

APIM Application Gateway POC

Tuesday, March 5, 2024 3:06 PM

The Apim

<https://learn.microsoft.com/en-us/azure/api-management/api-management-howto-integrate-internal-vnet-appgateway>

Create certificates

Create the root certificate for the self-signed certificate

```
$param1 = @{  
  
    Subject = "CN=contoso.net, C=US"  
  
    KeyLength = 2048  
  
    KeyAlgorithm = 'RSA'  
  
    HashAlgorithm = 'SHA256'  
  
    KeyExportPolicy = 'Exportable'  
  
    NotAfter = (Get-Date).AddYears(5)  
  
    CertStoreLocation = 'Cert:\LocalMachine\My'  
  
    KeyUsage = 'CertSign','CRLSign'  
  
}  
$rootCA = New-SelfSignedCertificate @param1  
  
##### Grab the thumbprint of the root certificate  
$thumb = $rootCA.Thumbprint  
$root = Get-Item -Path Cert:\LocalMachine\My\${thumb}  
  
##### This is a path you want to download the .cer of the root certificate.  
$path = "C:\Users\debar\Desktop\certificates\trustedroot.cer"  
  
##### Export the root certificate in a Base64 encoded X.509 to the path created above  
  
$base64certificate = @"  
  
-----BEGIN CERTIFICATE-----  
  
$([Convert]::ToBase64String($root.Export("Cert"), [System.Base64FormattingOptions]::InsertLineBreaks))  
  
-----END CERTIFICATE-----  
  
"@  
  
Set-Content -Path $path -Value $base64certificate
```

Import the root certificate of the self-signed certificate to the local machine trusted root store
Import-Certificate -CertStoreLocation 'Cert:\CurrentUser\My' -FilePath "C:\Users\debar\Desktop\certificates\trustedroot.cer"

Create a new self-signed certificate for api and then link the root and the self-signed certificate

```
$param2 = @{  
  
    DnsName = "*.contoso.net"  
  
    Subject = "api.contoso.net"  
  
    Signer = $rootCA  
  
    KeyLength = 2048  
  
    KeyAlgorithm = 'RSA'  
  
    HashAlgorithm = 'SHA256'  
  
    KeyExportPolicy = 'Exportable'  
  
    CertStoreLocation = 'Cert:\LocalMachine\My'  
  
    NotAfter = (Get-date).AddYears(2)  
  
}  
  
$selfCert = New-SelfSignedCertificate @param2
```

Export the certificate in .pfx format for the api

```
Export-PfxCertificate -Cert $selfCert -FilePath "C:\Users\debar\Desktop\certificates\gateway.pfx" -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)
```

Create a new self-signed certificate for portal and then link the root and the self-signed certificate

```
$param3 = @{  
  
    DnsName = "*.contoso.net"  
  
    Subject = "portal.contoso.net"  
  
    Signer = $rootCA  
  
    KeyLength = 2048  
  
    KeyAlgorithm = 'RSA'  
  
    HashAlgorithm = 'SHA256'  
  
    KeyExportPolicy = 'Exportable'  
  
    CertStoreLocation = 'Cert:\LocalMachine\My'  
  
    NotAfter = (Get-date).AddYears(2)  
  
}  
  
$selfCert = New-SelfSignedCertificate @param3
```

Export the certificate in .pfx format for the api

```
Export-PfxCertificate -Cert $selfCert -FilePath "C:\Users\debar\Desktop\certificates\portal.pfx" -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)
```

Create a new self-signed certificate for management and then link the root and the selfsigned certificate

```
$param4 = @{  
  
    DnsName = "*.contoso.net"  
  
    Subject = "management.contoso.net"  
  
    Signer = $rootCA
```

```

KeyLength = 2048

KeyAlgorithm = 'RSA'

HashAlgorithm = 'SHA256'

KeyExportPolicy = 'Exportable'

CertStoreLocation = 'Cert:\LocalMachine\My'

NotAfter = (Get-date).AddYears(2)
}

$SelfCert = New-SelfSignedCertificate @param4

##### Export the certificate in .pfx format for the api

Export-PfxCertificate -Cert $SelfCert -FilePath "C:\Users\debar\Desktop\certificates\management.pfx" -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)

#####

## APIM Integration with Application Gateway
Connect-AzAccount
Set-AzContext -Subscription 4ce58615-55cb-48bf-b92f-cf6cc7b80a64

# These variables must be changed.
$subscriptionId = "4ce58615-55cb-48bf-b92f-cf6cc7b80a64" # GUID of your Azure subscription
$domain = "contoso.net" # The custom domain for your certificate
$apimServiceName = "apim-contoso-004" # API Management service instance name, must be globally unique
$apimDomainNameLabel = $apimServiceName # Domain name label for API Management's public IP address, must be globally unique
$apimAdminEmail = "debarshi.eee@gmail.com" # Administrator's email address- use your email address
$gatewayHostname = "api.$domain" # API gateway host
$portalHostname = "portal.$domain" # API developer portal host
$managementHostname = "management.$domain" # API management endpoint host
$baseCertPath = "C:\Users\debar\Desktop\certificates\" # The base path where all certificates are stored
$trustedRootCertPath = "$baseCertPath\trustedroot.cer" # Full path to contoso.net trusted root .cer file
$gatewayCertPfxPath = "$baseCertPath\gateway.pfx" # Full path to api.contoso.net .pfx file
$portalCertPfxPath = "$baseCertPath\portal.pfx" # Full path to portal.contoso.net .pfx file
$managementCertPfxPath = "$baseCertPath\management.pfx" # Full path to management.contoso.net .pfx file

$gatewayCertPfxPassword = "certificatePassword123" # Password for api.contoso.net pfx certificate
$portalCertPfxPassword = " " # Password for portal.contoso.net pfx certificate
$managementCertPfxPassword = "certificatePassword123" # Password for management.contoso.net pfx certificate

# These variables may be changed.
$resGroupName = "rg-apim-agw" # Resource group name that will hold all assets
$location = "East US" # Azure region that will hold all assets
$apimOrganization = "Contoso" # Organization name
$appgwName = "agw-contoso-001" # The name of the Application Gateway

Get-AzSubscription -SubscriptionId $subscriptionId | Select-AzSubscription

New-AzResourceGroup -Name $resGroupName -Location $location

## Create NSG and NSG Rules for Application Gateway subnet

$appgwRule1 = New-AzNetworkSecurityRuleConfig -Name appgw-in -Description "AppGw inbound" `
-Access Allow -Protocol * -Direction Inbound -Priority 100 -SourceAddressPrefix `
GatewayManager -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 65200-65535
$appgwRule2 = New-AzNetworkSecurityRuleConfig -Name appgw-in-internet -Description "AppGw inbound Internet" `
-Access Allow -Protocol "TCP" -Direction Inbound -Priority 110 -SourceAddressPrefix `
Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 443

$appgwNsg = New-AzNetworkSecurityGroup -ResourceGroupName $resGroupName -Location $location -Name `
"nsg-agw" -SecurityRules $appgwRule1, $appgwRule2

## Create NSG and NSG Rules for API Management
$apimRule1 = New-AzNetworkSecurityRuleConfig -Name APIM-Management -Description "APIM inbound" `
-Access Allow -Protocol Tcp -Direction Inbound -Priority 100 -SourceAddressPrefix ApiManagement `
-SourcePortRange * -DestinationAddressPrefix VirtualNetwork -DestinationPortRange 3443
$apimRule2 = New-AzNetworkSecurityRuleConfig -Name AllowAppGatewayToAPIM -Description "Allows inbound App Gateway traffic to APIM" `
-Access Allow -Protocol Tcp -Direction Inbound -Priority 110 -SourceAddressPrefix "10.0.0.0/24" `
-SourcePortRange * -DestinationAddressPrefix "10.0.2.0/24" -DestinationPortRange 443
$apimRule3 = New-AzNetworkSecurityRuleConfig -Name AllowAzureLoadBalancer -Description "Allows inbound Azure Infrastructure Load Balancer traffic to APIM" `
-Access Allow -Protocol Tcp -Direction Inbound -Priority 120 -SourceAddressPrefix AzureLoadBalancer `
-SourcePortRange * -DestinationAddressPrefix "10.0.2.0/24" -DestinationPortRange 6390
$apimRule4 = New-AzNetworkSecurityRuleConfig -Name AllowKeyVault -Description "Allows outbound traffic to Azure Key Vault" `
-Access Allow -Protocol Tcp -Direction Outbound -Priority 100 -SourceAddressPrefix "10.0.2.0/24" `
-SourcePortRange * -DestinationAddressPrefix AzureKeyVault -DestinationPortRange 443

$apimNsg = New-AzNetworkSecurityGroup -ResourceGroupName $resGroupName -Location $location -Name `
"nsg-apim" -SecurityRules $apimRule1, $apimRule2, $apimRule3, $apimRule4

## Vnet and subnet configuration for APIM and Application Gateway

$appGatewaySubnet = New-AzVirtualNetworkSubnetConfig -Name "appGatewaySubnet" -NetworkSecurityGroup $appgwNsg -AddressPrefix "10.0.0.0/24"

$apimSubnet = New-AzVirtualNetworkSubnetConfig -Name "apimSubnet" -NetworkSecurityGroup $apimNsg -AddressPrefix "10.0.1.0/24"

$vnet = New-AzVirtualNetwork -Name "vnet-contoso" -ResourceGroupName $resGroupName `
-Location $location -AddressPrefix "10.0.0.0/16" -Subnet $appGatewaySubnet, $apimSubnet

$appGatewaySubnetData = $vnet.Subnets[0]
$apimSubnetData = $vnet.Subnets[1]

## Create APIM inside the Virtual Network
$apimPublicAddressId = New-AzPublicIpAddress -ResourceGroupName $resGroupName -Name "pip-apim" -location $location `
-AllocationMethod Static -Sku Standard -Force -DomainNameLabel $apimDomainNameLabel

$apimService = Get-AzApiManagement -ResourceGroupName apimtest -Name apim-contoso-004

## Set up custom domain names in API Management by using the pfx certificates we have created before

## Configure a private zone for DNS resolution in the virtual network

$myZone = New-AzPrivateDnsZone -Name $domain -ResourceGroupName $resGroupName
$link = New-AzPrivateDnsVirtualNetworkLink -ZoneName $domain `
-ResourceGroupName $resGroupName -Name "mylink" `
-VirtualNetworkId $vnet.Id

$apimIP = $apimService.PrivateIPAddresses[0]

New-AzPrivateDnsRecordSet -Name api -RecordType A -ZoneName $domain `
-ResourceGroupName $resGroupName -Ttl 3600 `
-PrivateDnsRecords (New-AzPrivateDnsRecordConfig -IPv4Address $apimIP)
New-AzPrivateDnsRecordSet -Name portal -RecordType A -ZoneName $domain `
-ResourceGroupName $resGroupName -Ttl 3600 `
-PrivateDnsRecords (New-AzPrivateDnsRecordConfig -IPv4Address $apimIP)
New-AzPrivateDnsRecordSet -Name management -RecordType A -ZoneName $domain `
-ResourceGroupName $resGroupName -Ttl 3600 `
-PrivateDnsRecords (New-AzPrivateDnsRecordConfig -IPv4Address $apimIP)

$appGatewaySubnetData = Get-AzVirtualNetwork -Name vnet-contoso -ResourceGroupName rg-apim-agw

## Setup and application gateway

$publicip = New-AzPublicIpAddress -ResourceGroupName $resGroupName `

```

```

-name "pip-appgateway" -location $location -AllocationMethod Static -Sku Standard

$gipconfig = New-AzApplicationGatewayIPConfiguration -Name "gatewayIP01" -Subnet $appGatewaySubnetData.Subnets[0]

$fp01 = New-AzApplicationGatewayFrontendPort -Name "port01" -Port 443

$flpconfig01 = New-AzApplicationGatewayFrontendIPConfig -Name "frontend1" -PublicIPAddress $publicip

## SSL Certificate for the apim endpoints

$certGateway = New-AzApplicationGatewaySslCertificate -Name "gatewaycert" `
-CertificateFile $gatewayCertPfxPath -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)

$certPortal = New-AzApplicationGatewaySslCertificate -Name "portalcert" `
-CertificateFile $portalCertPfxPath -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)

$certManagement = New-AzApplicationGatewaySslCertificate -Name "managementcert" `
-CertificateFile $managementCertPfxPath -Password (ConvertTo-SecureString -String 'certificatePassword123' -AsPlainText -Force)

## Application Gateway Listener configuration

$gatewayListener = New-AzApplicationGatewayHttpListener -Name "gatewaylistener" `
-Protocol "Https" -FrontendIPConfiguration $flpconfig01 -FrontendPort $fp01 `
-SslCertificate $certGateway -HostName $gatewayHostname -RequireServerNameIndication true

$portalListener = New-AzApplicationGatewayHttpListener -Name "portallistener" `
-Protocol "Https" -FrontendIPConfiguration $flpconfig01 -FrontendPort $fp01 `
-SslCertificate $certPortal -HostName $portalHostname -RequireServerNameIndication true

$managementListener = New-AzApplicationGatewayHttpListener -Name "managementlistener" `
-Protocol "Https" -FrontendIPConfiguration $flpconfig01 -FrontendPort $fp01 `
-SslCertificate $certManagement -HostName $managementHostname -RequireServerNameIndication true

## Application Gateway probe configuration

$apimGatewayProbe = New-AzApplicationGatewayProbeConfig -Name "apimgatewayprobe" `
-Protocol "Https" -HostName $gatewayHostname -Path "/status-0123456789abcde" `
-Interval 30 -Timeout 120 -UnhealthyThreshold 8
$apimPortalProbe = New-AzApplicationGatewayProbeConfig -Name "apimportalprobe" `
-Protocol "Https" -HostName $portalHostname -Path "/signin" `
-Interval 60 -Timeout 300 -UnhealthyThreshold 8
$apimManagementProbe = New-AzApplicationGatewayProbeConfig -Name "apimmanagementprobe" `
-Protocol "Https" -HostName $managementHostname -Path "/serviceStatus" `
-Interval 60 -Timeout 300 -UnhealthyThreshold 8

$trustedRootCert = New-AzApplicationGatewayTrustedRootCertificate -Name "allowlistcert1" -CertificateFile $trustedRootCertCerPath

## Application Gateway HTTP settings

$apimPoolGatewaySetting = New-AzApplicationGatewayBackendHttpSettings -Name "apimPoolGatewaySetting" `
-Port 443 -Protocol "Https" -CookieBasedAffinity "Disabled" -Probe $apimGatewayProbe `
-TrustedRootCertificate $trustedRootCert -PickHostNameFromBackendAddress -RequestTimeout 180
$apimPoolPortalSetting = New-AzApplicationGatewayBackendHttpsSettings -Name "apimPoolPortalSetting" `
-Port 443 -Protocol "Https" -CookieBasedAffinity "Disabled" -Probe $apimPortalProbe `
-TrustedRootCertificate $trustedRootCert -PickHostNameFromBackendAddress -RequestTimeout 180
$apimPoolManagementSetting = New-AzApplicationGatewayBackendHttpSettings -Name "apimPoolManagementSetting" `
-Port 443 -Protocol "Https" -CookieBasedAffinity "Disabled" -Probe $apimManagementProbe `
-TrustedRootCertificate $trustedRootCert -PickHostNameFromBackendAddress -RequestTimeout 180

## Application Gateway backend pool settings

$apimGatewayBackendPool = New-AzApplicationGatewayBackendAddressPool -Name "gatewaybackend" `
-BackendFqdns $gatewayHostname
$apimPortalBackendPool = New-AzApplicationGatewayBackendAddressPool -Name "portalbackend" `
-BackendFqdns $portalHostname
$apimManagementBackendPool = New-AzApplicationGatewayBackendAddressPool -Name "managementbackend" `
-BackendFqdns $managementHostname

## Application Gateway Routing rules

$gatewayRule = New-AzApplicationGatewayRequestRoutingRule -Name "gatewayrule" `
-RuleType Basic -HttpListener $gatewayListener -BackendAddressPool $apimGatewayBackendPool `
-BackendHttpSettings $apimPoolGatewaySetting
$portalRule = New-AzApplicationGatewayRequestRoutingRule -Name "portalrule" `
-RuleType Basic -HttpListener $portalListener -BackendAddressPool $apimPortalBackendPool `
-BackendHttpSettings $apimPoolPortalSetting
$managementRule = New-AzApplicationGatewayRequestRoutingRule -Name "managementrule" `
-RuleType Basic -HttpListener $managementListener -BackendAddressPool $apimManagementBackendPool `
-BackendHttpSettings $apimPoolManagementSetting

$sku = New-AzApplicationGatewaySku -Name Standard_v2 -Tier Standard_v2 -Capacity 1

$policy = New-AzApplicationGatewaySslPolicy -PolicyType Predefined -PolicyName AppGwSslPolicy20220101

$appgw = New-AzApplicationGateway -Name $appgwName -ResourceGroupName $resGroupName -Location $location `
-BackendAddressPools $apimGatewayBackendPool,$apimPortalBackendPool,$apimManagementBackendPool `
-BackendHttpSettingsCollection $apimPoolGatewaySetting, $apimPoolPortalSetting, $apimPoolManagementSetting `
-FrontendIPConfigurations $flpconfig01 -GatewayIPConfigurations $gipconfig -FrontendPorts $fp01 `
-HttpListeners $gatewayListener,$portalListener,$managementListener `
-RequestRoutingRules $gatewayRule,$portalRule,$managementRule `
-Sku $sku -WebApplicationFirewallConfig $config -SslCertificates $certGateway,$certPortal,$certManagement `
-TrustedRootCertificate $trustedRootCert -Probes $apimGatewayProbe,$apimPortalProbe,$apimManagementProbe `
-SslPolicy $policy













```

REQUIREMENTS

- A dedicated subnet
- A Trusted root certificate
- Three pfx certificates for three portals
- Api management with public ip and private ip deployed in dedicated subnet
- NSG rules as mentioned down below for application gateway and api management

ALL RESOURCES RECORDED

=====

	agw-contoso-001	Application gateway	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	apim-contoso-004	API Management service	apim-test	East US	Visual Studio Professional Subscription	***
	contoso.net	Private DNS zone	rg-apim-agw	Global	Visual Studio Professional Subscription	***
	myVm	Virtual machine	apim-test	East US	Visual Studio Professional Subscription	***
	myVm-ip	Public IP address	apim-test	East US	Visual Studio Professional Subscription	***
	myVm-vnet	Virtual network	apim-test	East US	Visual Studio Professional Subscription	***
	myVm.128	Network interface	apim-test	East US	Visual Studio Professional Subscription	***
	myVm_OsDisk_1_b728f81a777f4584997507d4d4c3db75	Disk	APIMTEST	East US	Visual Studio Professional Subscription	***
	NetworkWatcher-eastus	Network Watcher	NetworkWatcherRG	East US	Visual Studio Professional Subscription	***
	nsg-agw	Network security group	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	nsg-apim	Network security group	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	nsg-vm	Network security group	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	ipr-apim	Public IP address	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	ipr-appgateway	Public IP address	rg-apim-agw	East US	Visual Studio Professional Subscription	***
	vnet-contoso	Virtual network	rg-apim-agw	East US	Visual Studio Professional Subscription	***

TRUSTED ROOT CERTIFICATES

certificates

gateway.pfx management.pfx portal.pfx trustedroot.cer

VIRTUAL NETWORK SUBNET , AZURE PRIVATE DNS ZONE AND NSG CONFIGURATION

vnet-contoso | Subnets

Virtual network

Search

+ Subnet + Gateway subnet Refresh Manage users Delete

Search subnets

Name ↑↓	IPv4 ↑↓	IPv6 ↑↓	Available IPs ↑↓	Delegated to ↑↓	Security group ↑↓	Rot
appGatewaySubnet	10.0.0.0/24	-	availability dependent ...	-	nsg-agw	-
apimsubnet	10.0.2.0/24	-	249	-	nsg-apim	-

nsq-agw

Network security group

Search

Move Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Monitoring

Alerts

Diagnostic settings

Logs

NSG flow logs

Essentials

Resource group (move) : rg-apim-agw

Location : East US

Subscription (move) : Visual Studio Professional Subscription

Subscription ID : 4ce58615-55cb-48bf-b92f-dcecc7b80a64

Tags (edit) : Add tags

Custom security rules : 2 inbound, 0 outbound

Associated with : 1 subnets, 0 network interfaces

Filter by name

Port == all

Protocol == all

Source == all

Destination == all

Action == all

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound Security Rules						
100	appgw-in	65200-65535	Any	GatewayManager	Any	Allow
110	appgw-in-internet	443	TCP	Internet	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerIn...	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
Outbound Security Rules						
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

nsq-apim

Network security group

Search

Move Delete Refresh Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Inbound security rules

Outbound security rules

Network interfaces

Subnets

Properties

Locks

Monitoring

Alerts

Diagnostic settings

Logs

NSG flow logs

Automation

CLI / PS

Tasks (review)

Essentials

Resource group (move) : rg-apim-agw

Location : East US

Subscription (move) : Visual Studio Professional Subscription

Subscription ID : 4ce58615-55cb-48bf-b92f-dcecc7b80a64

Tags (edit) : Add tags

Custom security rules : 4 inbound, 1 outbound

Associated with : 1 subnets, 0 network interfaces

Filter by name

Port == all

Protocol == all

Source == all

Destination == all

Action == all

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound Security Rules						
100	APIM-Management	3443	Tcp	ApiManagement	VirtualNetwork	Allow
110	AllowAppGatewayToAPIM	443	TCP	10.0.0.0/24	10.0.2.0/24	Allow
120	AllowAzureLoadBalancer	6390	TCP	AzureLoadBalancer	10.0.2.0/24	Allow
130	AllowCidrBlockCustom443-3443--	Any	Any	10.1.0.0/16	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
Outbound Security Rules						
100	AllowKeyVault	443	TCP	10.0.2.0/24	AzureKeyVault	Allow
65000	AllowVnetOutBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowInternetOutBound	Any	Any	Any	Internet	Allow
65500	DenyAllOutBound	Any	Any	Any	Any	Deny

contoso.net

Private DNS zone

Search

Record set

Move

Delete zone

Refresh

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Virtual network links

Properties

Locks

Monitoring

Alerts

Metrics

Automation

Tasks (preview)

Export template

Essentials

JSON View

Resource group (move)

Subscription (move)

Subscription ID

Tags (edit)

rg-arm-aggw

Visual Studio Professional Subscription

4ce58615-55cb-48bf-b92f-dccc7b80a64

Add tags

You can search for record sets that have been loaded on this page. If you don't see what you're looking for, you can try scrolling to allow more record sets to load.

Search record sets

Name	Type	TTL	Value	Auto registered
@	SOA	3600	Email: azureprivatedns-host.microsoft.com Host: azureprivatedns.net Refresh: 3600 Retry: 300 Expire: 2419200 Minimum TTL: 10 Serial number: 1	False
api	A	3600	10.0.2.4	False
management	A	3600	10.0.2.4	False
portal	A	3600	10.0.2.4	False

APIM CONFIGURATION

apim-contoso-004 | Network

API Management service

Search

Issues (deprecated)

Monitoring

Analytics

Application Insights

Alerts

Metrics

Diagnostic settings

Logs

Advisor recommendations

Workbooks

Deployment + Infrastructure

Pricing tier

Locations

Scale out (auto-scale)

Gateways

External cache

Virtual network

Inbound private endpoint connections

Network status

Save Discard Diagnose Apply network configuration Send us your feedback

Securely access resources available in or through your Azure VNET. [Learn more.](#)

Changes can take from 15 to 45 minutes to apply. Please ensure you have the [required ports](#) unblocked before enabling virtual network connectivity to

Changing API Management service VNET will cause public VIP to change. [Learn more.](#)

Virtual network ☐ None ☐ External ☒ Internal

Location	Virtual network	Subnet
East US	vnet-contoso	apimsubnet

Virtual network

API Management service

Virtual network

Subnet

Management public IP address

apim-contoso-004 | Custom domains

API Management service

Search

Issues (deprecated)

Monitoring

Analytics

Application Insights

Alerts

Metrics

Diagnostic settings

Logs

Advisor recommendations

Workbooks

Deployment + Infrastructure

Pricing tier

Locations

Scale out (auto-scale)

Gateways

External cache

Custom domains

Network

+ Add Save Discard Refresh Columns

By default, your API Management service instance is available through *.azure-apim.net subdomain (for example, contoso.azure-apim.net). You can also expose the service through your own domain name, such as contoso.com. [Learn more](#)

Endpoint	Hostname	Source	Certificate	Negotiate client ce...	Default SSL binding	Certificate key vau...
Developer portal	portal.contoso.net	Custom	Expiry: 3/5/2026, thumbprint:
Management	management.contoso.net	Custom	Expiry: 3/5/2026, thumbprint:
Gateway	apim-contoso-004.azure-apim.net	Built-in				...
Gateway	api.contoso.net	Custom	Expiry: 3/5/2026, thumbprint: ...		✓	...

Developer portal

API Management service

Type

Developer portal

Hostname *

portal.contoso.net

Certificate

Custom

Certificate

Expiry: 3/5/2026, thumbprint: 5567ECAB8EA21453EB72CDB164E23683638B5719

Change

apim-contoso-004 | APIs

API Management service

Search

Developer portal

Send us your feedback

Search APIs

Filter by tags

Group by tag

+ Add API

All APIs

Test Api

REVISION 1

CREATED Mar 5, 2024, 2:35:00 AM

Design

Settings

Test

Revisions (1)

Change log

Search operations

Filter by tags

Group by tag

+ Add operation

All operations

GET Test call

Mocking is enabled

Frontend

GET /test

Responses

200 OK

Definitions

Inbound processing

Modify the request before it is sent to the backend service.

Policies

+ Add policy

base

Backend

HTTP(s) endpoint

Policies

base

Outbound processing

Modify the response before it is sent to the client.

```
1 {
2   "Sample Response": "Hello Dear your API is working fine as desired"
3 }
```

APPLICATION GATEWAY CONFIGURATION

agw-contoso-001

Application gateway

Search

Delete

Refresh

Feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Configuration

Web application firewall

Backend pools

Backend settings

Frontend IP configurations

Private link

SSL settings

Listeners

Rules

Rewrites

Health probes

Properties

Locks

Monitoring

Alerts

Essentials

JSON View

Resource group (move) : rg-apim-agw

Virtual network/subnet : vnet-contoso/appGatewaySubnet

Location : East US

Frontend public IP address : 57.151.45.233 (ip-apogateway)

Subscription (move) : Visual Studio Professional Subscription

Frontend private IP address : -

Subscription ID : 4ce58615-55cb-48bf-b92f-dfcc7b80a64

Tier : Standard V2

Availability zone : -

Tags (edit) : Add tags

Show data for last : 1 hour 6 hours 12 hours 1 day 7 days 30 days

Sum Total Requests

Sum Failed Requests

Total Requests (Sum)

33

Failed Requests (Sum)

6

agw-contoso-001 | Backend pools

Application gateway

Search

+ Add

Refresh

Backend health

Feedback

Search backend pools

Name

Rules associated

Targets

gatewaybackend

1

1

...

portalbackend

1

1

...

managementbackend

1

1

...

Home > All resources > apw-contoso-001 > Backend pools >

Edit backend pool

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.

Name
gatewaybackend

Add backend pool without targets
☐ Yes ☒ No

Backend targets
1 item

Target type	Target
IP address or FQDN	api.contoso.net
IP address or FQDN	

Here we have to refer to trusted root certificate and same settings we have to do for all three backend settings

Backend settings

+ Add Backend health Feedback

Search Backend settings

Name	Power state
apimPoolGatewaySetting	44%
apimPoolPortalSetting	44%
apimPoolManagementSetting	44%

Add Backend setting

Backend settings name
apimPoolPortalSetting

Backend protocol
☐ HTTP ☒ HTTPS

Backend port *
443

Backend server's certificate is issued by a well-known CA
☐ Yes ☒ No

Upload Root CA certificate

Information You must upload the Root certificate (CER) of the backend server to this Backend Setting. If a Private CA has issued that certificate or if it is a self-generated one. This root certificate allows your application gateway to complete the certificate chain validation.

- To extract the backend server's Root certificate, follow the [troubleshooting guide](#).
- You can also create your own Root CA and Server certificates. [Learn more](#).

Certificate

+ Add certificate

Additional settings

Cookie-based affinity
☐ Enable ☒ Disable

Connection draining
☐ Enable ☒ Disable

Request time-out (seconds) *
180

Override backend path

Host name

By default, the Application Gateway sends the same HTTP host header to the backend as it receives from the client. If your backend application/service requires a specific host value, you can override it using this setting.

Override with new host name
☒ Yes ☐ No

Information If the backend service is a multi-tenant Azure service such as App Services, Functions, or Portal Apps, we recommend using [Custom domain method](#), instead of overriding the hostname. Using override host name with default domains (azurewebsites.net, azuremicroservices.io, etc.) is good only for the basic tests and operations.

Host name override
☒ Pick host name from backend target
☐ Override with specific domain name

Use custom probe
☒ Yes ☐ No

Custom probe *

Save Cancel

agw-contoso-001 | Frontend IP configurations

Application gateway

Search

Feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Configuration

Web application firewall

Backend pools

Backend settings

Frontend IP configurations

Search frontend IP configurations

Type	Status	Name	IP address	Associated listeners
Public	Configured	frontend1	57.151.45.233 (pip-appgateway)	gatewaylistener, 2 more
Private	Not configured	-	-	-

LISTENER SETTINGS

Listeners

Listener TLS certificates (Preview)

+ Add listener

Refresh

Feedback

Application Gateway provides native support for WebSocket across all gateway sizes. There is no additional configuration required to enable or disable WebSocket support. If a WebSocket traffic is received on the Application Gateway, it is automatically directed to the WebSocket enabled backend server using the appropriate backend pool as specified in application gateway rules.
[Learn more about listeners and WebSocket support.](#)

Search listeners

Name	Port	Protocol	Frontend IP	Associated rule	Host name
managementlistener	443	HTTPS	Public	managementrule	> management.contoso.net ...
portallistener	443	HTTPS	Public	portallrule	> portal.contoso.net ...
gatewaylistener	443	HTTPS	Public	gatewayrule	> api.contoso.net ...

This are the pfx certificates which we have created and used in the apim

agw-contoso-001 | Listeners

Application gateway

Search

Feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Settings

Configuration

Web application firewall

Backend pools

Backend settings

Frontend IP configurations

Private link

SSL settings

Listeners

Rules

Rewrites

Health probes

Properties

Locks

Monitoring

Listeners

Listener TLS certificates (Preview)

+ Add certificate

Refresh

Bulk update

Feedback

This section allows you to view and manage all TLS certificates for listeners. An TLS certificate must be re-uploaded a certificate for renewals or perform other management operations. [Learn more.](#)

Search Listener Certificate

Certificate name	Type	Key vault ID
gatewaycert	Uploaded	NA
portalcert	Uploaded	NA
managementcert	Uploaded	NA

gatewaycert

agw-contoso-001

Edit

Delete

Refresh

Certificate Information

Listener certificate name

gatewaycert

Type

Uploaded

Status

Active

Expiry

Mar 4, 2026

SHA-1 Thumbprint

3A5B83B5B73EAB09BAB077EFB12D06F905...

Subject Common Name

api.contoso.net

Issuer Common Name

contoso.net

Subject Alternative Name

*.contoso.net

Associated listener

gatewaylistener

Close

[Home](#) > [All resources](#) > [agw-contoso-001](#) | [Listeners](#) >

managementlistener

agw-contoso-001

Listener name ⓘ

managementlistener

Frontend IP * ⓘ

Public

Protocol ⓘ

☐ HTTP ☒ HTTPS

Port * ⓘ

443

Choose a certificate

☐ Create new ☒ Select existing

Certificate *

managementcert

☐ Renew or edit selected certificate

☐ Enable SSL Profile ⓘ

Associated rule

[managementrule](#)

Listener type ⓘ

☐ Basic ☒ Multi site

Host type ⓘ

☒ Single ☐ Multiple/Wildcard

Host name * ⓘ

management.contoso.net

[Home](#) > [All resources](#) > [agw-contoso-001](#) | [Listeners](#) >

portallistener

agw-contoso-001

Listener name ⓘ

portallistener

Frontend IP * ⓘ

Public

Protocol ⓘ

☐ HTTP ☒ HTTPS

Port * ⓘ

443

Choose a certificate

☐ Create new ☒ Select existing

Certificate *

portalcert

☐ Renew or edit selected certificate

☐ Enable SSL Profile ⓘ

Associated rule

[portallrule](#)

Listener type ⓘ

☐ Basic ☒ Multi site

Host type ⓘ

☒ Single ☐ Multiple/Wildcard

Host name * ⓘ

ROUTING RULES

Rules ☆ ...

+ Routing rule ❤️ Backend health 📄 Feed

🔍 Search rules

Name

gatewayrule

portairule

managementrule

gatewayrule

agw-contoso-001

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name

gatewayrule

Priority * ⓘ

10

Listener

Backend targets

A listener "listens" on a specified port and IP address for traffic that uses a specified protocol. If the listener criteria are met, the application gateway will apply this routing rule.

Listener *

gatewaylistener

ules ☆ ...

+ Routing rule ❤️ Backend health 📄 Feed

🔍 Search rules

Name

gatewayrule

portairule

managementrule

gatewayrule

agw-contoso-001

Configure a routing rule to send traffic from a given frontend IP address to one or more backend targets. A routing rule must contain a listener and at least one backend target.

Rule name

gatewayrule

Priority * ⓘ

10

Listener

Backend targets

Choose a backend pool to which this routing rule will send traffic. You will also need to specify a set of Backend settings that define the behavior of the routing rule.

Target type

☒ Backend pool ☐ Redirection

Backend target * ⓘ

gatewaybackend

Backend settings * ⓘ

apimPoolGatewaySetting

Save

Cancel

Health Probe

apimmanagementprobe

agw-contoso-001

Name

apimmanagementprobe

Protocol *

☐ HTTP

☒ HTTPS

Pick host name from backend settings

☐ Yes

☒ No

Host *

Pick port from backend settings

☒ Yes

☐ No

Path *

Interval (seconds) *

Timeout (seconds) *

Unhealthy threshold *

Use probe matching conditions

☐ Yes

☒ No

Backend settings

☒ I want to test the backend health before adding the health probe

Test

Discard

apimportalprobe

agw-contoso-001

Name

apimportalprobe

Protocol *

☐ HTTP

☒ HTTPS

Pick host name from backend settings

☐ Yes

☒ No

Host *

Pick port from backend settings

☒ Yes

☐ No

Path *

Interval (seconds) *

Timeout (seconds) *

Unhealthy threshold *

Use probe matching conditions

☐ Yes

☒ No

Backend settings

☒ I want to test the backend health before adding the health probe

Test

Discard

apimgatewayprobe
agw-contoso-001

Name: apimgatewayprobe

Protocol: ☐ HTTP ☒ HTTPS

Pick host name from backend settings: ☐ Yes ☒ No

Host:

Pick port from backend settings: ☒ Yes ☐ No

Path:

Interval (seconds):

Timeout (seconds):

Unhealthy threshold:

Use probe matching conditions: ☐ Yes ☒ No

Backend settings:

☒ I want to test the backend health before adding the health probe

Entered application gateway public ip in the host file to pretend like dns registration

```

# Added by Docker Desktop
192.168.1.8 host.docker.internal
192.168.1.8 gateway.docker.internal
# To allow the same kube context to work on
host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
57.151.45.233 api.contoso.net
57.151.45.233 management.contoso.net
57.151.45.233 portal.contoso.net

```