

Course Title	Cloud Security
Topic	Deploying and securing an Ubuntu Web Server behind Check Point Firewall in Azure Cloud.
Date	16/03/2022
Content Owner	Mark Ashwin
Batch	13

Objectives

To host a web server in the azure cloud behind the checkpoint firewall.

Lab Environment

Virtual Labs using Azure cloud: Ubuntu Server 20.04 LTS, CloudGuard standalone FW.

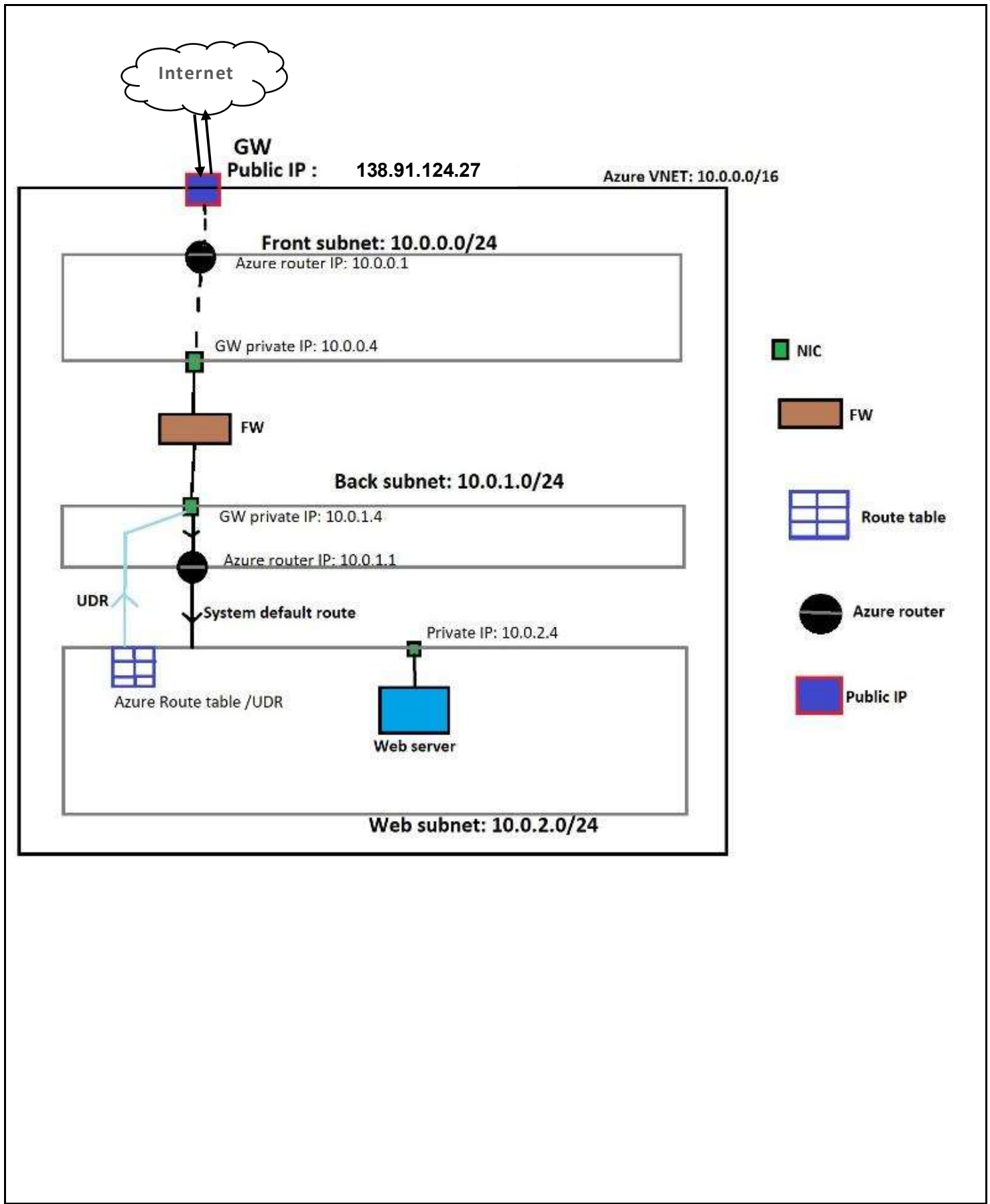
Azure Services used

1. Virtual Networks
2. IP route tables / User Defined Routes
3. Ubuntu Server 20.04 VM
4. CloudGuard Standalone Firewall VM (CPFWM)

Network Topology

The Azure cloud network is designed in the following way: -

1. Virtual Network – 10.0.0.0/16
 - a. Frontend subnet- 10.0.0.0/24:
 - i. CPFWM External interface's Private IP – 10.0.0.4/24
 - b. Backend subnet- 10.0.1.0/24:
 - i. CPFWM Internal interface's Private IP– 10.0.1.4/24
 - c. Web subnet- 10.0.2.0/24:
 - i. Ubuntu Server 20.04 VM Private IP – 10.0.2.4/24
 - ii. Azure Route table – Web-RT
2. CPFWM External interface's Public IP –138.91.124.27




Configuring a Virtual Network


1. Go to Azure Marketplace > Search “Virtual Network”.
2. Select “Virtual Network” > Click on “Create”
3. Select a resource group for your Virtual Network if you have created one already. I have already created one: “myVNET”.
4. If you have not created a resource group, you can simply select the “create new” link next to the “resource group” field and give a name to it.
5. Give the name of the Virtual Network. I have given the name as “myVNET”.
6. Give the region of the VNET as “East US”.
7. Click on “Next”
8. We give the IP address range of the Network as “10.0.0.0/16”.
9. We are going to divide this network into 3 subnets –
 - a. Frontend – 10.0.0.0/24, this is the subnet where firewall’s external interface will reside in which will be used to communicate with the internet.
 - b. Backend – 10.0.1.0/24, this is the subnet that will connect the Firewall’s internal interface with the internal networks.
 - c. Webnet – 10.0.2.0/24, this is the internal subnet in which our Ubuntu web server will reside in which will be hiding behind the internal interface of our Firewall.
10. To create these subnets, select “Add Subnet” button.
11. The subnet configurations are shown in the image below:

Basics **IP Addresses** Security Tags Review + create


The virtual network's address space, specified as one or more address prefixes in CIDR notation (e.g. 192.168.1.0/24).

IPv4 address space



10.0.0.0/16 10.0.0.0 - 10.0.255.255 (65536 addresses) 

Add IPv6 address space 


The subnet's address range in CIDR notation (e.g. 192.168.1.0/24). It must be contained by the address space of the virtual network.

+ Add subnet  Remove subnet

<input type="checkbox"/> Subnet name	Subnet address range	NAT gateway
<input type="checkbox"/> Frontend	10.0.0.0/24	-
<input type="checkbox"/> Backend	10.0.1.0/24	-
<input type="checkbox"/> Webnet	10.0.2.0/24	-

 Use of a NAT gateway is recommended for outbound internet access from a subnet. You can deploy a NAT gateway and assign it to a subnet after you create the virtual network. [Learn more](#) 

12. Leave the rest of the settings as default and click on “Next” till you reach “Review + create”.

 Validation passed

Basics **IP Addresses** Security Tags **Review + create**

Basics

Subscription Azure for Students

Resource group myVNET

Name myVNET

Region East US

IP addresses

Address space 10.0.0.0/16

Subnet Frontend (10.0.0.0/24), Backend (10.0.1.0/24), Webnet (10.0.2.0/24)

Tags

None

Security

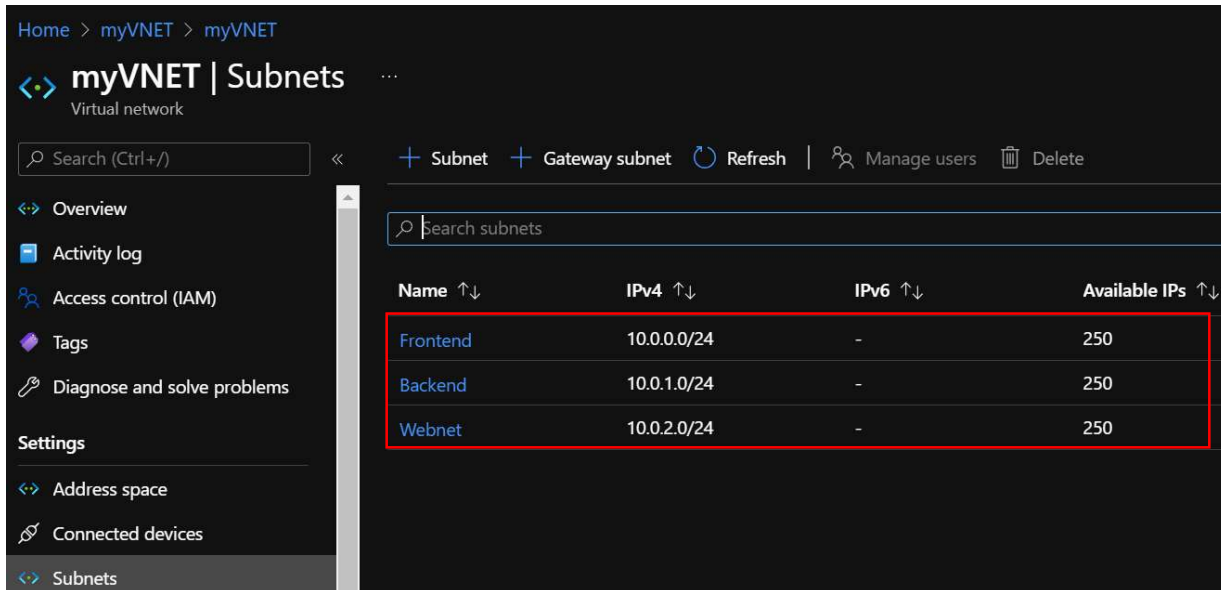
BastionHost Disabled

DDoS protection plan Basic

Firewall Disabled

13. Wait for the validation to pass and click on “Create”.

14. To view all the subnets that had created, go to Resource groups > Select “myVNET” resource group > Select “myVNET” virtual network > Subnets.



Deploying CloudGuard CPFW

1. Go to Azure Marketplace > Search for Check point firewall.
2. Select “CloudGuard Network Security - Firewall & Threat Prevention” and then Select the “CloudGuard Single Gateway” plan.
3. Select the Azure Subscription that you have taken.
4. If you haven’t made a resource group for the Check Point firewall then you can create one by selecting “create new” option and give a name to the resource group (in my case, it will be “CPFw-rg” as shown in the image below.)
5. Give a region for your firewall. I have selected “East US” since my account subscription is of East US region.
6. Give the name for your Firewall VM. I have given CPFW.
7. Select Authentication type as “Password” and give the password for your CP firewall’s admin account. Please remember this password as this the only way we can get access to the checkpoint web interface.
8. Select “Next”.

Basics | Check Point CloudGuard settings | Network settings | Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ ✓

Resource group * ⓘ ✓
[Create new](#)

Instance details

Region * ⓘ ✓

Please follow the Check Point Reference Architecture for Azure.
[Reference Architecture Guide](#)

VM Name * ⓘ ✓

Authentication type *
 Password
 SSH Public Key

Password * ⓘ ✓

Confirm password * ✓

9. Select the latest Check Point CloudGuard version.
10. Select the license type.
11. Give the installation type as Standalone.
12. Give "Allowed GUI clients" as "0.0.0.0/0" which means that anyone can access the Check Point firewall. (Do this setting only in a testing environment, we need this setting right now since we need to make some basic configurations on our firewall from the web interface first.)
13. Leave the rest of the settings as default and Select "Next".

Basics **Check Point CloudGuard settings** Network settings Review + create

Check Point CloudGuard version ⓘ R81.10

License type ⓘ Bring Your Own License

Virtual machine size * ⓘ **1x Standard D3 v2**
4 vcpus, 14 GB memory
[Change size](#)

Installation type ⓘ Standalone

Default shell for the admin user ⓘ /etc/cli.sh

Allowed GUI clients * ⓘ 0.0.0.0/0

Bootstrap script ⓘ Select a file

Allow download from/upload to Check Point ⓘ Yes No

Additional disk space (GB) ⓘ 0

Enable CloudGuard metrics * ⓘ Yes No

14. Select the Virtual Network that we had created, "myVNET".

15. Select the same frontend subnet and backend subnet that we had created before as shown in the image below:

Basics Check Point CloudGuard settings **Network settings** Review + create

Configure virtual networks

Virtual network * ⓘ myVNET
[Create new](#)

Frontend subnet * ⓘ Frontend (10.0.0.0/24)
[Manage subnet configuration](#)

Backend subnet * ⓘ Backend (10.0.1.0/24)
[Manage subnet configuration](#)

16. Now select "Next".

17. Review all the configured settings and wait for the validation to pass. Then Select "Create"

Validation Passed

Basics

Subscription: Azure for Students
 Resource group: CPFW-rg
 Region: East US
 VM Name: CPFW
 Password: *****

Check Point CloudGuard settings

Check Point CloudGuard version: R81.10
 License type: Bring Your Own License
 Virtual machine size: Standard_D3_v2
 Installation type: Standalone
 Default shell for the admin user: /etc/cli.sh
 Allowed GUI clients: 0.0.0.0/0
 Bootstrap script: -
 Allow download from/upload to Check ...: Yes
 Additional disk space (GB): 0
 Enable CloudGuard metrics: Yes

18. To view the Check Point Firewall we deployed, go to Resource groups > “CPFW-rg” resource group.

Home > checkpoint.vsec-20220307154043 > CPFW-rg

Resource group

Search (Ctrl+/) < + Create Edit columns Delete resource group Refresh Export to CSV Open query Assign tags Move Delete

Overview

Activity log

Access control (IAM)

Tags

Resource visualizer

Events

Settings

Deployments

Security

Policies

Properties

Locks

Cost Management

Cost analysis

Cost alerts (preview)

Essentials

Subscription (move): Azure for Students
 Subscriptions ID: 706ea93e-5030-4a94-bec7-c1cf71d39ba
 Location: East US
 Deployments: 3 Succeeded
 Tags (edit): Click here to add tags

Resources Recommendations

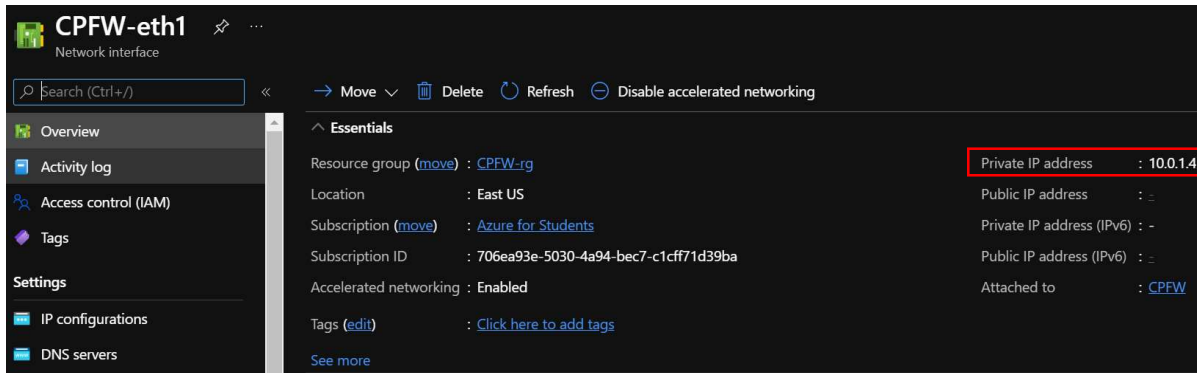
Filter for any field... Type == all Location == all Add filter

Showing 1 to 6 of 6 records. Show hidden types No grouping

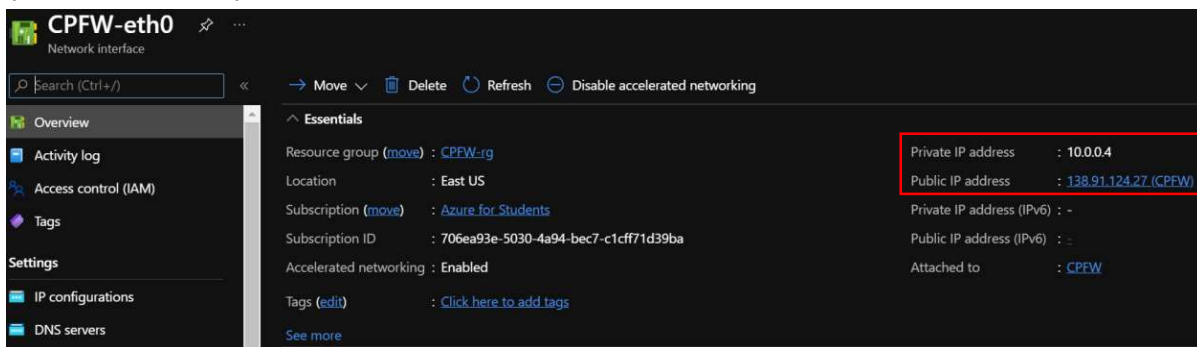
Name	Type	Location
bootdiag7qxlesya4kvhg	Storage account	East US
CPFW	Disk	East US
CPFW	Virtual machine	East US
CPFW	Public IP address	East US
CPFW-eth0	Network interface	East US
CPFW-eth1	Network interface	East US

19. As we can see, we have the CPFW VM, Internal interface of Check Point: “CPFW-eth1” and the External interface of Check Point: “CPFW-eth0”.

20. As you can see, our internal interface has only a Private IP (10.0.1.4) and no Public IP.



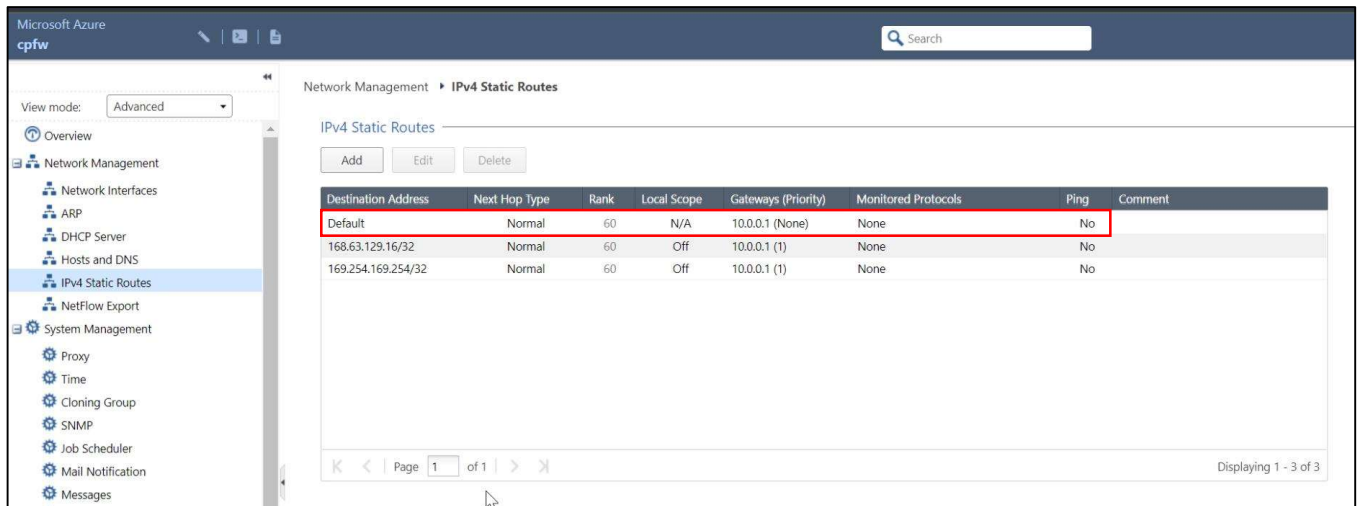
21. However, our External interface on the other hand has Private IP (10.0.0.4) and Public IP (138.91.124.27) as well.



Now we will be configuring our firewall via the Gaia Web interface of the firewall.

Configuring the initial setup of CPFW from web interface

1. Browse to the Public IP of the CPFW (138.91.124.27) via the browser.
2. Enter the admin credentials that we had configured for Firewall and login.
3. Download the Smart console from the CP web interface.
4. While that downloads, go to IPv4 Static Routes and verify if there is a default route to the internet so that our CPFW and Internal networks can reach the internet.
5. As you can see from the image below, by default, Azure has configured a default route via 10.0.0.1 which is the IP of the Azure router in 10.0.0.0/24 Frontend sub-network.



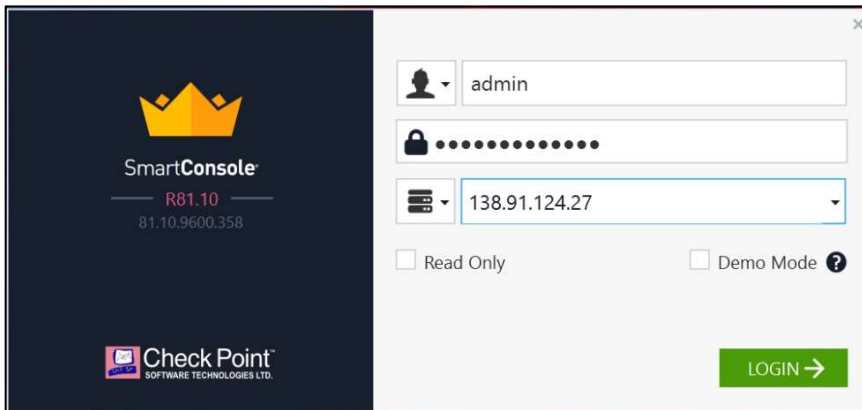
6. Side note: By default, whenever we create a virtual network or even a sub-network in Azure, the 1st IP of the network is reserved for the Azure router, 2nd and 3rd IP is reserved for the Azure DNS services, the last IP (x.x.x.255) is reserved for the broadcast address. So, if you notice the IP addresses of the interfaces of the CPFW, they always start with “x.x.x.4” by default.
7. So now, our internal webserver can reach the internet or in other words, our firewall is able to route outbound traffic.
8. But there is something missing in this Static Route configuration. If External users want to reach our Ubuntu Web Server’s IP, then traffic will first flow to the Public IP of our CPFW -> Azure router in 10.0.0.0/24 Frontend subnet -> Private IP of our CPFW-external interface -> Private IP of our CPFW-external interface -> Azure router in 10.0.1.0/24 Backend subnet -> Webserver.
9. So, we need to configure a static route to our Webserver’s subnet 10.0.2.0/24 via 10.0.1.1 (Azure router in 10.0.1.0/24 subnet).

Destination Address	Next Hop Type	Rank	Local Scope	Gateways (Priority)	Monitored Protocols	Ping	Comment
Default	Normal	60	N/A	10.0.0.1 (None)	None	No	
10.0.2.0/24	Normal	60	Off	10.0.1.1 (None)	None	No	
168.63.129.16/32	Normal	60	Off	10.0.0.1 (1)	None	No	
169.254.169.254/32	Normal	60	Off	10.0.0.1 (1)	None	No	

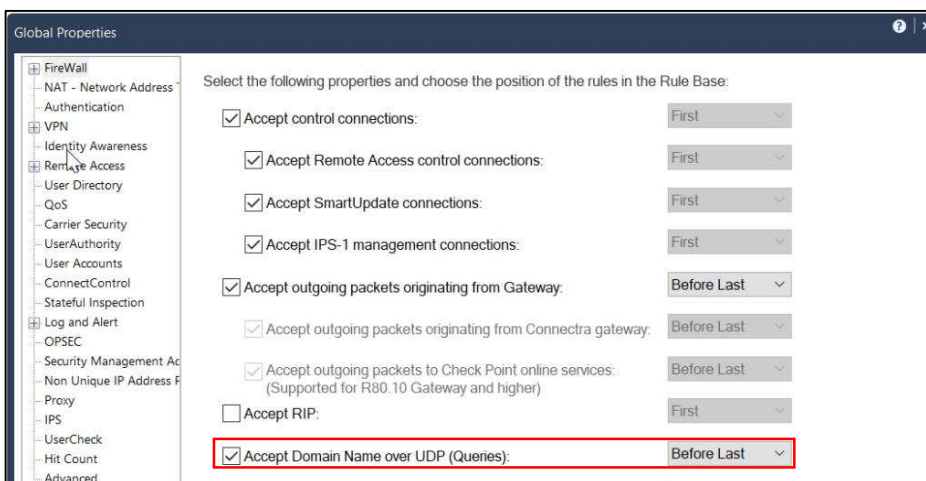
10. Save these changes.

Configuring the policies on CPFW

1. Login to your CPFW Public IP via the smart console using the same admin credentials that we used previously.



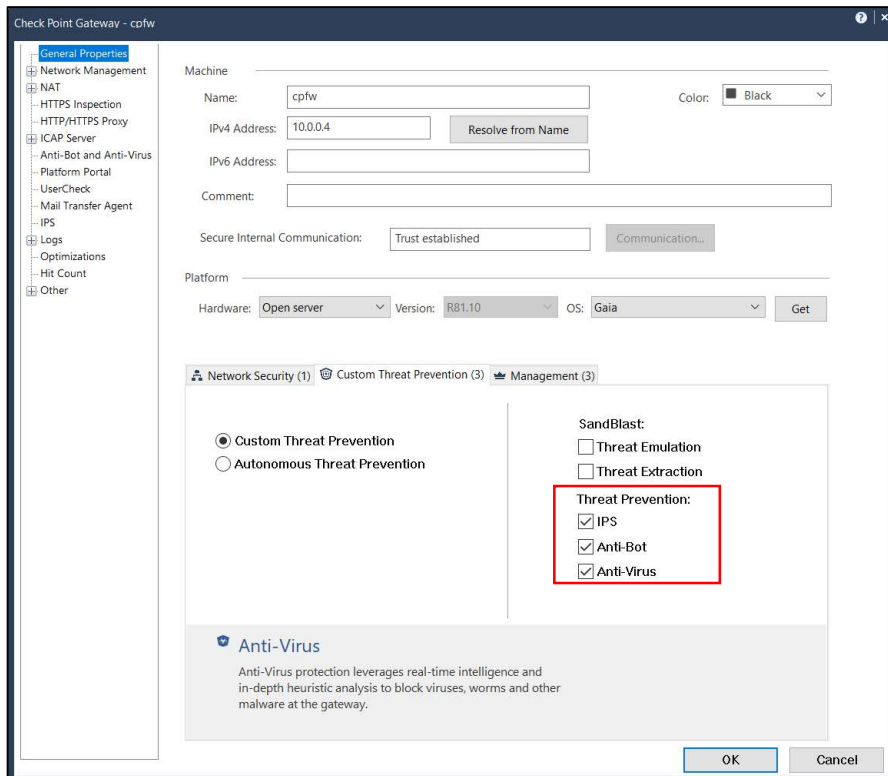
- 2.
3. We will need to turn on the implicit rule for allowing DNS queries over UDP by going to Menu > Global Properties > Firewall > Enable “Accept Domain Name over UDP” and select “Before Last” so that the DNS query will be allowed before the “Clean up rule” (Last rule) is processed.



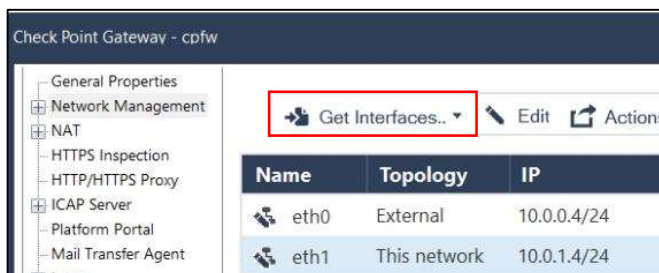
4. Now go to the “CPFw” object in “Gateway and Servers” tab on the left pane.



5. Make sure that all the necessary blades such as: Firewall, IPS, Anti-Bot, Anti-Virus blades are enabled by going to General Properties > Check the blades in Network Security and Custom Threat Prevention section.

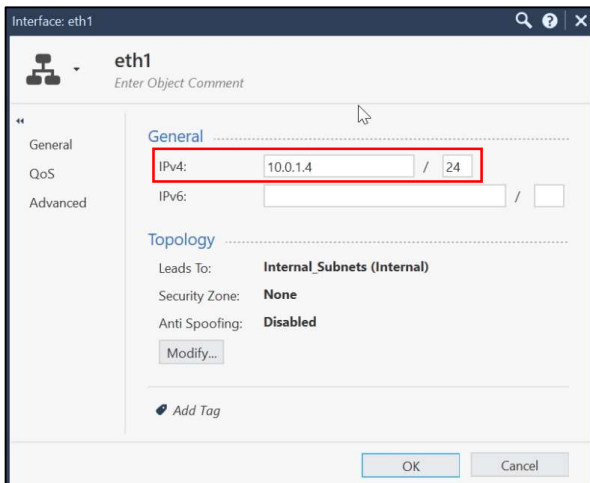
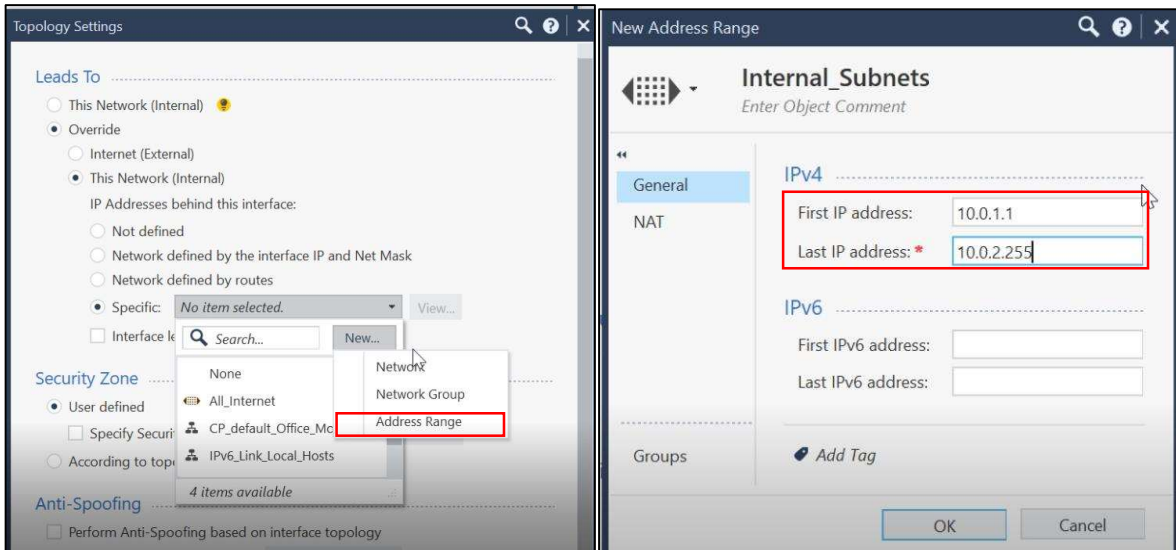


6. Go to Network Management > Get Interfaces with Topology.



7. Now, we need to our CPFW to know all the sub-networks that hide behind the "10.0.1.0/24" subnet.

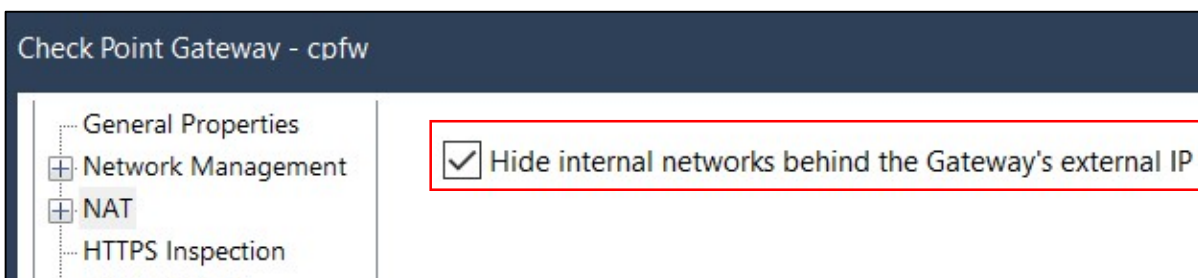
8. Select "eth1" interface > General > Topology > Modify > Leads To > Override > This Network > Specific > Select "New" > Address range. Give the range as (10.0.1.1 – 10.0.2.255)



9. Select “OK” for all the settings.

10. If any device in our internal networks wants to reach the internet, they will need to do so by hiding behind the external interface of our CPFW. So, we will need to enable “Hide NAT”.

11. Go to NAT > Select “Hide internal networks behind the Gateway’s external IP.”



12. Now if an external user tries to access my CPFW Public IP on port 80 (http), the request should get Natted to my Webserver’s Private IP address.

13. We are now going to configure a NAT rule for this.

14. Go to Security Policies > Access Control > NAT > Select “Add a rule above” option.



15. Configure the NAT rules as shown below:

No.	Name	Original Source	Original Destinati...	Original Services	Translated Source	Translated Destin...	Translated Services	Install On	Comments
1	WebNAT	* Any	Private-CPFW-IP	http	= Original	Webserver	= Original	* Policy Targets	
2	sshNAT	My PC	Private-CPFW-IP	Remote_Desk...	= Original	Webserver	ssh	* Policy Targets	
Automatic Generated Rules : Machine Static NAT (No Rules)									
Automatic Generated Rules : Machine Hide NAT (No Rules)									
Automatic Generated Rules : Address Range Static NAT (No Rules)									
Automatic Generated Rules : Network Static NAT (No Rules)									
Automatic Generated Rules : Address Range Hide NAT (No Rules)									
▼ Automatic Generated Rules : Network Hide NAT (3-4)									
3	Automatic Rule: CP_default_Office_Mo de_addresses_pool	CP_default_Off...	CP_default_Offici	* Any	= Original	= Original	= Original	* All	
4	Automatic Rule: CP_default_Office_Mo de_addresses_pool	CP_default_Off...	* Any	* Any	CP_default_Offici	= Original	= Original	* All	
Manual Lower Rules (No Rules)									

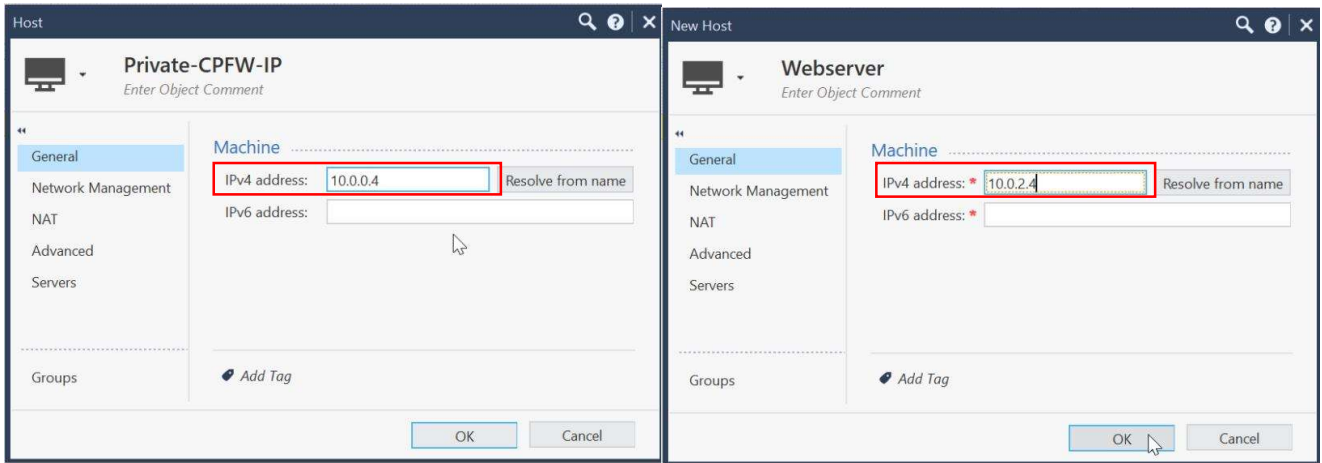
16. Why are we Natting the Private IP of the Ext-interface of CPFW to Private IP of Webserver? Shouldn't we NAT the Public IP of CPFW-ext to Private IP of Webserver?

The reason is because Public IP of CPFW-external interface is already being Natted to Private IP of CPFW-external interface by the Azure router in the "10.0.0.0/24" Frontend subnet)

17. Since my Webserver doesn't have a Public IP, I cannot access it via SSH from Putty. If I do a normal "Destination NAT" where I only NAT the Public IP of my CPFW to the Private IP of my Webserver over SSH, then I will be able to access my Webserver which is hidden behind the CPFW, but I will lose SSH access to my CPFW console which is also on port 22.

18. So, the solution here is to use "Port Address Translation" or PAT where I can translate the request going to the Public IP of RDP port (3389) on my CPFW to Private IP of my Webserver on SSH port (22). This way I have access to both of the devices.

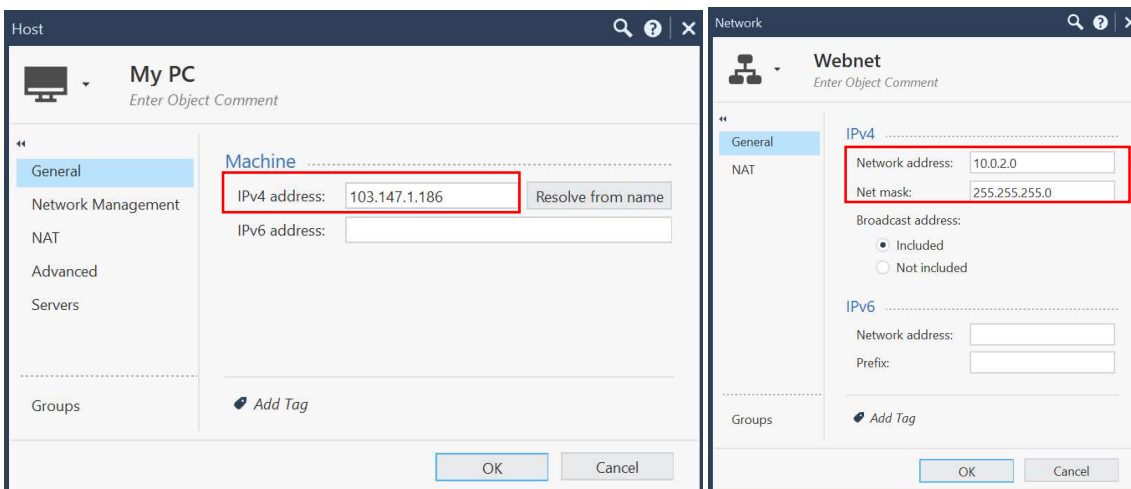
19. The network objects present in the NAT rules configured above is shown in the images below:



20. Now go to Access Control > Policy. Configure the access policies in the following way:

No.	Name	Source	Destination	VPN	Services & Applications	Action	Track
1	Allow incoming HTTP traffic	* Any	Private-CPFW-IP	* Any	http https	Accept	Log
2	Allow SSH from my IP	My PC	* Any	* Any	Remote_Desktop_Pr... ssh	Accept	Log
3	Internet rule: Allow outgoing HTTP traffic	Webnet	* Any	* Any	icmp-requests https http	Accept	Log
4	Cleanup rule	* Any	* Any	* Any	* Any	Drop	Log

21. The network objects used in the image above is shown below:



22. We will configure a Threat Prevention policy to make our Internal networks more secure.

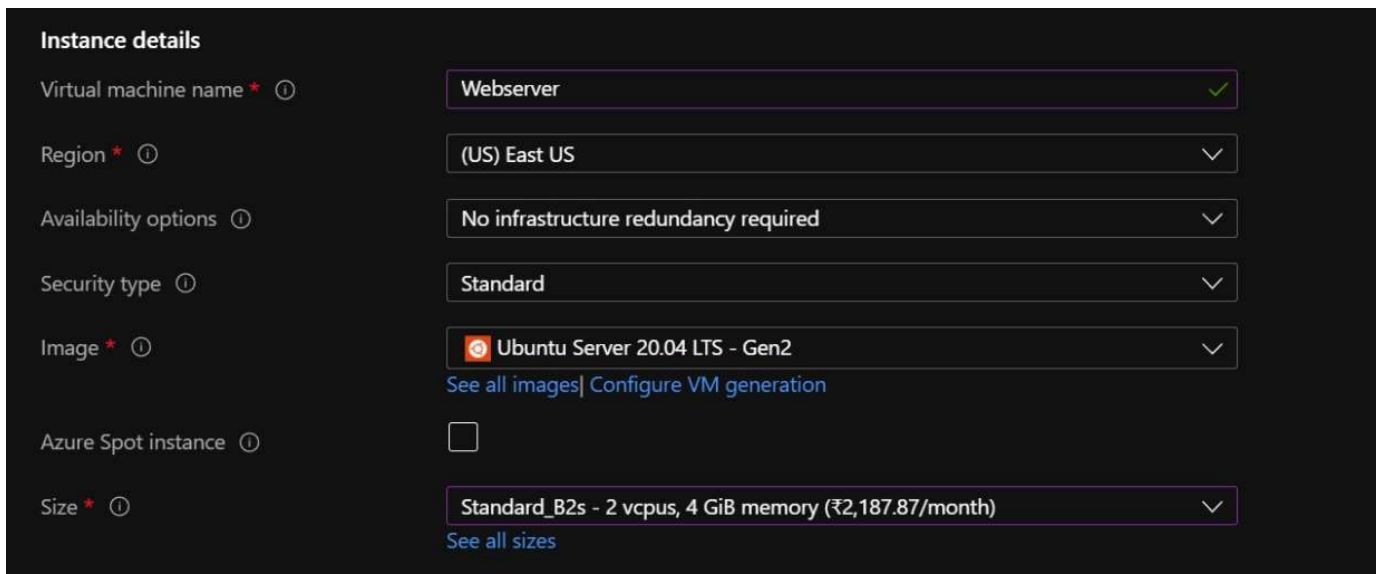
23. Go to Threat Prevention > Custom Policy. Configure the policy as per the image below:

No.	Name	Protected Scope	Protection/Site/File/Blade	Action	Track	Install On
1	Threat Prevention rule	* Any	N/A	Strict	Log Packet Capture Forensics	* Policy Targets

24. Now Publish and Install the Policy.

Deploying Ubuntu Web Server

1. Go to Azure Marketplace > Search for Ubuntu Server 20.04 LTS and select the Ubuntu web server.
2. Select the resource group for your Ubuntu Web Server. I have selected “myVNET” resource group since I don’t have any Virtual Machines occupying that resource group. You can also choose to create a separate resource group for the Ubuntu web server if you wish to do so.
3. Give the VM name as “Webserver”.
4. Give region as “East US”
5. Select the size as “Standard B2s”. I have selected this option since it is the most economical option available.
6. Select the Authentication type as “Password”.
7. Enter the username and password that you will use to login to the web server.
8. Leave the rest of the settings as default and click on “Next” till you reach “Networking” tab.



The screenshot shows the 'Instance details' section of the Azure portal. It contains the following configuration options:

Field	Value
Virtual machine name *	Webserver ✓
Region *	(US) East US
Availability options	No infrastructure redundancy required
Security type	Standard
Image *	Ubuntu Server 20.04 LTS - Gen2 See all images Configure VM generation
Azure Spot instance	<input type="checkbox"/>
Size *	Standard_B2s - 2 vcpus, 4 GiB memory (₹2,187.87/month) See all sizes

Administrator account

Authentication type ⓘ SSH public key Password

Username * ⓘ ✓

Password * ⓘ ✓

Confirm password * ⓘ ✓

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ None Allow selected ports

Select inbound ports * ✓

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

9. Now, select the virtual network that we had created in the beginning which is myVNET.
10. Select the subnet that we had created specifically for our webserver which is "Webnet".
11. Leave rest of the settings as default including the default Public IP that is given to us by Azure as we will be needing this IP to make some initial configurations to the server and then we will disable the Public IP of the server so that we can hide our web server behind the Public IP of the Firewall.
12. Now click on "Next" till you reach the "Review + create" tab.

Basics Disks **Networking** Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
[Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * ⓘ myVNET ▼
[Create new](#)

Subnet * ⓘ Webnet (10.0.2.0/24) ▼
[Manage subnet configuration](#)

Public IP ⓘ (new) Webserver-ip ▼
[Create new](#)

NIC network security group ⓘ None
 Basic
 Advanced

Public inbound ports * ⓘ None
 Allow selected ports

Select inbound ports * SSH (22) ▼

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

13. Now wait for the validation to pass and select “create”.

✓
Validation passed

Subnet	Webnet (10.0.2.0/24)
Public IP	(new) Webserver-ip
Accelerated networking	Off
Place this virtual machine behind an existing load balancing solution?	No
Delete public IP and NIC when VM is deleted	Disabled
Management	
Azure Security Center	Basic (free)
Boot diagnostics	Off
Enable OS guest diagnostics	Off
System assigned managed identity	Off
Login with Azure AD	Off
Auto-shutdown	Off
Backup	Disabled
Enable hotpatch	Off
Patch orchestration options	Image Default

You will be able to find the Ubuntu VM in the “myVNET” resource group.

Configuring Ubuntu Web Server

1. Go to the Public IP of the Ubuntu VM which is in the “myVNET” resource group.

Name ↑↓	Type ↑↓	Location ↑↓
myVNET	Virtual network	East US
Webserver	Virtual machine	East US
Webserver-ip	Public IP address	East US
Webserver-nsg	Network security group	East US
webserver96	Network interface	East US
Webserver_disk1_bf3c08297b734ca6974c27461df4aaf2	Disk	East US

2. Copy the Public IP address of the Ubuntu VM.

The screenshot shows the Azure portal interface for the 'Webserver-ip' public IP address. The 'Essentials' section displays the following information:

Resource group (move)	: myVNET	SKU	: Basic
Location	: East US	Tier	: Regional
Subscription (move)	: Azure for Students	IP address	: 20.121.8.205
Subscription ID	: 706ea93e-5030-4a94-bec7-c1cf71d39ba	DNS name	: -
Tags (edit)	: Click here to add tags	Associated to	: webserver968

3. Open Putty.exe client and login using ssh to the Public IP of Ubuntu with username and password that we had configured before deployment and then select “Open”.

The screenshot shows the PuTTY Configuration dialog box. The 'Host Name (or IP address)' field is filled with 'markashwin13@20.121.8.205' and the 'Port' field is filled with '22'. The 'SSH' connection type is selected. The 'Open' button is highlighted.

4. As you can see, we have successfully logged in.

```
markashwin13@Webserver: ~
System information as of Mon Mar 7 10:56:45 UTC 2022

System load: 0.0          Processes:              113
Usage of /:  4.7% of 28.90GB  Users logged in:      0
Memory usage: 6%          IPv4 address for eth0: 10.0.2.4
Swap usage:  0%

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

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.

markashwin13@Webserver:~$
```

5. Now we are going to run some commands to setup this web server.

```
# sudo apt update
```

```
# sudo apt upgrade
```

```
# sudo apt-get install apache2
```

```
# sudo systemctl start apache2
```

```
# sudo systemctl enable apache2 [So that the web service starts at boot by default.]
```

These commands are going to update our Ubuntu system, install and start the Apache2 web service on it.

6. To check if the Apache web service is running run the following command:

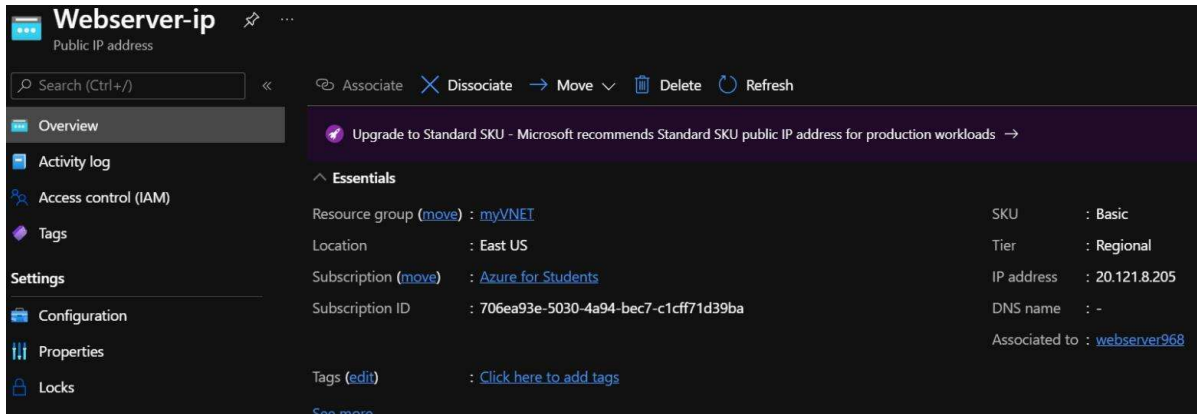
```
# sudo systemctl status apache2
```

```
markashwin13@Webserver:~$ systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Mon 2022-03-07 10:59:23 UTC; 19s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 3384 (apache2)
     Tasks: 55 (limit: 4704)
    Memory: 9.0M
   CGroup: /system.slice/apache2.service
           └─3384 /usr/sbin/apache2 -k start
             └─3386 /usr/sbin/apache2 -k start
               └─3387 /usr/sbin/apache2 -k start

Mar 07 10:59:23 Webserver systemd[1]: Starting The Apache HTTP Server...
Mar 07 10:59:23 Webserver systemd[1]: Started The Apache HTTP Server.
markashwin13@Webserver:~$
```

7. If it says "Active", then you are good to go.

- Now we are going to disable the Public IP of the Ubuntu Web server since we don't need it anymore. (We will be accessing the webserver via our CPFW's Public IP)
- Go to "myVNET" resource group > "Webserver-ip" which is the public IP of the ubuntu web server.



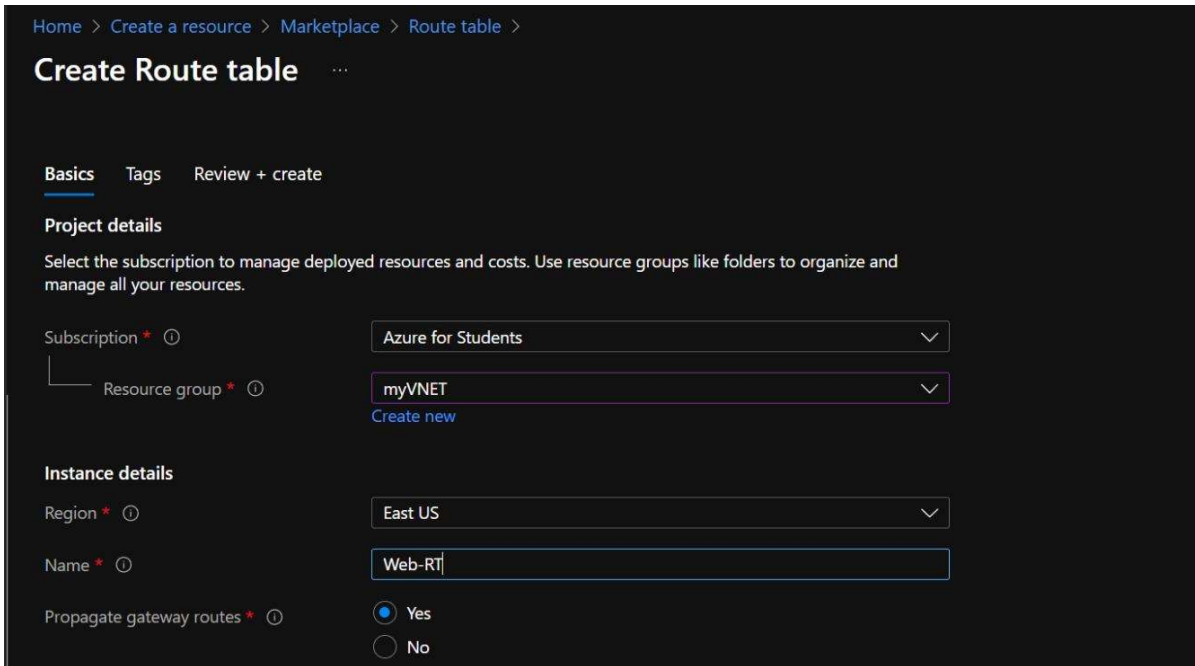
- Select "Dissociate" which will disable the public IP.

We are only going to need the Private IP of the webserver for communication.

Configuring routing on the "Webnet" subnet using IP route tables

- Now by default, whenever we create a new subnet, Azure has "System routes" in place which connects all the various subnets within a virtual network. So therefore, I don't need to configure routing between the various subnets that I had created since Azure's System routes already handle that for me. But, since we need traffic from our Webserver's to traverse via our Check Point FW so that we can monitor traffic and protect the server, we need to override the System routes in place with our own "User Defined Routes" (UDR).
- UDR's help us gain more control over the routing of traffic of a specific subnet that it is associated with. The way we can define user defined routes is with the help of IP Route tables in Azure. If you wish to know more about System routes and User defined routes in Azure, check out this reference link: <https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview>
- To add an IP route table for our Webserver, go to Azure marketplace and search for "Route tables".
- Select "Route tables" and click on "Create".
- Now select a resource group for this Route table. I have selected "myVNET" resource group that I had created earlier (If you don't have a resource group for this you can simply click on "Create new" near the "Resource group" field)

6. Give the name of the Route table as “Web-RT”.
7. Leave the rest of the settings as default and select “Next” till you reach “Review + create”



Home > Create a resource > Marketplace > Route table >

Create Route table

Basics Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure for Students

Resource group * ⓘ myVNET
[Create new](#)

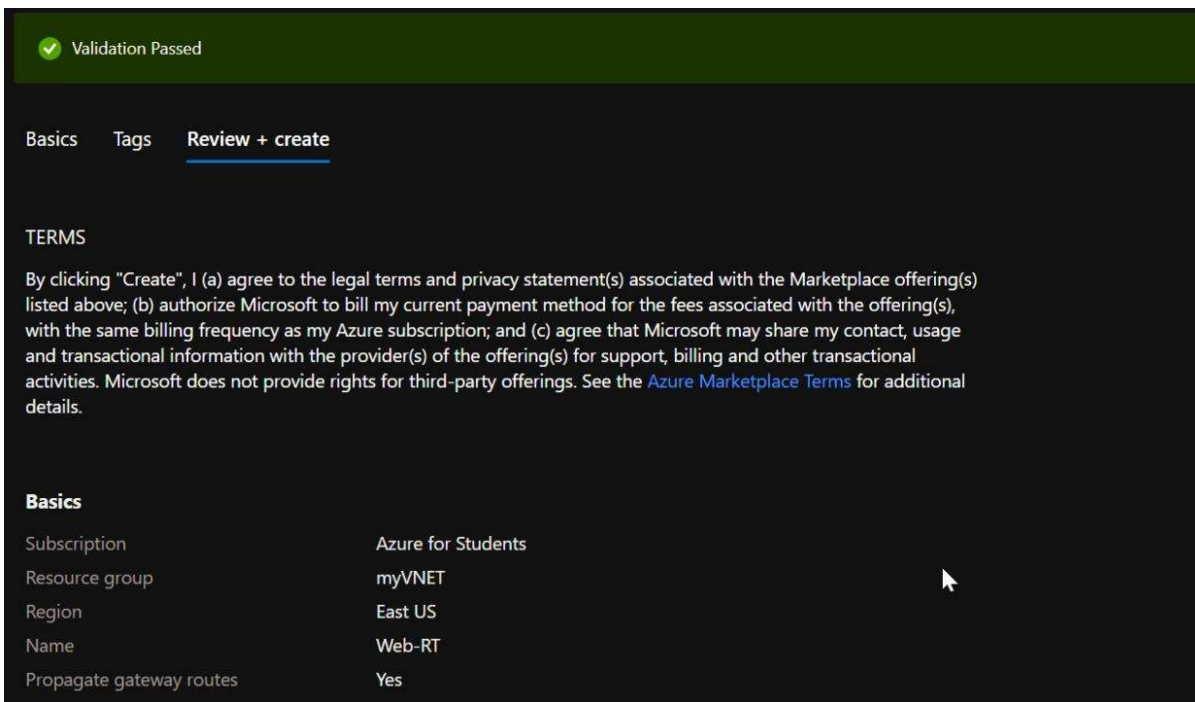
Instance details

Region * ⓘ East US

Name * ⓘ Web-RT

Propagate gateway routes * ⓘ Yes No

8. Wait for the validation to pass and select “Create”.



Validation Passed

Basics Tags **Review + create**

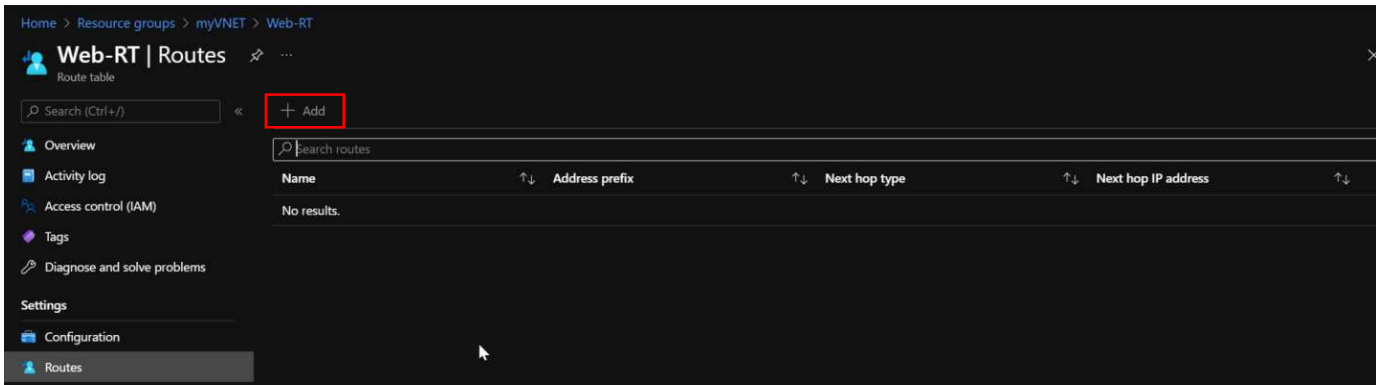
TERMS

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. See the [Azure Marketplace Terms](#) for additional details.

Basics

Subscription	Azure for Students
Resource group	myVNET
Region	East US
Name	Web-RT
Propagate gateway routes	Yes

9. Now, go to the “Web-RT” route table that we had configured in the “myVNET” resource group and click on “Routes”.



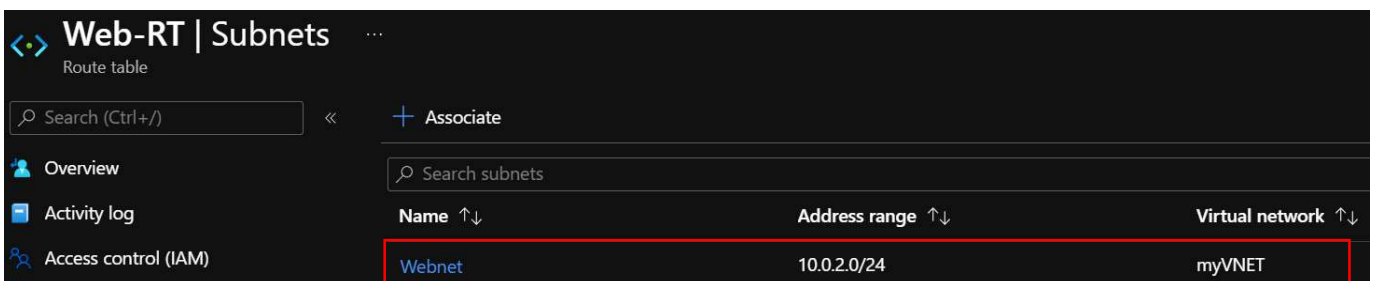
10. We are going to add 2 routes: -

- a. 1 route for communication within the “10.0.0.0/16” Virtual network
- b. 1 route for communication with the internet (0.0.0.0/0) via our CPFW internal interface gateway.



11. We need to attach this Route table to our “Webnet” subnet.

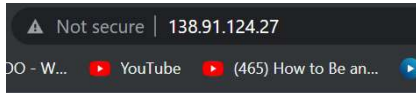
12. To do this, go to “Subnets” on the left pane > Click on “Associate”. Add the “Webnet” subnet and select “OK”.



13. Now our Webserver will be able to communicate with all the devices in the “myVNET” Virtual Network and the internet.

Testing our Ubuntu Web Server Connectivity

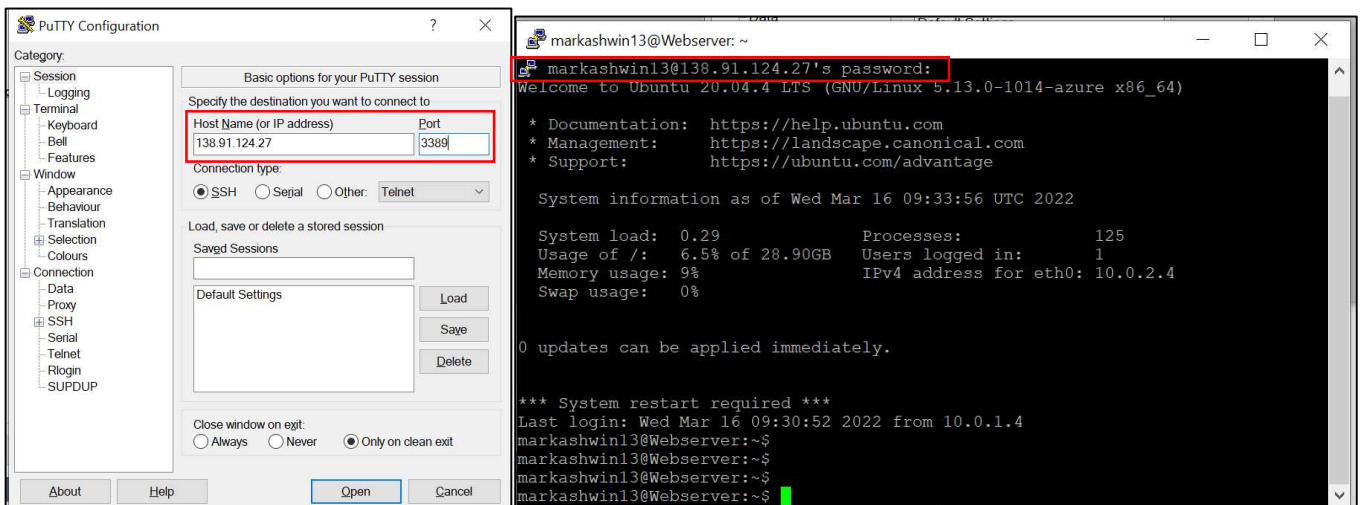
1. Now, browse to the Public IP of our CPFW using http service.
2. The link in my case is: `http:// 138.91.124.27`



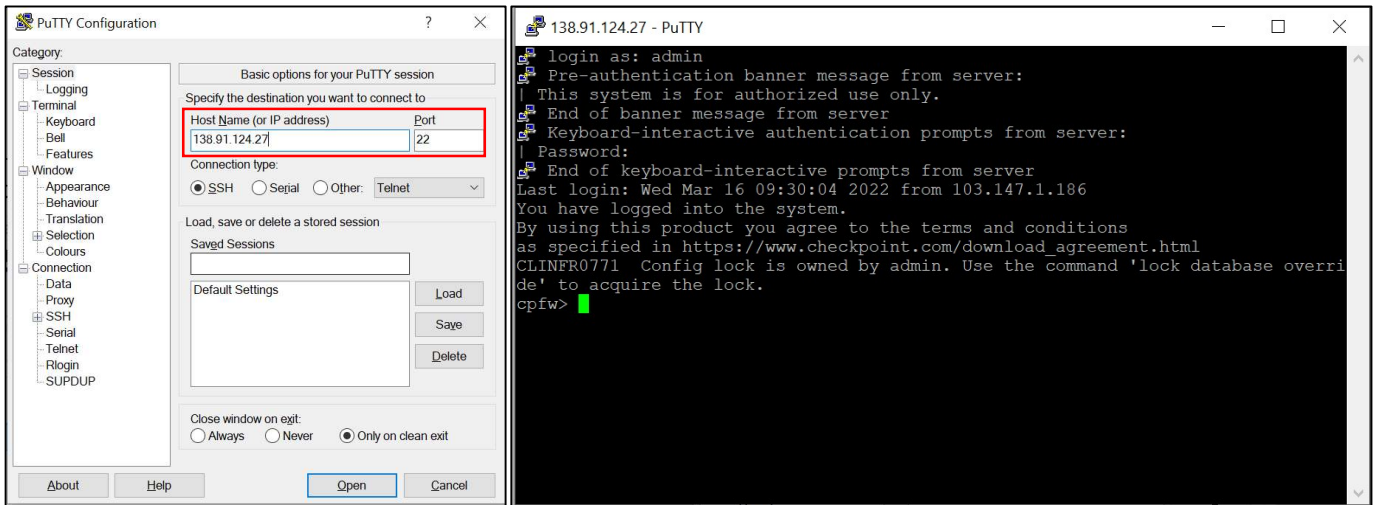
3. As you can see, our Webserver is working perfectly.



4. Let's try to access our Webserver console via SSH on port 3389 with the help of Putty



5. Let's try to access our CPFW console via SSH on port 22 with the help of Putty



6. As you can see, we are able to get console access via SSH on the respective devices properly. This means, our PAT and Destination NAT policies are working perfectly.

7. Let us take a look at the logs to verify this from our SmartConsole.

The image shows a screenshot of the SmartConsole logs. The logs display traffic from 'My PC (103.147.1.186)' to 'cpfw (10.0.0.4)'. The logs are filtered by 'src:"My PC"'. The table below shows the log entries:

Time	Origin	Source	Source User...	Destination	Service	Ac...	Access Rule...	Policy...	Description
Today, 3.03.42 PM	cpfw	My PC (103.147.1.186)		cpfw (10.0.0.4)	Remote_Desktop_Pro...	2	Allow SSH from...	Standard	Remote_Desktop_Protocol Traffic Accepted from 103.147.1.186 to 10.0.0.4
Today, 3.02.51 PM	cpfw	My PC (103.147.1.186)		cpfw (10.0.0.4)	https (TCP/443)	1	Allow incoming...	Standard	https Traffic Accepted from 103.147.1.186 to 10.0.0.4
Today, 2.54.18 PM	cpfw	My PC (103.147.1.186)		cpfw (10.0.0.4)	https (TCP/443)	1	Allow incoming...	Standard	https Traffic Accepted from 103.147.1.186 to 10.0.0.4
Today, 2.54.16 PM	cpfw	My PC (103.147.1.186)		cpfw (10.0.0.4)	https (TCP/443)	1	Allow incoming...	Standard	https Traffic Accepted from 103.147.1.186 to 10.0.0.4
Today, 2.54.05 PM	cpfw	My PC (103.147.1.186)		cpfw (10.0.0.4)	https (TCP/443)	1	Allow incoming...	Standard	https Traffic Accepted from 103.147.1.186 to 10.0.0.4

8. SSH to our Webserver on RDP port 3389:

The screenshot shows the 'Log Details' window for an 'Accept' action. The traffic is from 'My PC (103.147.1.186)' to 'Webserver (10.0.2.4)' on port 22. The blade is 'Firewall' and the service is 'Remote_Desktop_Protocol (TCP/3389)'. The action is 'Accept' under the 'Standard' policy.

Section	Field	Value
Details	Origin	cpfw
	Time	Today, 3.03.42 PM
	Blade	Firewall
	Type	Connection
Traffic	Source	My PC (103.147.1.186)
	Source Port	58386
	Destination	cpfw (10.0.0.4)
	Service	Remote_Desktop_Protocol (TCP/3389)
Policy	Action	Accept
	Policy Name	Standard
	Policy Date	Today, 2.58.38 PM
	Access Rule Name	Allow SSH from my IP
Matched Rules	Xlate (NAT) Destination IP	Webserver (10.0.2.4)
	Xlate (NAT) Source Port	0
	Xlate (NAT) Destination P...	22
	NAT Rule Number	2
Actions	Report Log	Report Log to Check Point
	Id	2693254b-7448-e7fe-6231-ae600000...
	Logid	0
	Description	Remote_Desktop_Protocol Traffic Acce...

Testing the Threat Prevention blade for our Ubuntu Web Server

1. I tried to launch a basic Reconnaissance attack on the Ubuntu Web Server hiding behind the CPFW with the help of nmapAutomator tool from my Kali Linux VM.
2. I used the following command on my Kali VM for conducting the Recon attack:

```
# ./nmapAutomator.sh -H 138.91.124.27 80 -t Recon
```

```
(root@kali) - [~/home/kali/thm/nmapAutomator]
# ./nmapAutomator.sh -H 138.91.124.27 80 -t Recon
Running a Recon scan on 138.91.124.27
No ping detected.. Will not use ping scans!
Host is likely running Unknown OS!
-----Starting Port Scan-----
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
264/tcp   open  bgmp
443/tcp   open  https
3389/tcp  open  ms-wbt-server
```

- Once the Attack started, I went to the SmartConsole on my PC and looked for the Threat Prevention blade logs.
- As you can see, our Check Point that was deployed in Azure was successfully able to defend against many critical vulnerabilities with the help the Threat Prevention blades available.

Time	B.	A.	T.	Seve...	Con...	Perf...	Source	Destination	Attack Name	Protection Name	Resource
Today, 5.45.22 PM							My PC (103.147....	cpfw (10.0.0.4)	SSL Enforcement Violation	Secure Sockets Layer (SSL) Version 2.0	
Today, 5.45.21 PM							My PC (103.147....	cpfw (10.0.0.4)	Scanner Enforcement Violation	Nmap Scripting Engine Scanner Ov...	http://138.91
Today, 5.45.21 PM							My PC (103.147....	cpfw (10.0.0.4)	Web Server Enforcement Violation	Web Server Exposed Git Repository I...	http://138.91
Today, 5.45.20 PM							My PC (103.147....	cpfw (10.0.0.4)	SSL Enforcement Violation	Secure Sockets Layer (SSL) Version 2.0	
Today, 5.45.20 PM							My PC (103.147....	cpfw (10.0.0.4)	Scanner Enforcement Violation	Nmap Scripting Engine Scanner Ov...	http://138.91

- A detailed view of one of the logs:

Log Details

Prevent
Prevented web server exposed git repository information disclosure originating from 103.147.1.186 against 10.0.0.4

Details | Matched Rules

Time: Today, 5.45.21 PM

Blade: **IPS**

Product Family: Threat

Type: Log

Policy

Action: Prevent

Access Rule Name: Allow incoming HTTP traffic

Threat Prevention Rule ID: F851ACAD-CAA9-4753-AF6B-3359627BDD26

Threat Prevention Policy: Standard

Policy Date: Today, 2.58.38 PM

Threat Prevention Policy ...: Today, 4.03.42 PM

Policy Name: Standard

Policy Management: cpfw

Threat Prevention Rule N...: Threat Prevention rule

Threat Profile: Strict

Add Exception: Add Exception...

Protection Details

Severity: **Critical**

Confidence Level: **High**

Attack Name: Web Server Enforcement Violation

Attack Information: Web Server Exposed Git Repository Information Disclosure

Performance Impact: **Medium**

Protection Name: Web Server Exposed Git Repository Information Disclosure

Protection Type: IPS

Traffic

Source: My PC (103.147.1.186)

Service: http (TCP/80)

Source Port: 60670

Bytes (sent/received): 19.8 KB \ 143.3 KB

Interface: eth0

Destination: cpfw (10.0.0.4)

Suppressed Logs: 87

Packet Capture Unique Id: time1647432921.id4a3d9240.blade02

Packet Captures: src-103.147.1.186.cap

Packet Capture: Packet Capture

Threat Wiki: Go to Threat Wiki

Advanced Forensics Details

Method: GET

Actions

More

Id: 2693254b-7448-e7fe-6231-d4d900000002

Id Generated By Indexer: false

First: false

Sequencenum: 2

Description URL: GIT_EXPOSED_help.html

Log ID: 2

Reject ID Kid: 6231d4d9-1-4b259326-fee74874

Ser Agent Kid: Other: Mozilla/5.0 (compatible; Nmap Scripting Engine; ht...

Last Update Time: 2022-03-16T12:16:22Z

Db Tag: {6207AD16-F287-6F49-B63E-6ADABC0F099}

Log Server Origin: cpfw (10.0.0.4)

Tags: Vendor_Git, Product_Web_Servers, Threat_Year_2015, Threat_Prevalence_True, Protection_Type_Vulnerability, Product_Prevalence_Common, Tuning_Non_Configurable, Protocol_HTTP, Direction_SERVER

