

In-Line Traffic Analysis Appliance

Building and deploying an in-line network analysis appliance in Azure
(though, really any network where you can manage traffic routing)

Why are we here?

Network taps in cloud environments are not as easy as they are when on premises.

Vendor appliances often come with high licensing costs.

We have the tools.

Who Am I?

Ken Netzorg

25 Years across various responsibilities ranging from network administration to DBA to C# development.

I like to solve problems, IT related.

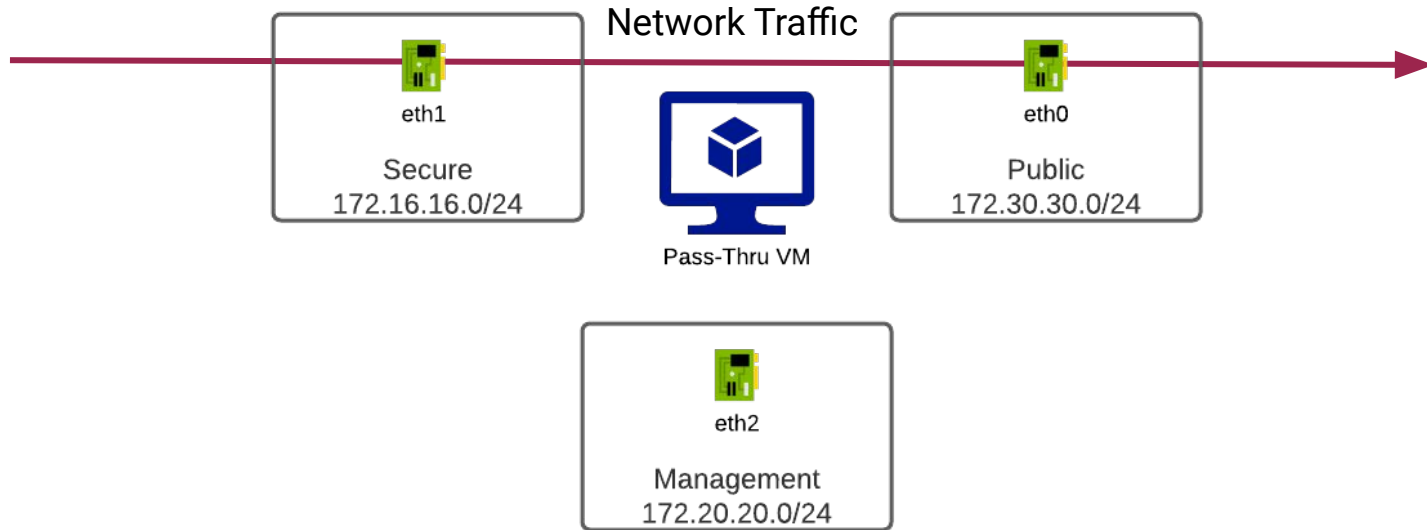
Currently the Director of Technology Operations and Security at DecisivEdge and heavily involved in Azure infrastructure and security.

Certs: CISA/CISSP

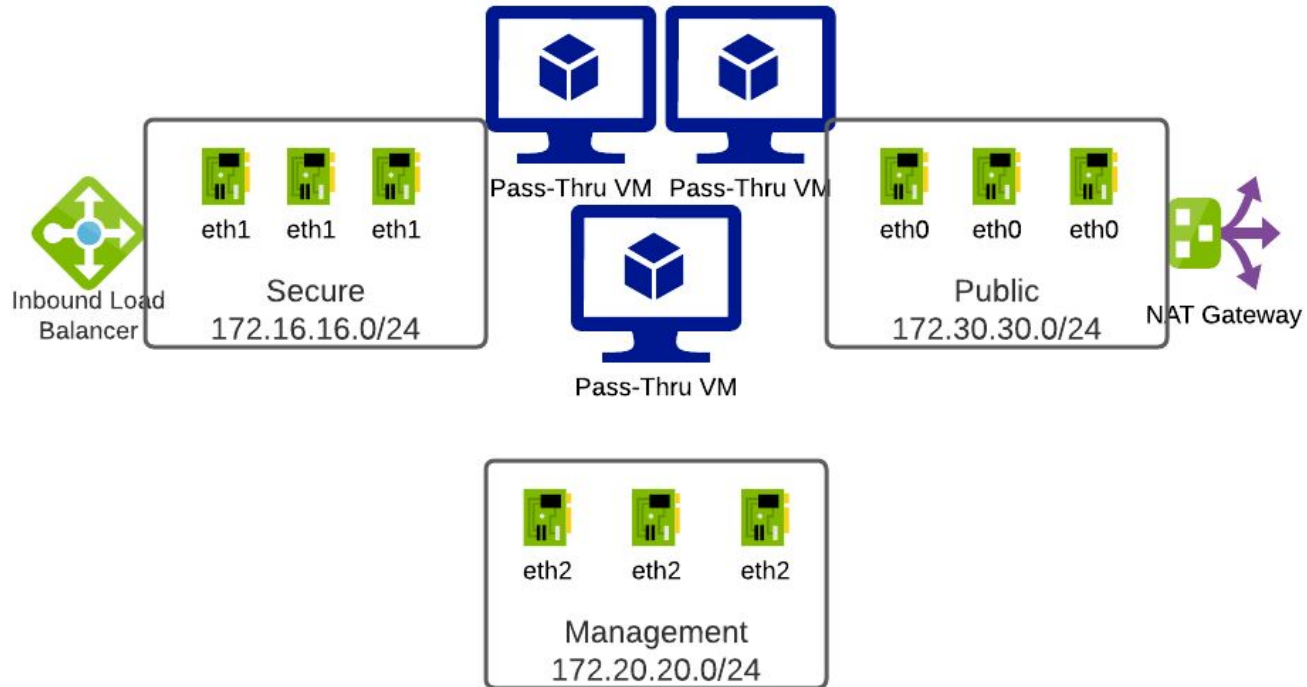
Overview

- Architecture
- Azure Objects
- OS Modifications
- Monitoring Traffic
- High Availability Options
- Enhancements/Hardening/The Future
- Conclusion
- Questions

Architecture - Basic



Architecture - Advanced



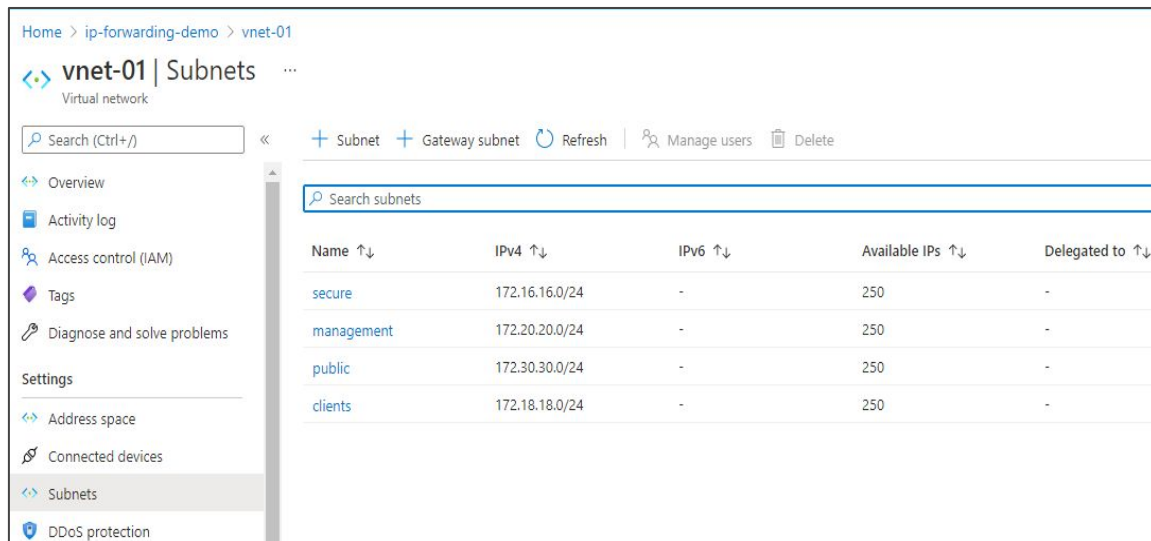
Let's build this

Azure Objects - Network

1 Virtual Network

3 Subnets for Appliance

1 Subnet for testing/validation



The screenshot displays the Azure portal interface for managing a virtual network. The breadcrumb navigation at the top shows the path: Home > ip-forwarding-demo > vnet-01. The main heading is 'vnet-01 | Subnets', with 'Virtual network' indicated below it. A search bar with the placeholder 'Search (Ctrl+J)' is present. Action buttons include '+ Subnet', '+ Gateway subnet', 'Refresh', 'Manage users', and 'Delete'. A left-hand navigation pane lists various options: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Address space, Connected devices, Subnets (which is highlighted), and DDoS protection. The main content area features a 'Search subnets' bar and a table listing the subnets.

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓
secure	172.16.16.0/24	-	250	-
management	172.20.20.0/24	-	250	-
public	172.30.30.0/24	-	250	-
clients	172.18.18.0/24	-	250	-

Azure Objects - Compute

Check for NIC quantity. I used the “Standard B2s” as it is the smallest server and has a 3 NIC capacity.

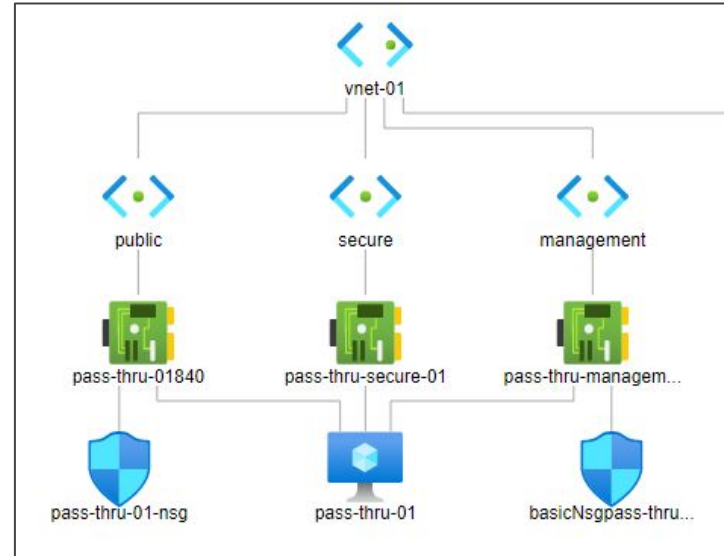
NOTE: Set your first NIC as the Outbound NIC

Virtual machine		Networking	
Computer name	pass-thru-01	Public IP address	-
Health state	-	Public IP address (IPv6)	-
Operating system	Linux	Private IP address	172.30.30.4
Publisher	debian	Private IP address (IPv6)	-
Offer	debian-10	Virtual network/subnet	vnet-01/public
Plan	10-backports	DNS name	-
VM generation	V1	Size	
Host group	None	Size	Standard B2s
Host	-	vCPUs	2
Proximity placement group	-	RAM	4 GiB
Colocation status	N/A		

Azure Objects - Additional NICs

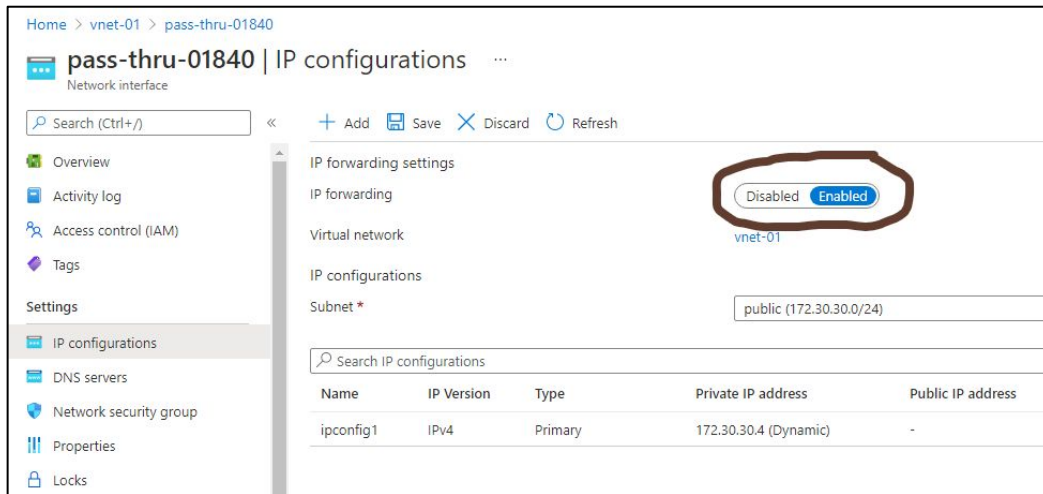
2 Additional NICs

- eth1 in Secure Subnet
- eth2 in Management Subnet



Azure Objects - IP Forwarding

Enable 'IP Forwarding' option on the Outbound, eth0 NIC.



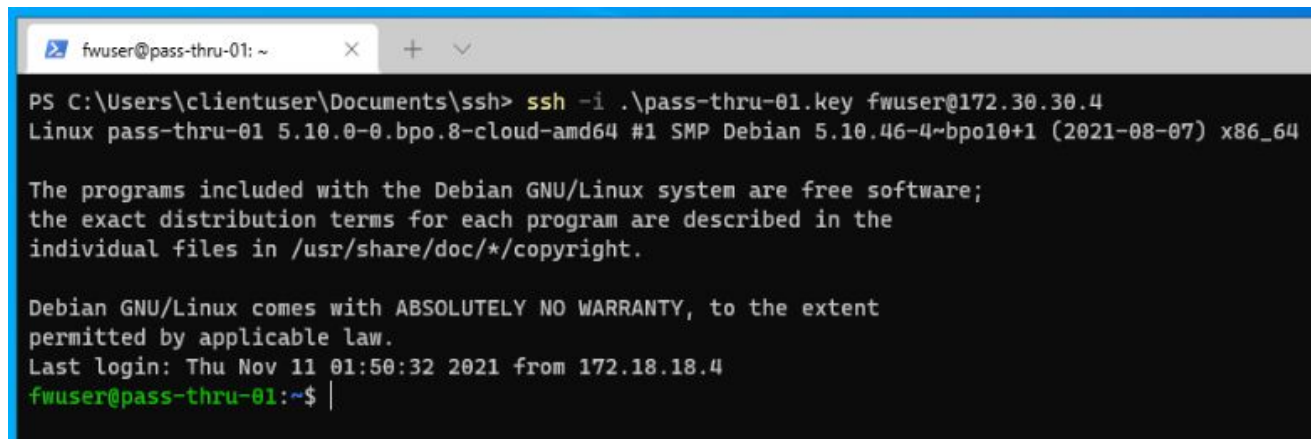
The screenshot shows the Azure portal interface for the network interface 'pass-thru-01840'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Settings, IP configurations (selected), DNS servers, Network security group, Properties, and Locks. The main content area is titled 'pass-thru-01840 | IP configurations' and includes a search bar and action buttons: Add, Save, Discard, and Refresh. Under the 'IP forwarding settings' section, the 'IP forwarding' toggle is highlighted with a red circle and is currently set to 'Enabled'. Below this, the 'Virtual network' is listed as 'vnet-01'. The 'IP configurations' section shows a 'Subnet' dropdown set to 'public (172.30.30.0/24)'. A table below lists the IP configurations:

Name	IP Version	Type	Private IP address	Public IP address
ipconfig1	IPv4	Primary	172.30.30.4 (Dynamic)	-

Log into the OS

On first connection, SSH to the outbound IP interface, eth0.

Routing is not in your favor if you attempt an alternate.

A terminal window with a blue title bar. The title bar contains a tab labeled 'fwuser@pass-thru-01: ~' and standard window controls (close, maximize, minimize). The terminal content shows an SSH command being executed from a Windows command prompt, followed by the Debian system's boot information, a license notice, and the login prompt.

```
PS C:\Users\clientuser\Documents\ssh> ssh -i .\pass-thru-01.key fwuser@172.30.30.4
Linux pass-thru-01 5.10.0-0.bpo.8-cloud-amd64 #1 SMP Debian 5.10.46-4~bpo10+1 (2021-08-07) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Nov 11 01:50:32 2021 from 172.18.18.4
fwuser@pass-thru-01:~$ |
```

OS Changes - Kernel IP Forwarding

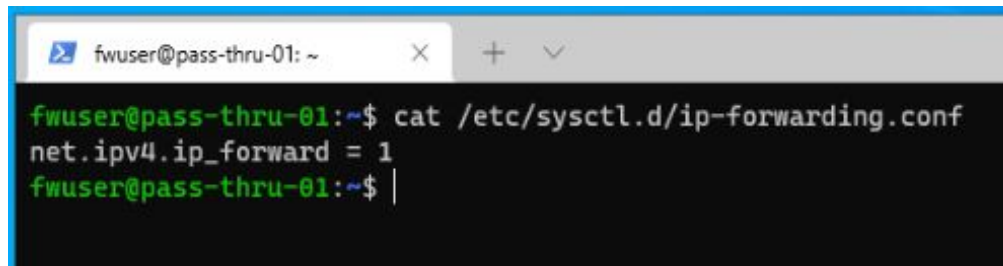
Configure the Kernel so it knows it is a router.

Non-persistent but immediate:

```
sysctl -w net.ipv4.ip_forward=1
```

Persistent, requires a restart

Write it to the /etc/sysctl.d directory

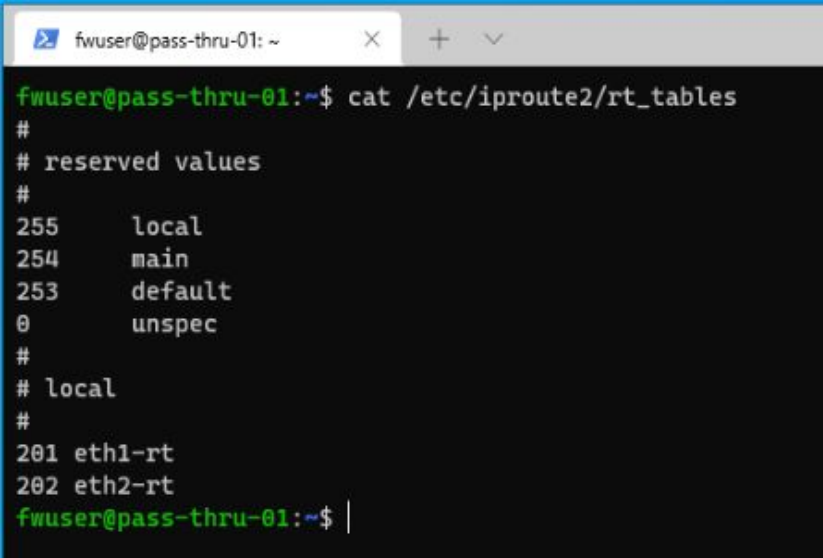
A terminal window with a blue title bar. The title bar contains a terminal icon, the text 'fwuser@pass-thru-01: ~', and window control buttons (close, maximize, and a dropdown arrow). The terminal content shows a user prompt 'fwuser@pass-thru-01:~\$' followed by the command 'cat /etc/sysctl.d/ip-forwarding.conf'. The output of the command is 'net.ipv4.ip_forward = 1'. The prompt is followed by a vertical bar '|'.

```
fwuser@pass-thru-01: ~  
fwuser@pass-thru-01:~$ cat /etc/sysctl.d/ip-forwarding.conf  
net.ipv4.ip_forward = 1  
fwuser@pass-thru-01:~$ |
```

OS Changes - IP Routing

Define route tables:

Used 201 and 202 for eth1 and eth2 respectively, though the values are not significant within the 1-252 range.

A terminal window with a title bar showing 'fwuser@pass-thru-01: ~'. The terminal output shows the contents of the file /etc/iproute2/route_tables. The output includes comments for reserved values and a table of route tables. The table lists values 255 (local), 254 (main), 253 (default), and 0 (unspec). Below this, it lists 'local' and then two specific route tables: '201 eth1-rt' and '202 eth2-rt'. The prompt 'fwuser@pass-thru-01:~\$' is visible at the bottom.

```
fwuser@pass-thru-01:~$ cat /etc/iproute2/route_tables
#
# reserved values
#
255    local
254    main
253    default
0      unspec
#
# local
#
201 eth1-rt
202 eth2-rt
fwuser@pass-thru-01:~$ |
```

OS Changes - IP Routing

For each additional interface, eth1 and eth2, specify default routes and rules for traffic handling.

```
fwuser@pass-thru-01: ~  
fwuser@pass-thru-01:~$ cat /etc/network/interfaces.d/eth1  
auto eth1  
iface eth1 inet dhcp  
    post-up ip route add default via 172.16.16.1 table eth1-rt  
    post-up ip rule add from 172.16.16.4/32 table eth1-rt  
    post-up ip rule add to 172.16.16.4/32 table eth1-rt  
fwuser@pass-thru-01:~$ |
```

```
fwuser@pass-thru-01: ~  
fwuser@pass-thru-01:~$ cat /etc/network/interfaces.d/eth2  
auto eth2  
iface eth2 inet dhcp  
    post-up ip route add default via 172.20.20.1 table eth2-rt  
    post-up ip rule add from 172.20.20.4/32 table eth2-rt  
    post-up ip rule add to 172.20.20.4/32 table eth2-rt  
fwuser@pass-thru-01:~$ |
```

OS Changes - IP Routing

Once interface tables have been created,
reboot and validate the configuration.

SSH to the management IP!

```
PS C:\Users\clientuser\Documents\ssh> ssh -i .\pass-thru-01.key fwuser@172.20.20.4
Linux pass-thru-01 5.10.0-0.bpo.8-cloud-amd64 #1 SMP Debian 5.10.46-4~bpo10+1 (2021-08-07) x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Nov 10 01:58:11 2021 from 172.18.18.4
fwuser@pass-thru-01:~$ |
```


OS Changes - IPTables/Firewall

Linux distribution firewall variations

You may need to install iptables

Depending on distribution iptables-persistent (Debian ≤ 10) or iptables-services (RHEL ≤ 7) packages if you want it to start after reboot.

Iptables files may be in /etc/sysconfig (RHEL) or /etc/iptables (Debian)

NFTables, the new IPTables, is handled a little differently

OS Changes - IPTables/Firewall

Create the 'nat' chain

Add postrouting line to handle outbound address translation

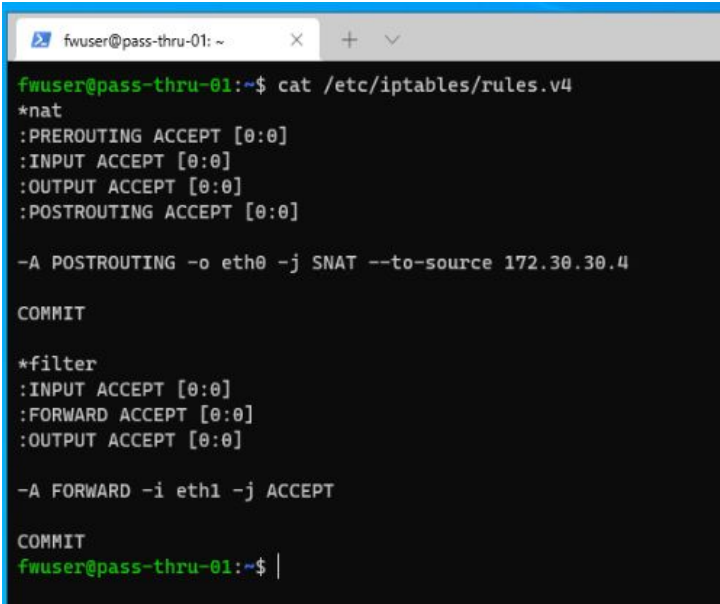
```
-A POSTROUTING -o eth0 -j SNAT --to-source 172.30.30.4
```

Create/modify 'filter' chain

Accept all traffic inbound

(use network NSGs/ACLs to do the real restriction)

```
-A FORWARD -i eth1 -j ACCEPT
```

A terminal window titled 'fwuser@pass-thru-01: ~' with standard window controls. The terminal shows the command 'cat /etc/iptables/rules.v4' and its output. The output defines two chains: 'nat' and 'filter'. The 'nat' chain includes rules for PREROUTING, INPUT, OUTPUT, and POSTROUTING, all set to ACCEPT, followed by a SNAT rule in the POSTROUTING chain and a COMMIT command. The 'filter' chain includes rules for INPUT, FORWARD, and OUTPUT, all set to ACCEPT, followed by a FORWARD rule in the FORWARD chain and a COMMIT command.

```
fwuser@pass-thru-01:~$ cat /etc/iptables/rules.v4
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

-A POSTROUTING -o eth0 -j SNAT --to-source 172.30.30.4

COMMIT

*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

-A FORWARD -i eth1 -j ACCEPT

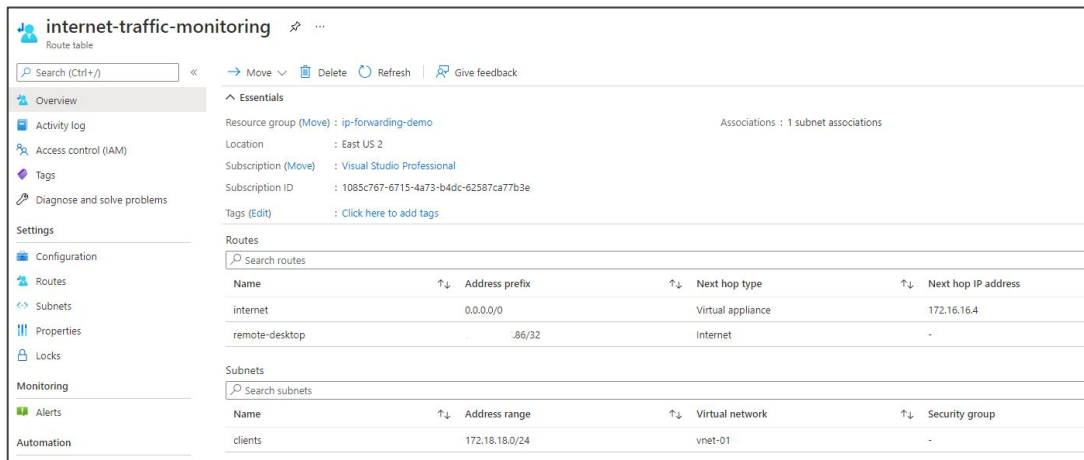
COMMIT
fwuser@pass-thru-01:~$
```

Azure Objects - IP Routing

Forwarding should now be configured, now to get traffic to the inbound port.

Create a default route using 0.0.0.0/0 and specify next hop as the eth1 IP address.

Assign it to any subnets that need their traffic monitored.



The screenshot displays the Azure portal interface for a route table named 'internet-traffic-monitoring'. The left sidebar contains navigation links for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration, Routes, Subnets, Properties, Locks), Monitoring (Alerts), and Automation. The main content area shows the 'Overview' tab with details about the resource group, location, subscription, and subscription ID. Below this, the 'Routes' section displays a table of routes, and the 'Subnets' section displays a table of subnets.

Routes

Name	Address prefix	Next hop type	Next hop IP address
internet	0.0.0.0/0	Virtual appliance	172.16.16.4
remote-desktop	.86/32	Internet	-

Subnets

Name	Address range	Virtual network	Security group
clients	172.18.18.0/24	vnet-01	-

We can forward traffic.
Now what?

TCPDump

Running a tcpdump

`tcpdump -i any host <client>`

```
fwuser@pass-thru-01: ~  
fwuser@pass-thru-01:~$ sudo tcpdump -i any host 172.18.18.4 and !(tcp port 22)  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes  
03:01:43.124104 IP client-01.internal.cloudapp.net.49700 > 20.60.88.36.https: Flags [P.], seq 736279204, ack 1824743231, win 2053, length 459  
03:01:43.126599 IP 20.60.88.36.https > client-01.internal.cloudapp.net.49700: Flags [P.], seq 1:507, ack 459, win 16415, length 506  
03:01:43.127469 IP client-01.internal.cloudapp.net.49700 > 20.60.88.36.https: Flags [P.], seq 459:918, ack 507, win 2051, length 459  
03:01:43.129194 IP 20.60.88.36.https > client-01.internal.cloudapp.net.49700: Flags [P.], seq 507:1013, ack 918, win 16414, length 506  
03:01:43.131246 IP client-01.internal.cloudapp.net.49700 > 20.60.88.36.https: Flags [P.], seq 918:1376, ack 1013, win 2049, length 458  
03:01:43.134469 IP 20.60.88.36.https > client-01.internal.cloudapp.net.49700: Flags [P.], seq 1013:12249, ack 1376, win 16417, length 11236  
03:01:43.135281 IP client-01.internal.cloudapp.net.49700 > 20.60.88.36.https: Flags [.], ack 12249, win 2053, length 0  
03:01:50.413050 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [S], seq 3915695718, win 65535, options [mss 1418,nop,wscale 8,nop,nop,sackOK], length 0  
03:01:50.472192 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [S.], seq 2317592902, ack 3915695719, win 65535, options [mss 1440,nop,wscale 8,nop,nop,sackOK], length 0  
03:01:50.473345 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [.], ack 1, win 1024, length 0  
03:01:50.475559 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [P.], seq 1:212, ack 1, win 1024, length 211  
03:01:50.535283 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [.], seq 1:1399, ack 212, win 2052, length 1398  
03:01:50.535303 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [.], seq 1399:2797, ack 212, win 2052, length 1398  
03:01:50.535305 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [.], seq 2797:4195, ack 212, win 2052, length 1398  
03:01:50.535307 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [.], seq 4195:5593, ack 212, win 2052, length 1398  
03:01:50.535309 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [.], seq 5593:6991, ack 212, win 2052, length 1398  
03:01:50.535311 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [P.], seq 6991:7116, ack 212, win 2052, length 125  
03:01:50.536346 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [.], ack 7116, win 1024, length 0  
03:01:50.555232 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [P.], seq 212:370, ack 7116, win 1024, length 158  
03:01:50.614894 IP 13.87.188.105.https > client-01.internal.cloudapp.net.50407: Flags [P.], seq 7116:7167, ack 370, win 2051, length 51  
03:01:50.615003 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [.], ack 7167, win 1023, length 0  
03:01:50.620658 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [P.], seq 370:838, ack 7167, win 1023, length 468  
03:01:50.620666 IP client-01.internal.cloudapp.net.50407 > 13.87.188.105.https: Flags [P.], seq 838:2090, ack 7167, win 1023, length 1252  
03:01:50.643333 IP client-01.internal.cloudapp.net.50408 > 13.107.42.16.https: Flags [S], seq 355156978, win 64240, options [mss 1418,nop,wscale 8,nop,nop,sackOK], length 0  
03:01:50.644365 IP 13.107.42.16.https > client-01.internal.cloudapp.net.50408: Flags [S.], seq 2610646305, ack 355156979, win 65535, options [mss 1440,nop,wscale 8,nop,nop,sackOK], length 0  
03:01:50.645765 IP client-01.internal.cloudapp.net.50408 > 13.107.42.16.https: Flags [.], ack 1, win 2053, length 0  
03:01:50.646380 IP client-01.internal.cloudapp.net.50408 > 13.107.42.16.https: Flags [P.], seq 1:518, ack 1, win 2053, length 517  
03:01:50.646906 IP 13.107.42.16.https > client-01.internal.cloudapp.net.50408: Flags [.], ack 518, win 16426, length 0  
03:01:50.648184 IP 13.107.42.16.https > client-01.internal.cloudapp.net.50408: Flags [.], seq 1:1399, ack 518, win 16426, length 1398  
03:01:50.648186 IP 13.107.42.16.https > client-01.internal.cloudapp.net.50408: Flags [.], seq 1399:2797, ack 518, win 16426, length 1398
```

Suricata In-Line

Adding the following line to the IPTables configuration
(assuming you are using an installation package that was compiled with NFQ support)

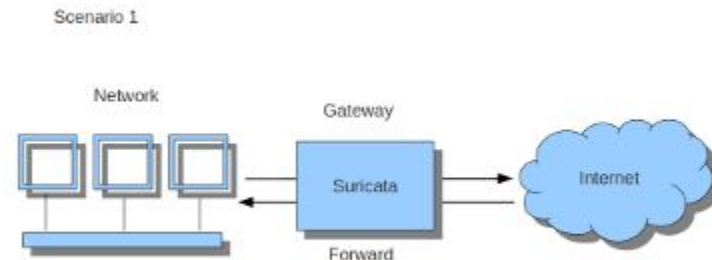
```
-A FORWARD -j NFQUEUE
```

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]

-A FORWARD -j NFQUEUE

-A FORWARD -i eth1 -j ACCEPT

COMMIT
```



Zeek In-Line (YMMV)

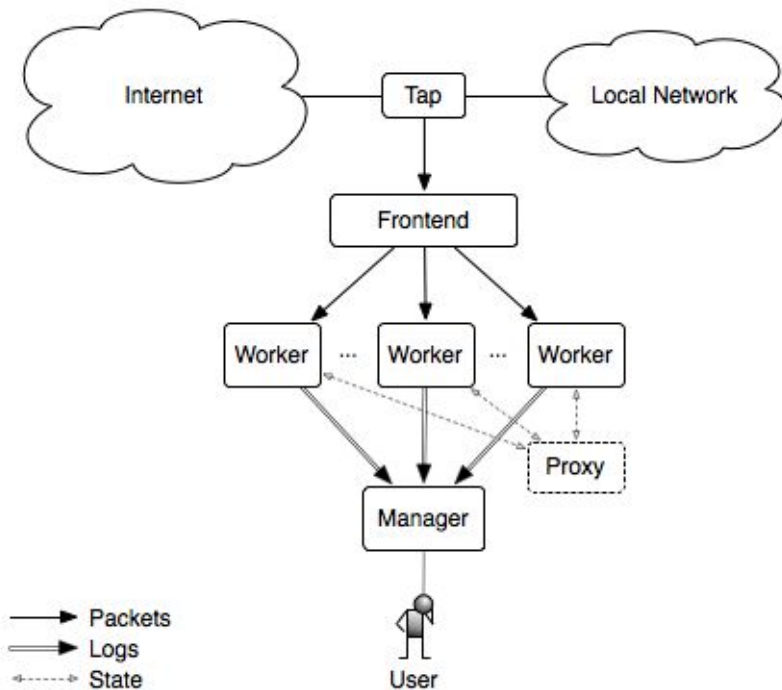
(FKA Bro)

This installation would be the combination of Tap, Frontend, and Worker processes.

Location of the Manager process should be on a separate server, with logs sent via the Management interface, eth2.

Installation and use of PF_RING outlined on the Zeek docs under 'Zeek Cluster Setup'

<https://docs.zeek.org/en/master/cluster-setup.html>



Going High Performance

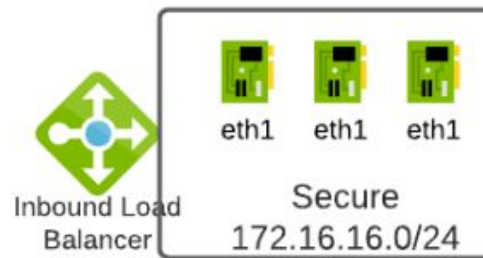
Inbound Load Balancer

Use a Basic (non-complex design) or Standard (Complex, multi-region) level Load Balancer to spread traffic across multiple appliances.

Provides fault tolerance and horizontal scaling capabilities.

Challenges:

- Uses a polling heartbeat of tcp or http to know if an appliance is healthy. Need to determine the best service to run on the appliance to answer the heartbeat.
- Need to map each port that is being forwarded through the load balancer.
- May face challenges overlaying NSGs on the Secure subnet.



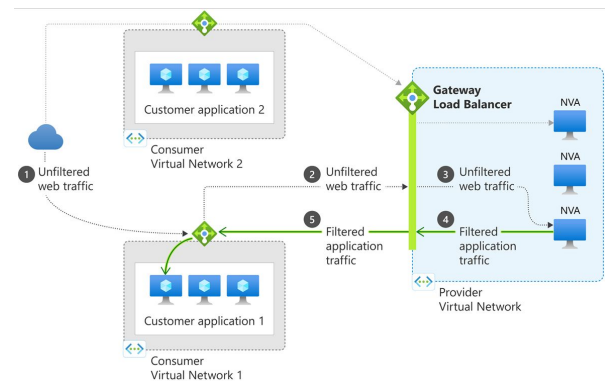
Inbound Gateway Load Balancer (Preview)

No personal experience, though the documentation states it will link to 'custom appliance' backends.

May need to 'chain' a Standard level load balancer ahead of this service.

Primary purpose for including this service here based on the documentation:

- Integrate virtual appliances transparently into the network path.
- Easily add or remove network virtual appliances in the network path.
- Scale with ease while managing costs.
- Improve network virtual appliance availability.

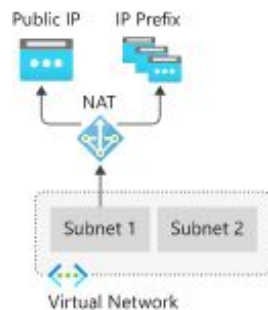


Source: <https://docs.microsoft.com/en-us/azure/load-balancer/gateway-overview>

Outbound Virtual Network “NAT Gateway”

Use Microsoft’s preferred method for outbound connectivity.

Beware the SLA.....?



Azure's outbound connectivity methods

Outbound connectivity to the internet can be enabled in the following ways:

#	Method	Type of port allocation	Production-grade?	Rating
1	Using the frontend IP address(es) of a Load Balancer for outbound via Outbound rules	Static, explicit	Yes, but not at scale	OK
2	Associating a NAT gateway to the subnet	Static, explicit	Yes	Best
3	Assigning a Public IP to the Virtual Machine	Static, explicit	Yes	OK
4	Using the frontend IP address(es) of a Load Balancer for outbound (and inbound)	Implicit	No	Second worst
5	Using default outbound access	Implicit	No	Worst

Source: <https://docs.microsoft.com/en-us/azure/load-balancer/load-balancer-outbound-connections>

Pricing and SLA

For pricing details, see [Virtual Network pricing](#). NAT data path is at least 99.9% available.

Source: <https://docs.microsoft.com/en-us/azure/virtual-network/nat-gateway/nat-overview>

Agreed SLA level: % (enter SLA level and hit the <enter> key)

SLA level of 99.9 % uptime/availability results in the following periods of allowed downtime/unavailability:

- Daily: 1m 26s
- Weekly: 10m 4s
- Monthly: 43m 49s
- Quarterly: 2h 11m 29s
- Yearly: 8h 45m 56s

Source: <https://uptime.is>



Next Steps/Considerations

Network Security Groups (Network ACLs)

Add/configure Network Security Groups (ACLs) on Secure, Public, Management subnets.

Block all inbound traffic on **Public subnet**, this is outbound only.

Block all but the ports you wish to be forwarded on the **Secure subnet**. NOTE: May not be compatible with basic or standard load balancer configuration. Simply having an NSG defined on that subnet killed all traffic during testing, YMMV.

Limit SSH/Management traffic into the **Management subnet** to trusted sources.

Management/SSH

Update SSH daemon or any other services to listen only on management, eth2, interface

This avoids the potential of forwarded SSH traffic received on eth1 from being handled by the local host

NFTables

Unfortunately, no configuration to share yet. At some point it will have to happen as iptables is supplanted by NFTables on all the newer distributions.

Wrapping Up

Conclusion

- We can build our own appliance using cost effective resources
- The appliance can be outfitted with our own products we choose
- Cloud services can be implemented to increase availability and throughput

Questions/Comments?

Contact

Ken Netzorg

knetzorg@gmail.com

zorg_the_blue (discord)

Good Luck.

Happy Hunting!

(Copy of slides will be posted to my
github repo: zorg-the-blue)
