

Deploying CloudGuard with Azure Application Gateway WAF

Chris Beckett
chrisbe@checkpoint.com
7th November 2018

Contents

Deploying CloudGuard with Azure Application Gateway WAF	1
Introduction	2
Deployment Steps.....	2
Adding a backend pool.....	5
Upgrading the VMSS	6
Verifying backend health.....	7
Configuring Backend HTTP Settings.....	7
Create new TCP service in SmartConsole	8
Configuring Access Control and NAT Rules.....	8
Testing the connection	9
Troubleshooting	9
Revision History	9

Introduction

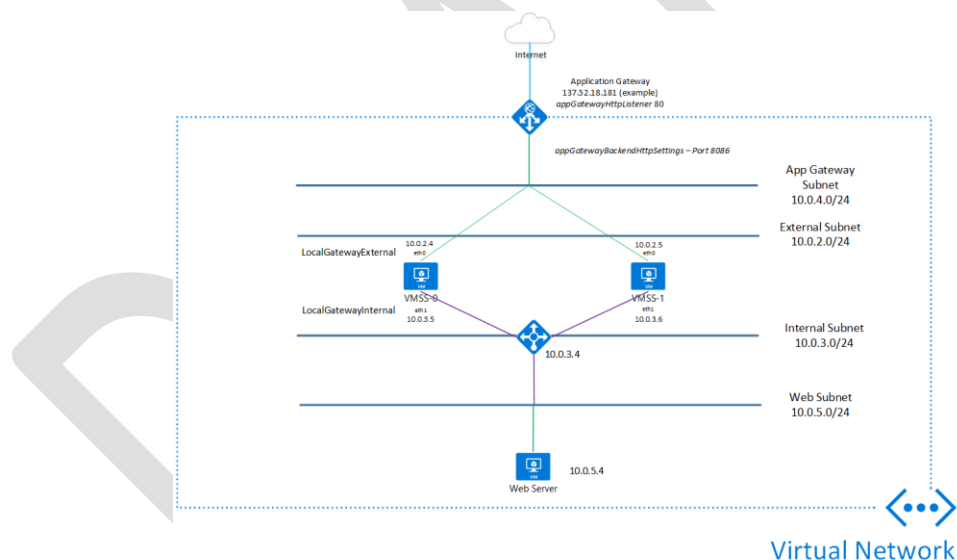
Check Point CloudGuard IaaS is deployed from the Azure Marketplace using a supplied ARM template that scripts the installation and configuration of Azure load balancers and Virtual Machine Scale Sets (VMSS).

As CloudGuard is not strictly a Web Application Firewall (WAF) but has WAF like capabilities, sometimes a WAF is required to complete any missing functionality and/or for design compliance, depending on the deployment scenario.

In order to add a WAF to a CloudGuard secured environment, the native Azure WAF can be deployed and used by deploying an Application Gateway instead of the external load balancer which is created by the Marketplace ARM template. This provides load balancing facilities as well as WAF functionality.

A diagram is shown below of a sample environment protected by the Azure Application Gateway, illustrating the change of external perimeter device from Azure load balancer to Azure Application Gateway.

The benefits of this solution include WAF capabilities, auto scaling (provided by the VMSS) and connection draining during scale in operations.



Deployment Steps

As a pre-requisite, it is assumed that the Check Point Management server is present and has the ability to manage the Azure environment. The CloudGuard scale set Marketplace solution should also be deployed as per the diagram below. To keep the deployment simple, it is recommended to choose the “Internal Only” load balancer deployment.

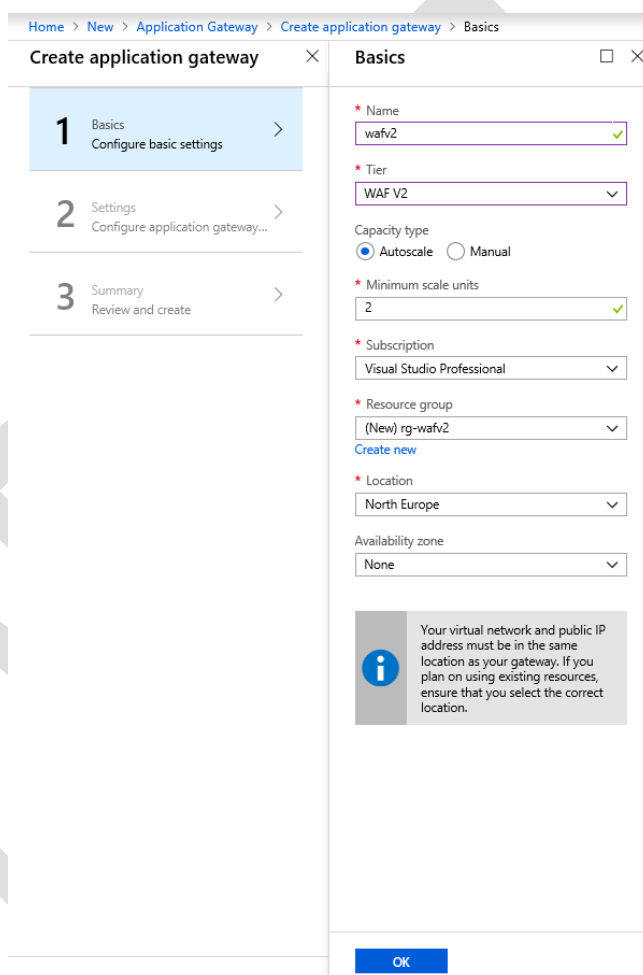
The purpose of this is to ensure an external load balancer is not created that is not used and may incur further charges.

For testing purposes, place a web server into the web subnet as shown in the diagram below, in this example we will use Nginx running on Linux.

Once the VMSS has been created with the internal load balancer, the next step is to create the Azure Application Gateway.

Creating the Azure Application Gateway

The Application Gateway is created from the Azure Portal by clicking the top left “Create a resource” button. In step 1 of the wizard, give the Application Gateway a name, choose the appropriate tier (WAF), instance count (minimum of 2 is recommended), SKU size (Medium is recommended for testing purposes), subscription, resource group and location.



In step 2, choose the Virtual Network and subnet for the Application Gateway to reside in (**remember that the subnet that the Application Gateway is created in must be empty**. A small subnet is recommended for this, a /26 or /27 for example to prevent IP address wastage).

Leave the **Frontend IP configuration** as the default settings and select the required options for **Additional Settings – HTTP2 support, Web Application Firewall** (set to

enabled) and Firewall Mode (set to Prevention).

Create application gateway × **Settings** □ ×

1 Basics
Configure basic settings ✓

2 Settings
Configure application gateway... >

3 Summary
Review and create >

Subnet configuration

* Virtual network ⓘ
cplab-vnet >

* Subnet ⓘ
WAFSubnet (10.0.6.0/24) v

Frontend IP configuration

* IP address type
 Public Private

* Public IP address ⓘ
 Create new Use existing

wafv2-ip

^ Configure public IP address

SKU ⓘ
Basic

* Idle timeout (minutes) ⓘ
4

DNS name label ⓘ
northeurope.cloudapp.azure.com

* Assignment
 Dynamic Static

Listener configuration

* Protocol
 HTTP HTTPS

* Port
80 ✓

Additional Settings

* HTTP2
Disabled Enabled

Web application firewall

* Firewall status
Disabled Enabled

* Firewall mode
Detection Prevention

OK

Check the summary screen in step 3 and click **Create**. This process can take up to **20 minutes, be patient!**

Home > New > Application Gateway > Create application gateway > Summary

Create application gateway ✕

- 1 Basics ✓
Configure basic settings
- 2 Settings ✓
Configure application gateway...
- 3 Summary >
Review and create

Summary

Basics

Name	wafv2
Availability zone	None
Tier	WAF V2
Instance count	2
Subscription	Visual Studio Professional (new) rg-wafv2
Resource Group	(new) rg-wafv2
Location	North Europe

Settings

Virtual network	cplab-vnet
Subnet	WAFSubnet
IP address type	Public
Public IP address	(new) wafv2-ip
Enable SSL for listener	No
Firewall status	Enabled
Firewall mode	Prevention

OK
[Automation options](#)

Adding a backend pool

The CloudGuard template will create an Azure VMSS for you, using the latest Marketplace image. We will use this as the backend target for the Application Gateway.

Select the **Backend Pools** blade of the Application Gateway

The screenshot shows the 'wafv2 - Backend pools' blade in the Azure portal. The left-hand navigation pane includes sections for Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings (Configuration, Web application firewall, Backend pools, HTTP settings, Frontend IP configurations, Listeners, Rules, Health probes, Properties, Locks, Automation script), and Monitoring (Alerts, Metrics, Backend health, Diagnostics logs). The main area shows a table of backend pools:

NAME	RULES ASSOCIATED	TARGETS
appGatewayBackendPool	1	0

Double click the **appGatewayBackendPool** backend pool and you have the option to add a target, as shown below.

Home > wafv2 - Backend pools > Edit backend pool

Edit backend pool

Save Discard Delete

Name
appGatewayBackendPool

Remove all targets from backend pool

Targets
A backend pool can be pointed to a specific virtual machine, virtual machine scale set, an IP address/FQDN or an app service.

VMSS

VMSS	NETWORK INTERFACE CONFIGURATIONS
cpvmss	eth0
Select a virtual machine scale set	Waiting for virtual machine selection

Virtual machine scale set 'cpvmss' was added to this backend pool. Upgrade all the instances of 'cpvmss' for this change to work.

Associated rule
rule1

Choose the CloudGuard VMSS as the target and add **eth0** as the network configuration (the external facing interface). You may see two VMSS entries – one per interface, make sure you use the **external** interface. Click **Save**.

Upgrading the VMSS

For the backend pool change to take effect, the VMSS must be upgraded. This restarts the VMSS instances so they are aware of the Application Gateway monitoring. Go to the VMSS in the Azure Portal, check the box next to the VMSS and click **Upgrade**. This takes around 2 minutes.

Home > Virtual machine scale sets > cpvmss - Instances

Virtual machine scale set - cpvmss - Instances

Start Restart Deallocate Reimage Delete Upgrade

NAME	STATUS	LATEST MODEL
<input type="checkbox"/> cpvmss_2	Running	No

Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems

Settings

- Instances
- Scaling
- Storage
- Operating system
- Size
- Extensions
- Continuous delivery (Preview)
- Identity
- Properties
- Locks
- Automation script

Monitoring

- Alerts
- Metrics

Support + troubleshooting

- New support request

NAME	STATUS	LATEST MODEL
<input checked="" type="checkbox"/> cpmss_2	Running	No

Verifying backend health

To ensure the backend VMSS is being successfully monitored by the Application Gateway, click **Backend Health** under the **Monitoring** blade of the Application Gateway.

appgw - Backend health
Application gateway

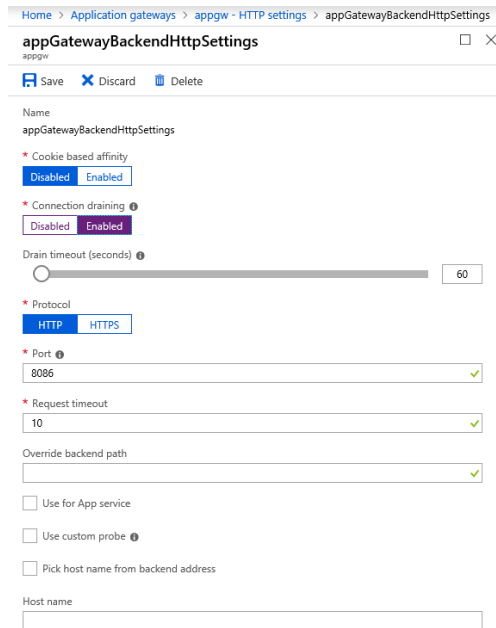
Refresh

SERVER (BACKEND POOL)	PORT (HTTP SETTING)	STATUS	DETAILS
10.0.2.4 (appGatewayBackendPool)	80 (appGatewayBackendHttpSettings)	Healthy	Success

Configuring Backend HTTP Settings

It is possible to configure Application Gateway specific functionality such as Connection Draining prior to putting the gateway into use. This allows sessions to be gracefully closed when scaling operations take place.

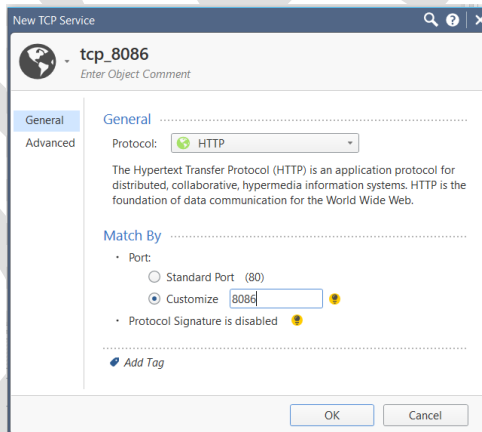
In the backend HTTP settings, the appropriate port should be set for the gateway listener that we use for NAT traffic to a back end web service (TCP 8086 to TCP 80, in this case). This section can be found under **HTTP Settings** blade for the Application Gateway.



Create new TCP service in SmartConsole

The Application Gateway is configured to send traffic to a specific port (in this case 8086), so a new TCP service must be created in SmartConsole so that rules can be configured in the Check Point rule base.

Add the TCP service and set the protocol as HTTP.

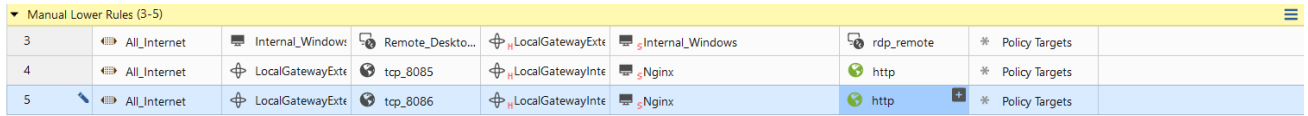


Configuring Access Control and NAT Rules

Add the new service to an access rule – you may need to create dynamic objects for **LocalGatewayExternal** (eth0 on the gateways) and **LocalGatewayInternal** (eth1). Using dynamic objects allows scaling without defining specific gateways.

Application Access (2)									
2	Allow App Gateway traffic to Nginx	* Any	LocalGatewayExternal	* Any	tcp_8086	Accept	Log	* Policy Targets	
Cleanup Rules (3)									
3	Cleanup rule	* Any	* Any	* Any	* Any	Drop	Log	* Policy Targets	

Add a corresponding NAT rule



The screenshot shows a table of Manual Lower Rules with the following columns: Rule ID, Action, Source, Destination, Service, NAT, Policy Targets, and other settings. Rule 5 is highlighted in blue.

Rule ID	Action	Source	Destination	Service	NAT	Policy Targets	Other
3	All_Internet	Internal_Window...	Remote_Desкто...	LocalGatewayExt...	Internal_Windows	rdp_remote	* Policy Targets
4	All_Internet	LocalGatewayExt...	tcp_8085	LocalGatewayInte...	Nginx	http	* Policy Targets
5	All_Internet	LocalGatewayExt...	tcp_8086	LocalGatewayInte...	Nginx	http	* Policy Targets

Publish and install the policy to the CloudGuard gateway(s).

Testing the connection

From a web browser, check the connection to the internal web server (shown as Nginx in the above example) using the frontend IP address of the Application Gateway, you should see the default Nginx landing page.

Troubleshooting

Ensure the backend VMSS is shown as healthy in the Application Gateway.

Verify the backend listener port is configured to the custom TCP port (8086, for example)

Check you have added an entry for the TCP service 8086

Verify the access control policy

Verify the NAT policy

Check route table entries – there should be a UDR for the internal subnet that routes all traffic (0.0.0.0/0) to the internal load balancer and one on the Web subnet that does the same

Ensure all Check Point changes are published and policy changes applied to gateways!

Revision History

<u>Version</u>	<u>Date</u>	<u>Comments</u>
0.1	7/11/2018	Initial draft