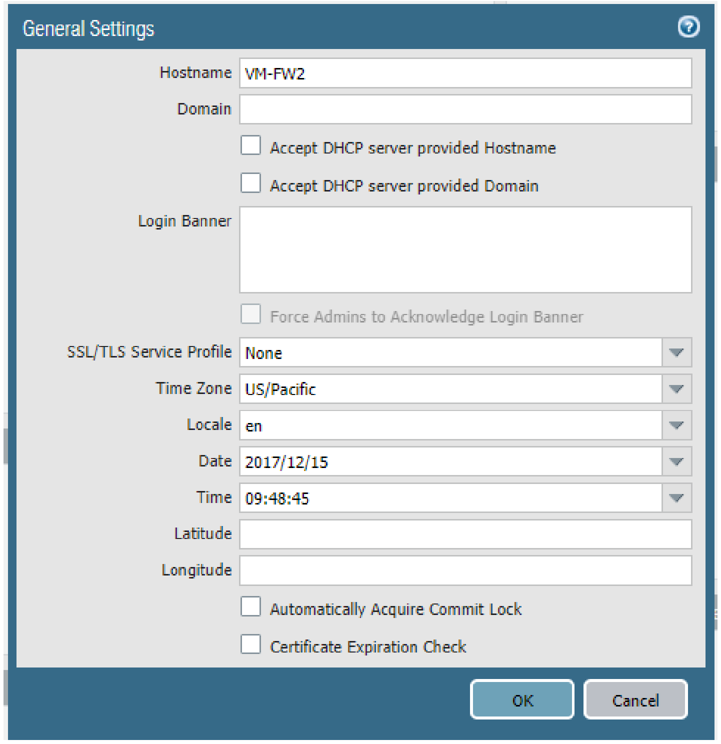
Step 15: Select the **Network** tab, then click **Ethernet.** Verify **Ethernet1/1** and **Ethernet1/2** are green.



Interface	Interface Type	Management Profile	Link State	IP Address	Virtual Router	Tag	VLAN / Virtual-Wire	Security Zone
ethernet1/1	Layer3		🟩	Dynamic-DHCP Client	default	Untagged	none	untrust
ethernet1/2	Layer3		🟩	Dynamic-DHCP Client	default	Untagged	none	trust
ethernet1/3			▨	none	none	Untagged	none	none
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

Step 16: Repeat Steps 9-16 of this task on **VM-FW2**.

Step 17: VM-FW2 will be named VM-FW1 because the same configuration file was used to configure both firewalls. To avoid confusion, change the hostname to VM-FW2 by navigating to **Devices**, in the left side pane select **Setup**, then click **Management**. Change the hostname then commit the changes. Disregard any commit warnings.



General Settings

Hostname	VM-FW2
Domain	
	□ Accept DHCP server provided Hostname
	□ Accept DHCP server provided Domain
Login Banner	
	□ Force Admins to Acknowledge Login Banner
SSL/TLS Service Profile	None
Time Zone	US/Pacific
Locale	en
Date	2017/12/15
Time	09:48:45
Latitude	
Longitude	
	□ Automatically Acquire Commit Lock
	□ Certificate Expiration Check

End of Task: In this task you downloaded the firewall snapshot configuration file from GitHub, you imported the configuration into the firewall and you committed the configuration. Once the configuration was applied you confirmed the interface was configured. You also changed the hostname of VM-FW2 so that it's hostname and Azure portal name are consistent. Now that you have completed this section please move on to the next task.

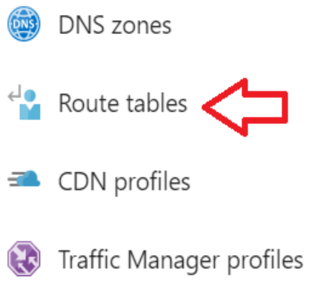
Task 2 – Add Route to UDR Route Table

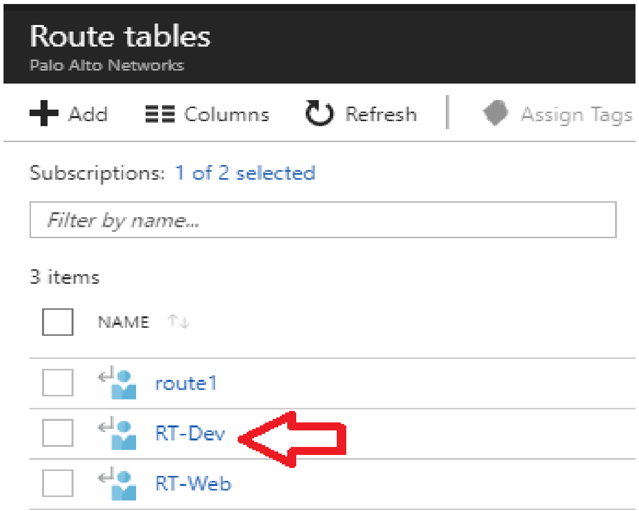
In this lab you will configure a user defined route to allow SSH access to VM-Dev1 and VM-Web1. User define routes are also used to forward traffic to the firewall for outbound inspection. Without user define routes traffic would never be sent to the firewalls, which would cause a lack of visibility for outbound traffic.

Step 1: From your laptop, go to https://ifconfig.co/. Add your IP address in your text editor. This is not the Public IP of any resource in Azure, but your laptop public IP address.

Step 2: Go back to portal.azure.com, In the left side pane select **Route Tables**. If you don't see the "Route Tables" option scroll down and click **More Services**

Step 3: From there click **Route Tables,** then double click **RT-DEV**



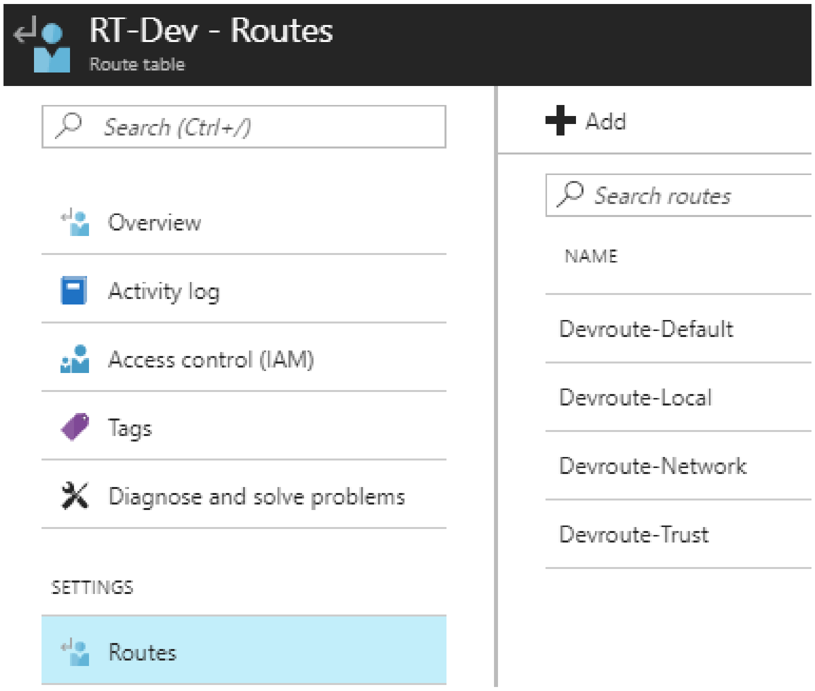

Route tables
Palo Alto Networks

Subscriptions: 1 of 2 selected

3 items

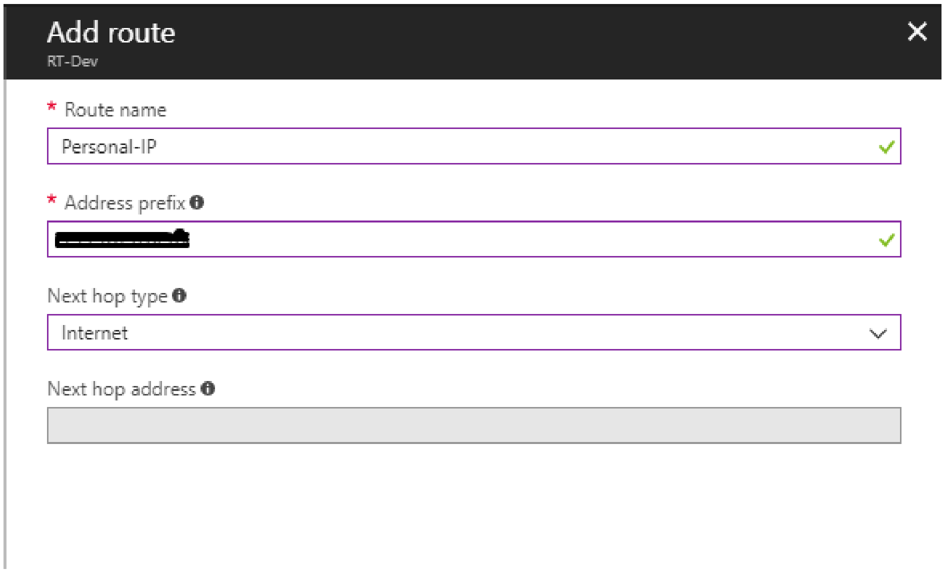
NAME
route1
RT-Dev
RT-Web

Step 4: On the expansion tab under Settings, select **Routes**. Click **Add**.


RT-Dev - Routes
Route table

NAME
Devroute-Default
Devroute-Local
Devroute-Network
Devroute-Trust

Step 5: Name the route **Personal-IP**. Add your **Public IP address** from your laptop with a **/32** mask. **Do Not** add the public IP address from any Virtual Machines. **Next hop type** will be **Internet**. Click **Ok**. Azure can often take minutes to apply changes to their backend network so be patient.


Add route
RT-Dev

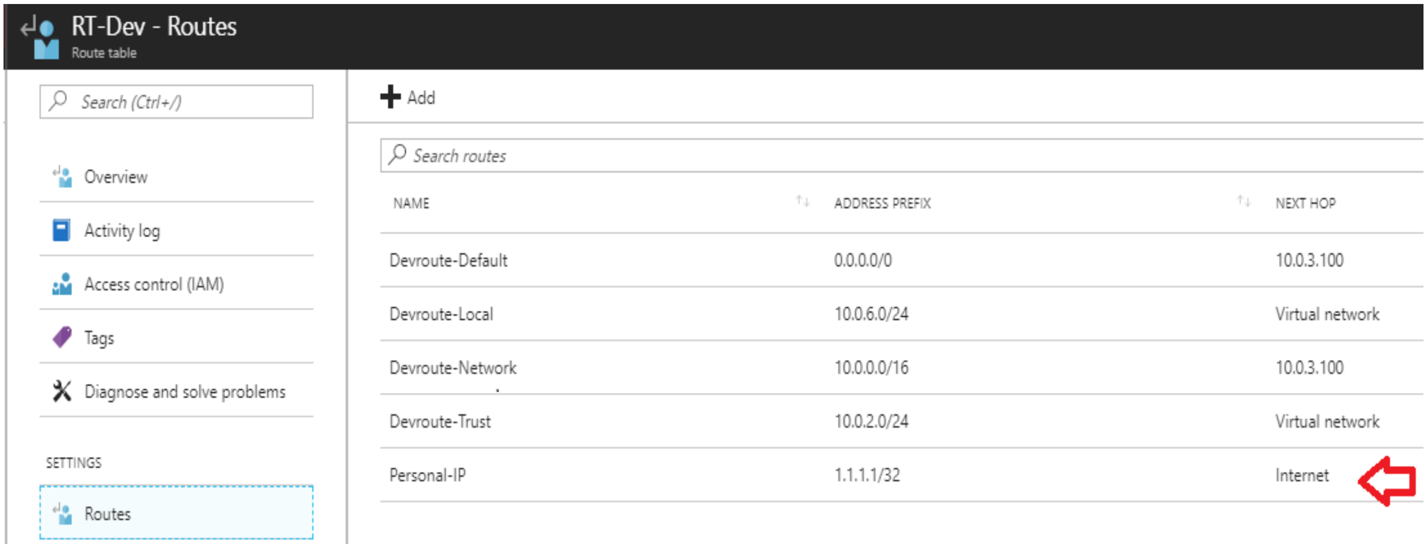
* Route name
Personal-IP

* Address prefix
[redacted]

Next hop type
Internet

Next hop address

Step 6: Look at the route table. The **user defined route** has different policies that route different subnets and IP addresses to different next hops. The route you added tells Azure to route all traffic for your **Personal-IP** address straight to the internet and bypass the load balancer. Without this route, you can't access VM-Dev1 and VM-Web1. If you don't see your IP address, then **refresh your browser**.


RT-Dev - Routes
Route table

NAME	ADDRESS PREFIX	NEXT HOP
Devroute-Default	0.0.0.0/0	10.0.3.100
Devroute-Local	10.0.6.0/24	Virtual network
Devroute-Network	10.0.0.0/16	10.0.3.100
Devroute-Trust	10.0.2.0/24	Virtual network
Personal-IP	1.1.1.1/32	Internet

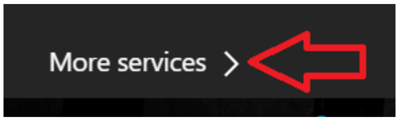
Step 7: Repeat **Steps 2-6** on Route Table **RT-Web**.

Task 3 – Create Security Group

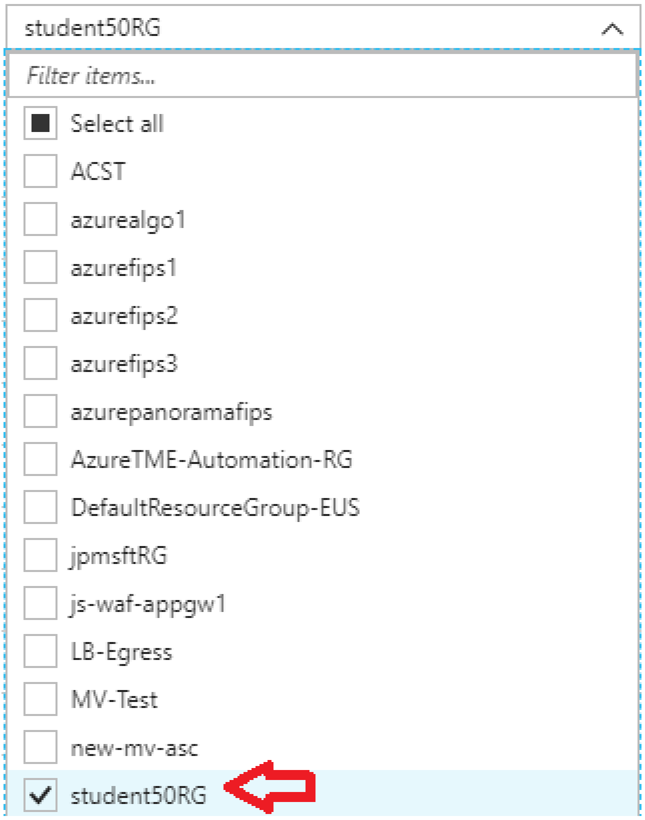
To emphasize security to the management interface of the firewall, you will create a security group. You will create an inbound rule that allows your personal IP address access to the azure management subnet explicitly where the firewall management interfaces reside.

Step 1: You gathered this information before but if you have forgotten, from your laptop, go to https://ifconfig.co/. This is not the Public IP of any resource in Azure, but your laptop public IP address.

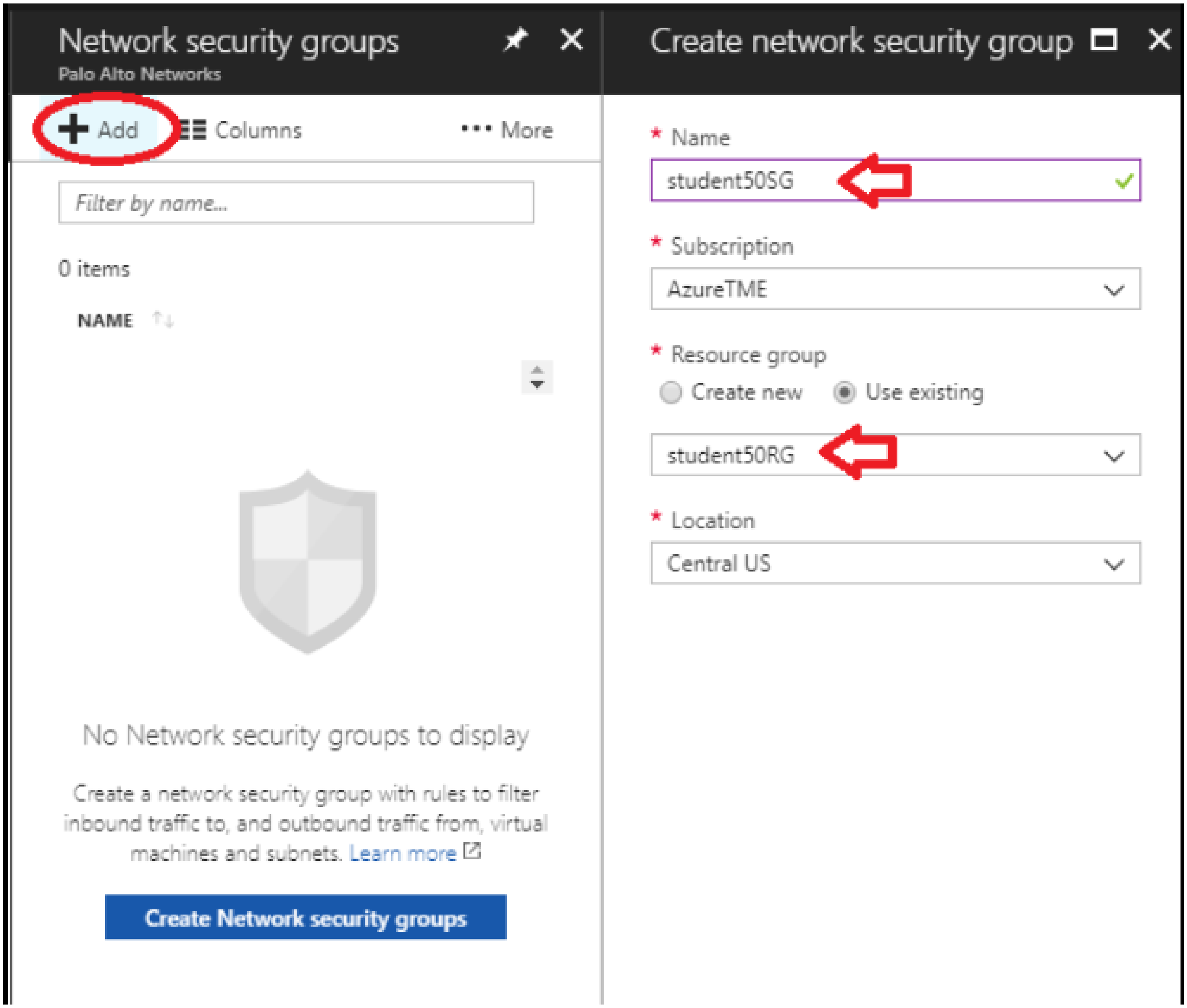
Step 2: Within Azure in the left side pane click **Network Security Groups**. If you don't see Network Security Groups click "**More Services**" at the bottom of the menu.



Step 3: In the top area under "**All resource groups**" click "**select all"** to remove all selections. Then select your resource group. You should **NOT** have a security group. If you see a security group, make sure you are in the correct resource group.



Step 4: Click **Add** to create a security group. Then use the same naming convention that you have used through this lab. Your security group should be named **studentxxSG.** Select "**Use existing"** then select your resource group. Click **Create.**


Network security groups
Palo Alto Networks
Add
Columns
More
Filter by name...
0 items
NAME
No Network security groups to display
Create a network security group with rules to filter inbound traffic to, and outbound traffic from, virtual machines and subnets. Learn more
Create Network security groups
Create network security group

* Name: student50SG

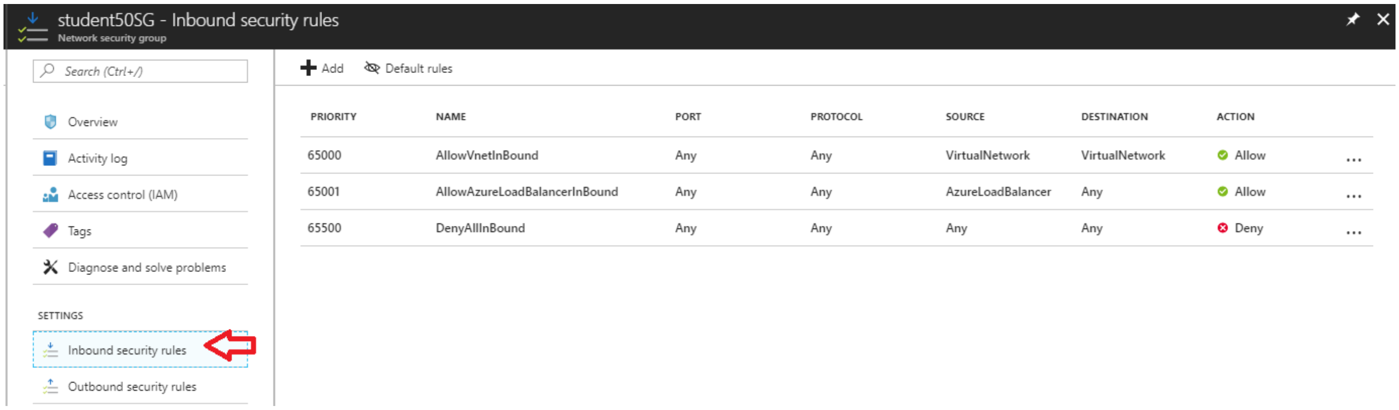
* Subscription: AzureTME

* Resource group: ○ Create new ● Use existing

student50RG

* Location: Central US

Step 5: Once the security group has completed creation navigate to the "**Inbound security rules**" tab in the security group. You may have to refresh your browser to see the newly created security group.

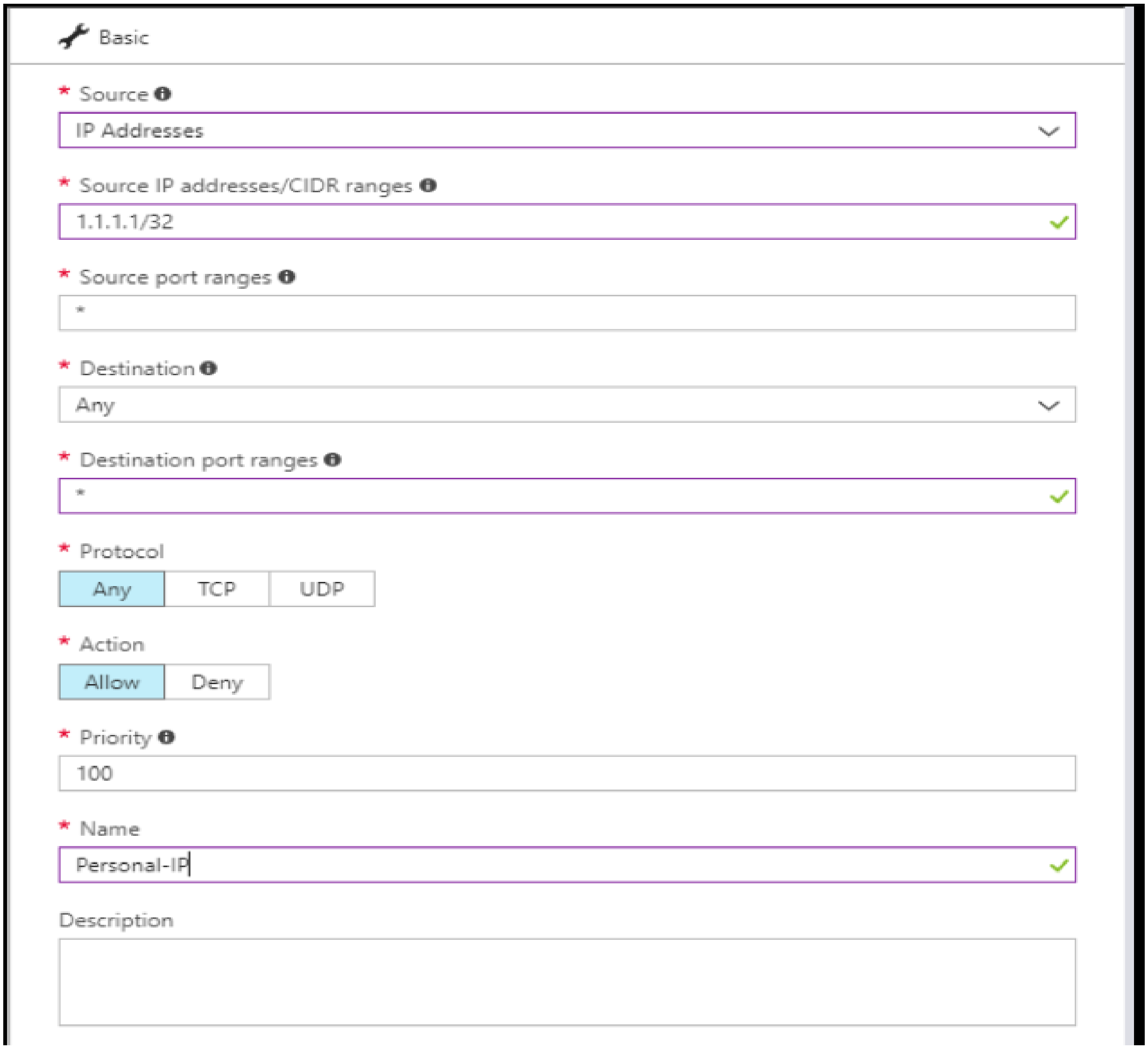


student50SG - Inbound security rules

Network security group

PRIORITY	NAME	PORT	PROTOCOL	SOURCE	DESTINATION	ACTION
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	✓ Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	✓ Allow
65500	DenyAllInBound	Any	Any	Any	Any	✗ Deny

Step 6: Click "**Add**", then duplicate the inbound security rule parameters you see on the screenshot below. Replace 1.1.1.1/32 with your Personal public IP address. Click **"OK"**



Basic

* Source: IP Addresses

* Source IP addresses/CIDR ranges: 1.1.1.1/32

* Source port ranges: *

* Destination: Any

* Destination port ranges: *

* Protocol: [✓ Any] TCP UDP

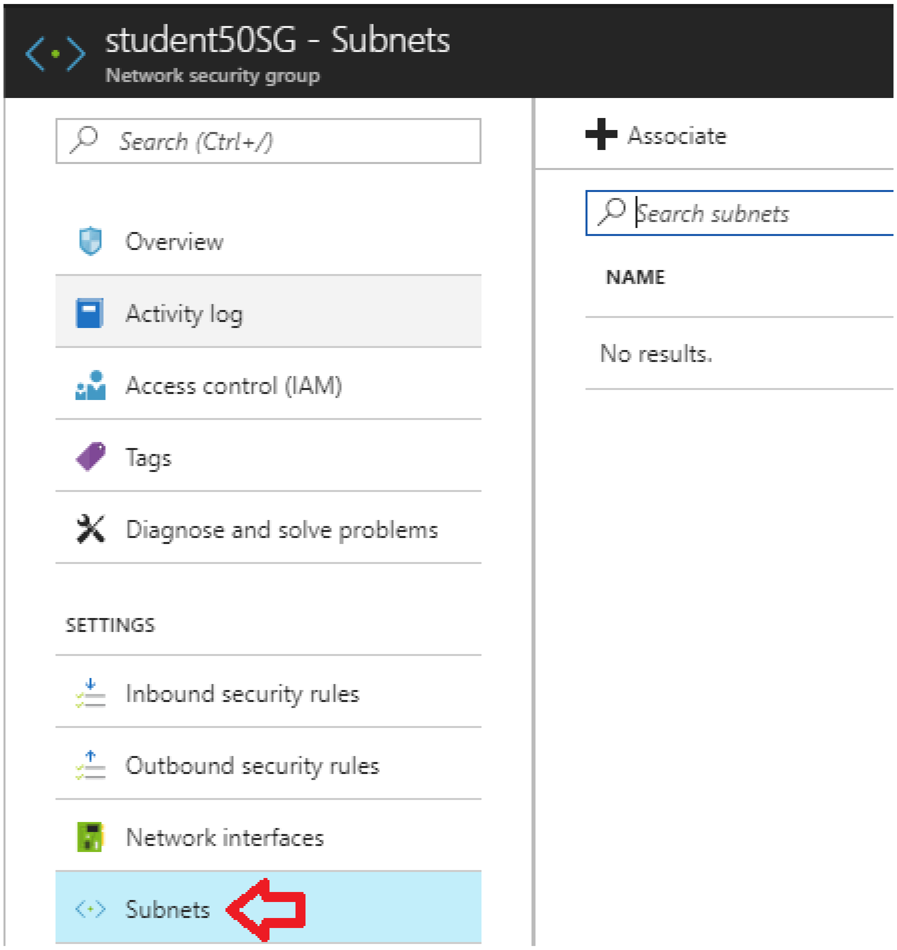
* Action: [✓ Allow] Deny

* Priority: 100

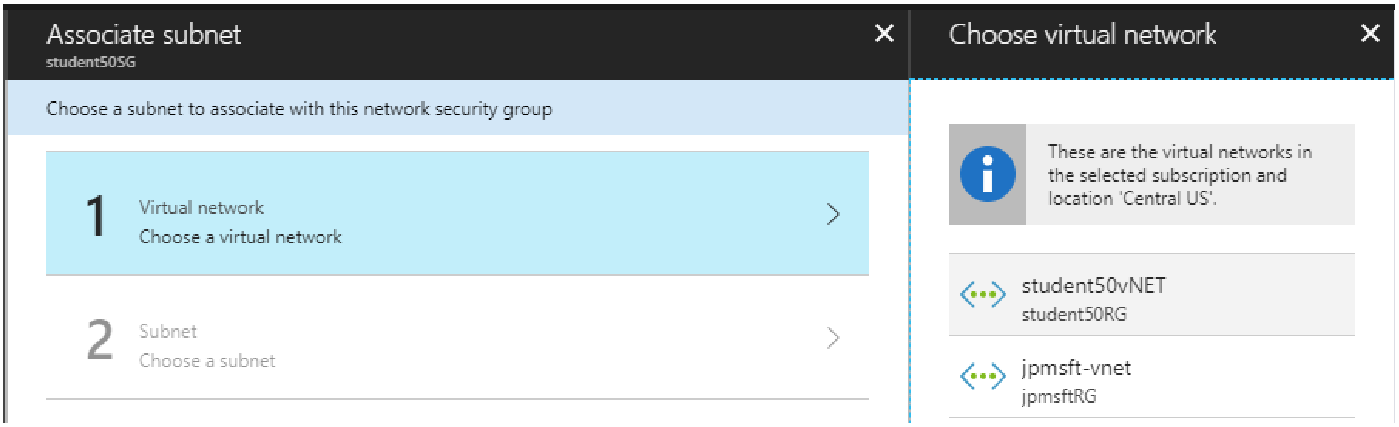
* Name: Personal-IP

Description:

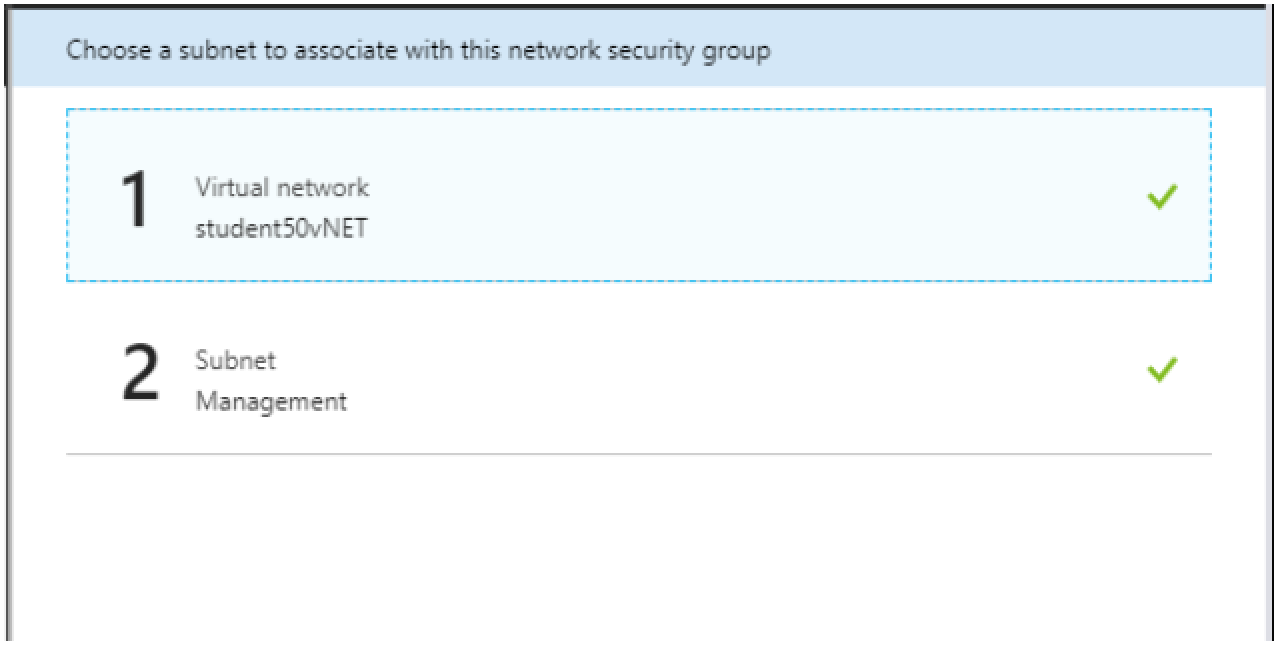
Step 7: Once the inbound security rule has been created, select "**Subnets".** This should be located 2 spaces below inbound security rules.



Step 8: Select your **vNET** which should be named **studentxxvNET**. Double check the **student ID xx.**



Step 9: Select the **Management** subnet to apply the security group to the management subnet. Click **OK**.

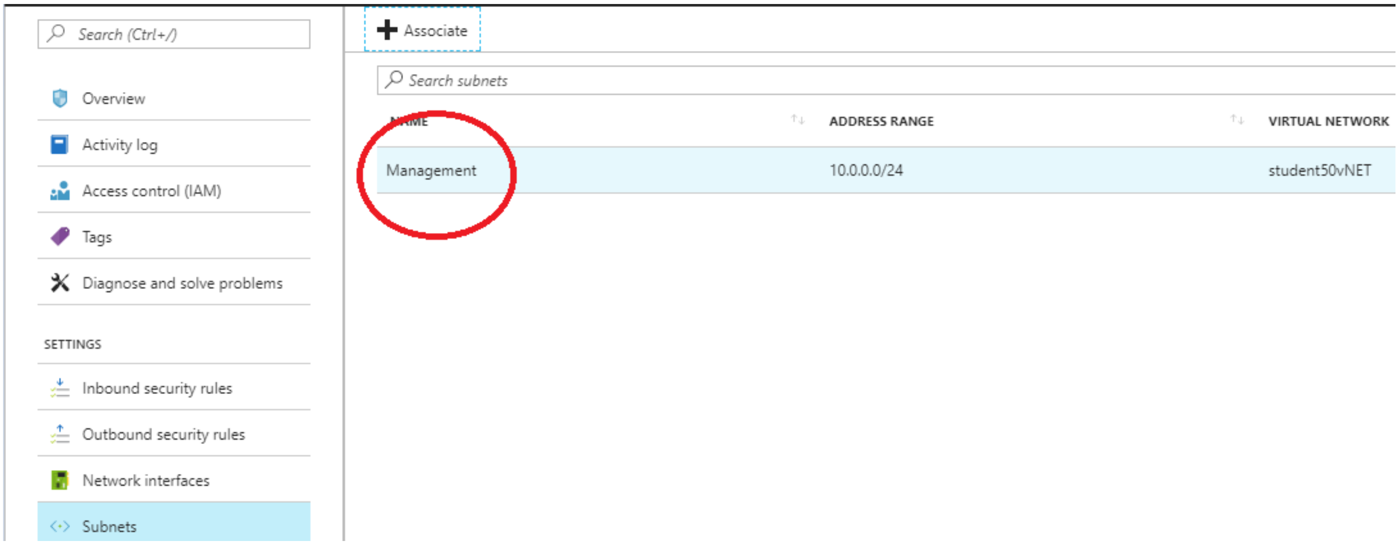


Choose a subnet to associate with this network security group

1 Virtual network
student50vNET ✓

2 Subnet
Management ✓

Step 10: Once the security group is successfully associated, you will see the management subnet as an association in the Subnets tab within **Network Security Groups**.



NAME	ADDRESS RANGE	VIRTUAL NETWORK
Management	10.0.0.0/24	student50vNET

End of Activity: In this Activity, you downloaded and applied the firewall configuration to both firewalls. You made changes to the route table to allow SSH management traffic to VM-Dev1 and VM-Web1. You also created a network security group to explicitly whitelist access to the firewall management interface. Now that you've completed this task, please move forward to the next activity.

Activity 3 – Outbound Access During Failover

In this activity you will:

- Run the Wget command from VM-Dev1
- Check the firewall traffic logs to see which firewall is passing traffic
- Release DHCP Lease on ethernet1/2 of the firewall passing traffic
- Check traffic logs of the second firewall to verify traffic is picked up after failover
- Delete the template deployment from Azure

Now that you've completed the previous tasks, it's time to begin testing. The goal of this test is to provide fault tolerance and secured access to the internet. The internal load balancer handles the fault tolerance and decides when a firewall is no longer suitable to receive traffic. The last step is to test failover using Wget. Failover is simulated by releasing the DHCP assigned IP address on the trust interface of the firewall that is passing traffic. Once DHCP is released the load balancer will send traffic to the next available firewall. Username:**paloalto** Password:**Paloalto123!**

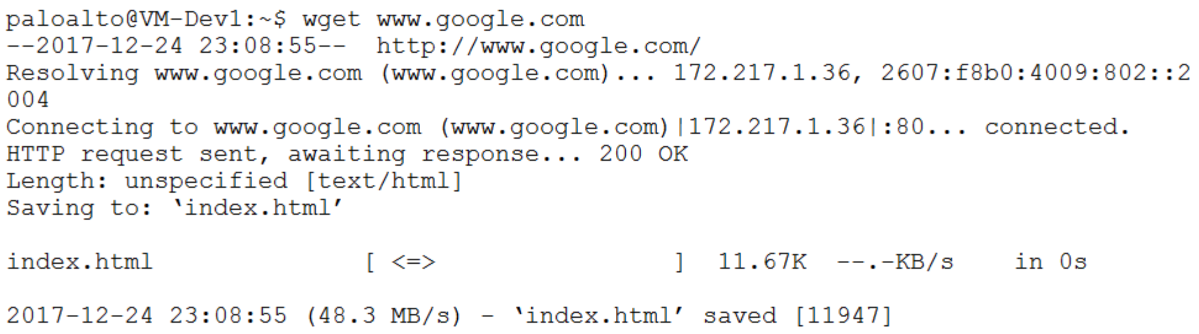
Task 1 – Run the Wget Command From VM-Dev1

For this task you will need to be logged into VM-Dev1 via SSH, and both firewalls simultaneously via https. If you are not already logged into these virtual machines, please do so now. Username:**paloalto** Password:**Paloalto123!**

Step 1: The private IP address for **VM-Dev1** is listed below. You will need this IP address when looking through the traffic logs on the firewall.

VM-Dev1: IP 10.0.6.50, **VM-Web1:** IP 10.0.4.50

Step 2: From **VM-Dev1**, run the "**wget www.google.com**" command. Hit the up arrow and press enter to run this command multiple times on **VM-Dev1**.

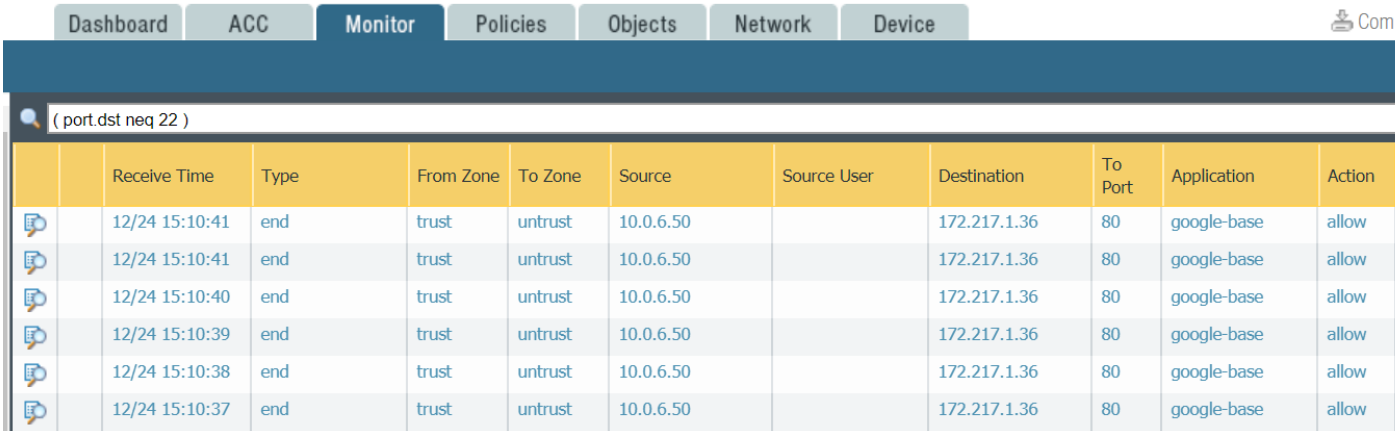


```
paloalto@VM-Dev1:~$ wget www.google.com
--2017-12-24 23:08:55--  http://www.google.com/
Resolving www.google.com (www.google.com)... 172.217.1.36, 2607:f8b0:4009:802::2
004
Connecting to www.google.com (www.google.com)|172.217.1.36|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [text/html]
Saving to: 'index.html'

index.html              [ <=>                 ]  11.67K  --.-KB/s    in 0s

2017-12-24 23:08:55 (48.3 MB/s) - 'index.html' saved [11947]
```

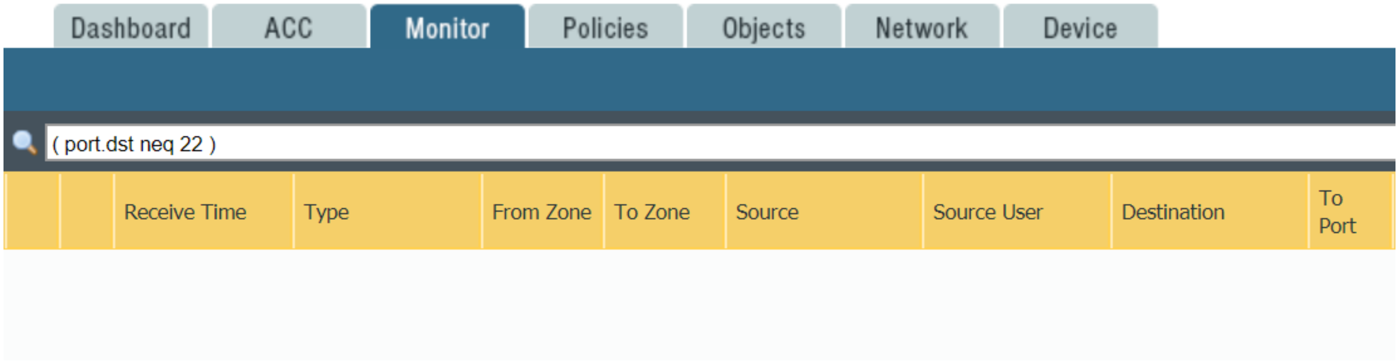
Step 3: From **VM-FW1**, go to the **Monitor** tab, select **Traffic**, and filter by (**port.dst neq 22**). Here you should see google-base traffic in the logs. If you don't see the traffic in **VM-FW1**, then check **VM-FW2.**



(port.dst neq 22)

		Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action
		12/24 15:10:41	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
		12/24 15:10:41	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
		12/24 15:10:40	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
		12/24 15:10:39	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
		12/24 15:10:38	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
		12/24 15:10:37	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow

Step 4: From **VM-FW2**, go to the **Monitor** tab, select **Traffic**, and filter by (**port.dst neq 22**). Here you shouldn't see any google based traffic in the logs. Now you know **VM-FW1** is passing traffic. In your lab, **VM-FW2** may be the actual firewall passing traffic during this step. If this is true, you can simply switch **VM-FW1** with **VM-FW2** for this rest of this task.

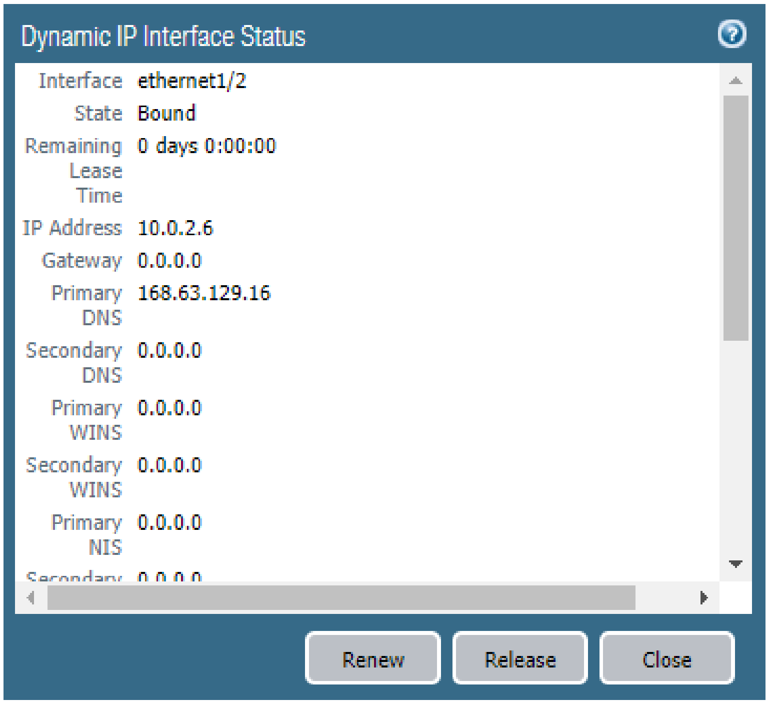


(port.dst neq 22)

		Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port

Step 5: Back on **VM-Dev1**, run the "**wget www.google.com**" command multiple times using the up-arrow + enter key sequence.

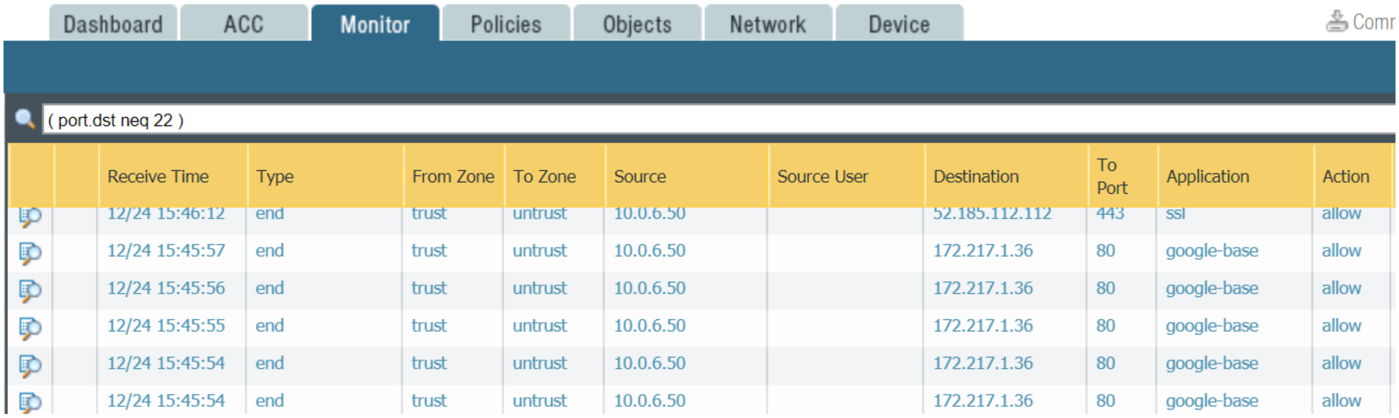
Step 6: From **VM-FW1** or the firewall that is passing traffic, Navigate to the **Network** tab, **Interfaces**. Click the **Dynamic-DHCP Client** link, then click **Release**. The release should be instantaneous.



Interface	ethernet1/2
State	Bound
Remaining Lease Time	0 days 0:00:00
IP Address	10.0.2.6
Gateway	0.0.0.0
Primary DNS	168.63.129.16
Secondary DNS	0.0.0.0
Primary WINS	0.0.0.0
Secondary WINS	0.0.0.0
Primary NIS	0.0.0.0
Secondary	0.0.0.0

Step 7: On **VM-Dev1**, continue to run the "**Wget www.google.com**" command multiple times using the up-arrow + enter key sequence.

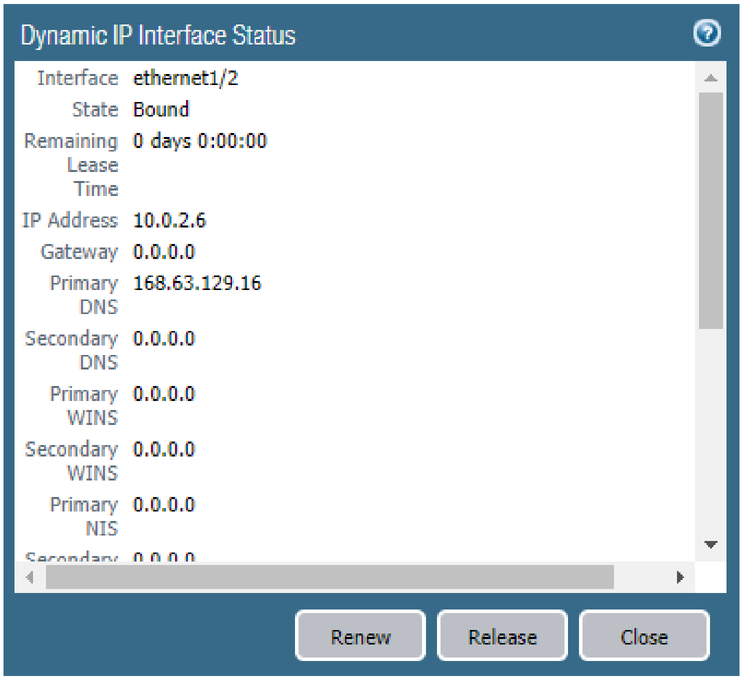
Step 8: On **VM-FW2**, navigate to the **Monitor** tab, then select **Traffic**. Notice the private IP address of **VM-Dev1** in the traffic logs. This shows that the load balancer has successfully failed over traffic to **VM-FW2.** Please remember to check the other firewall if you don't see any traffic.



(port.dst neq 22)

Receive Time	Type	From Zone	To Zone	Source	Source User	Destination	To Port	Application	Action
12/24 15:46:12	end	trust	untrust	10.0.6.50		52.185.112.112	443	ssl	allow
12/24 15:45:57	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
12/24 15:45:56	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
12/24 15:45:55	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
12/24 15:45:54	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow
12/24 15:45:54	end	trust	untrust	10.0.6.50		172.217.1.36	80	google-base	allow

Step 9: Renew the DHCP lease on the interface of the firewall that you released the DHCP lease.



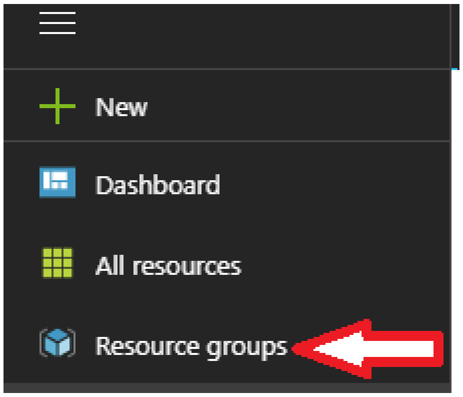
Dynamic IP Interface Status

Interface	ethernet1/2
State	Bound
Remaining Lease Time	0 days 0:00:00
IP Address	10.0.2.6
Gateway	0.0.0.0
Primary DNS	168.63.129.16
Secondary DNS	0.0.0.0
Primary WINS	0.0.0.0
Secondary WINS	0.0.0.0
Primary NIS	0.0.0.0
Secondary	0.0.0.0

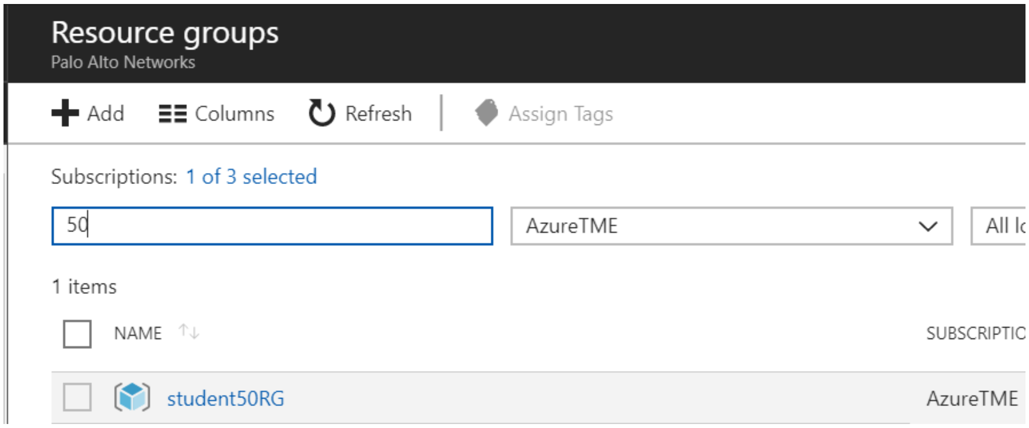
In this task you used Wget to test internet access. This test was performed to demonstrate how Azure based high availability handles outbound transaction requests during a failover.

Task 2 – Delete the deployment from Azure

Step 1: Now that the lab is complete, delete your template deployment from Azure. Within Azure in the left side pane click **Resource Groups**.



Step 2: Select your **Subscription** and filter by **student ID**



Subscriptions: 1 of 3 selected

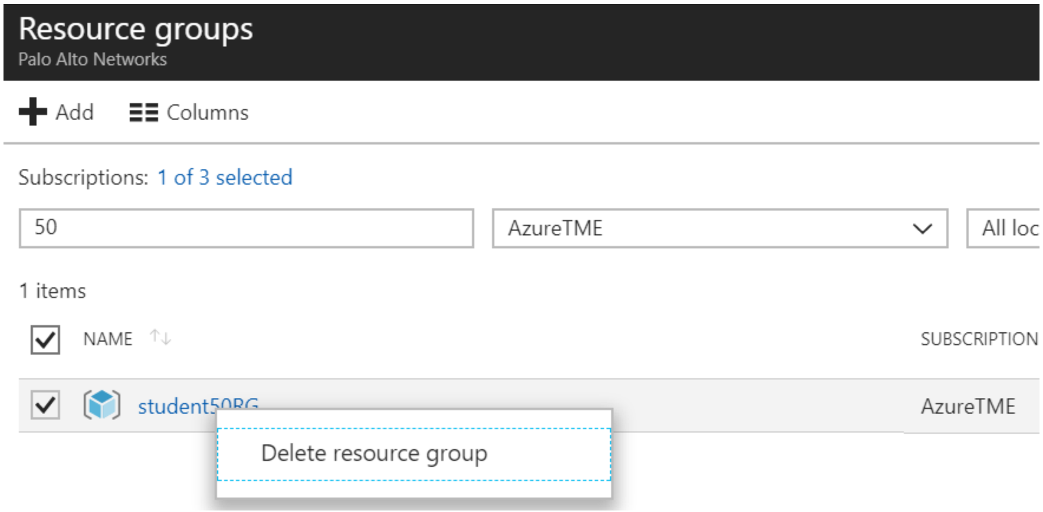
50

AzureTME

1 items

NAME	SUBSCRIPTIC
student50RG	AzureTME

Step 3: Right click your resource group and click **Delete resource group**



Subscriptions: 1 of 3 selected

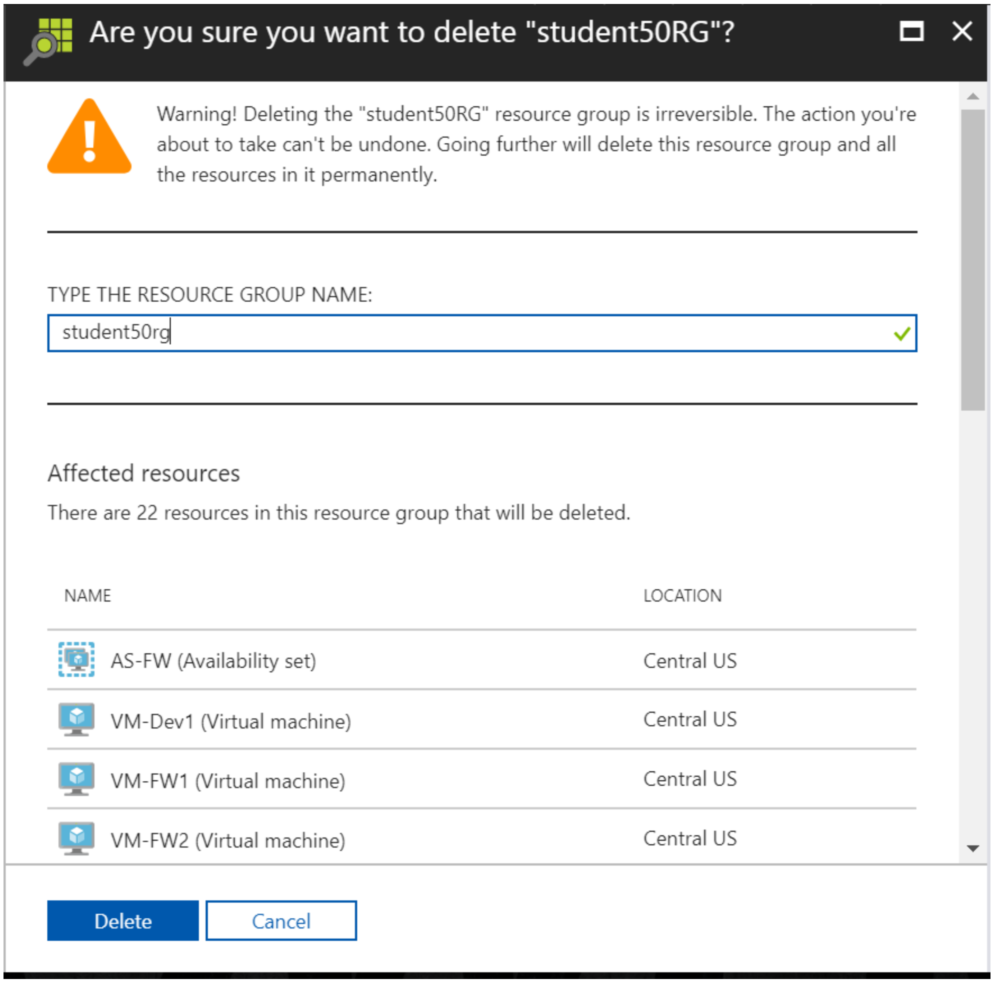
50

AzureTME

1 items

☑ NAME	SUBSCRIPTION
☑ student50RG	AzureTME

Step 4: Type in the resource group name. Once you do this the delete button can be clicked. Be sure you delete the correct resource group.



Are you sure you want to delete "student50RG"?

Warning! Deleting the "student50RG" resource group is irreversible. The action you're about to take can't be undone. Going further will delete this resource group and all the resources in it permanently.

TYPE THE RESOURCE GROUP NAME:

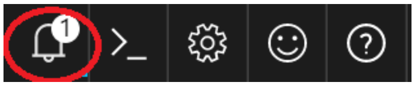
student50rg ✓

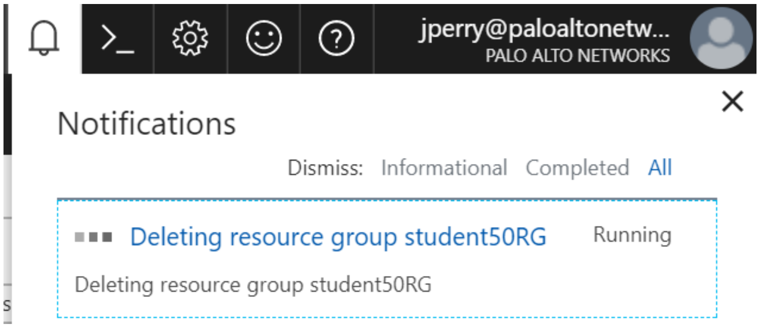
Affected resources

There are 22 resources in this resource group that will be deleted.

NAME	LOCATION
AS-FW (Availability set)	Central US
VM-Dev1 (Virtual machine)	Central US
VM-FW1 (Virtual machine)	Central US
VM-FW2 (Virtual machine)	Central US

Step 5: Click the bell icon in the top right to view status.





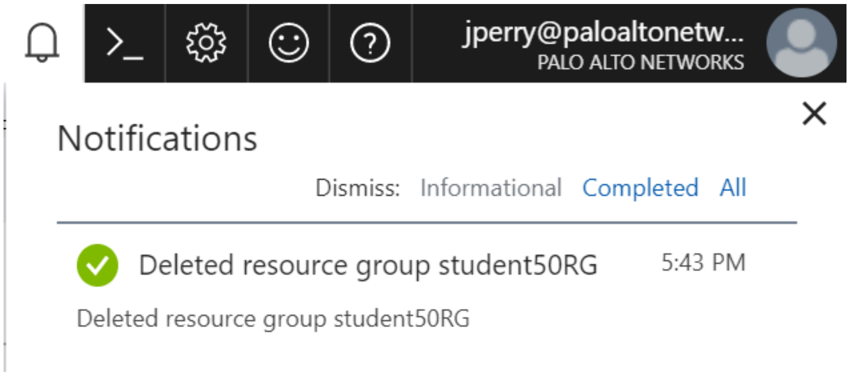
Notifications

Dismiss: Informational Completed All

Deleting resource group student50RG Running

Deleting resource group student50RG

Step 6: When complete you will receive the output in the screenshot below. You do not have to wait for it to complete before logging out of Azure Portal.



Notifications

Deleted resource group student50RG 5:43 PM

Deleted resource group student50RG

End of Lab: During this lab you deployed an Azure load balancer, two Palo Alto Networks firewalls, and two Linux servers using an Azure ARM template. After verifying your resources were successfully deployed, you then imported and loaded the firewall configuration files on each firewall. To be able to manage the two servers you added a route to send any traffic destined to your public IP directly to the internet bypassing the load balancer. You then ran Wget on VM-Dev1 server multiple times. While running Wget you simulated a failover on VM-FW1 and verified that traffic was now picked up by VM-FW2. This lab demonstrated how to secure outbound access requests using Palo Alto Networks Layer-7 inspection capabilities while leveraging Azure load balancers for high-availability. Being able to inspect outbound traffic is critical not only to prevent malware and anti-virus, but also to protect intellectual property by preventing data exfiltration. This topology is very flexible and can also be used with an existing vNet deployment.