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 self-signed 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-cfecc7b80a64

# These variables must be changed.
$subscriptionId                 = "4ce58615-55cb-48bf-b92f-cfecc7b80a64"     # 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
$trustedRootCertCerPath         = "${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
$apimPublicIpAddressId = 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

$fipconfig01 = 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 $fipconfig01 -FrontendPort $fp01 `
  -SslCertificate $certGateway -HostName $gatewayHostname -RequireServerNameIndication true

$portalListener = New-AzApplicationGatewayHttpListener -Name "portallistener" `
  -Protocol "Https" -FrontendIPConfiguration $fipconfig01 -FrontendPort $fp01 `
  -SslCertificate $certPortal -HostName $portalHostname -RequireServerNameIndication true

$managementListener = New-AzApplicationGatewayHttpListener -Name "managementlistener" `
  -Protocol "Https" -FrontendIPConfiguration $fipconfig01 -FrontendPort $fp01 `
  -SslCertificate $certManagement -HostName $managementHostname -RequireServerNameIndication true

  ## Application Gateway probe configuration

$apimGatewayProbe = New-AzApplicationGatewayProbeConfig -Name "apimgatewayprobe" `
  -Protocol "Https" -HostName $gatewayHostname -Path "/status-0123456789abcdef" `
  -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-AzApplicationGatewayBackendHttpSettings -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 $fipconfig01 -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

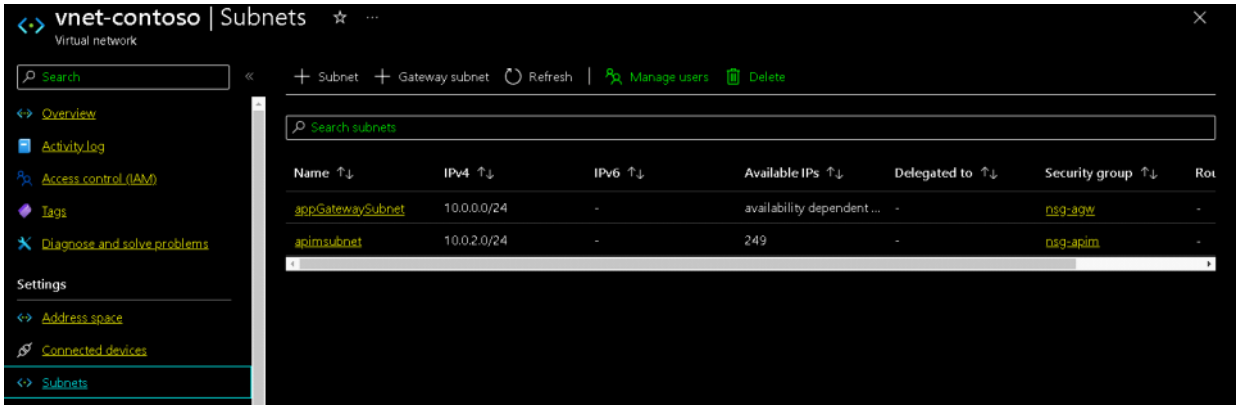
- A dedicated subnet
- A Trusted root certificate
- Three pfx certificates for three portals
- Api management with public ip and private ip deployed in dedicated subnet
- NSG rules as mentioned down below for application gateway and api management

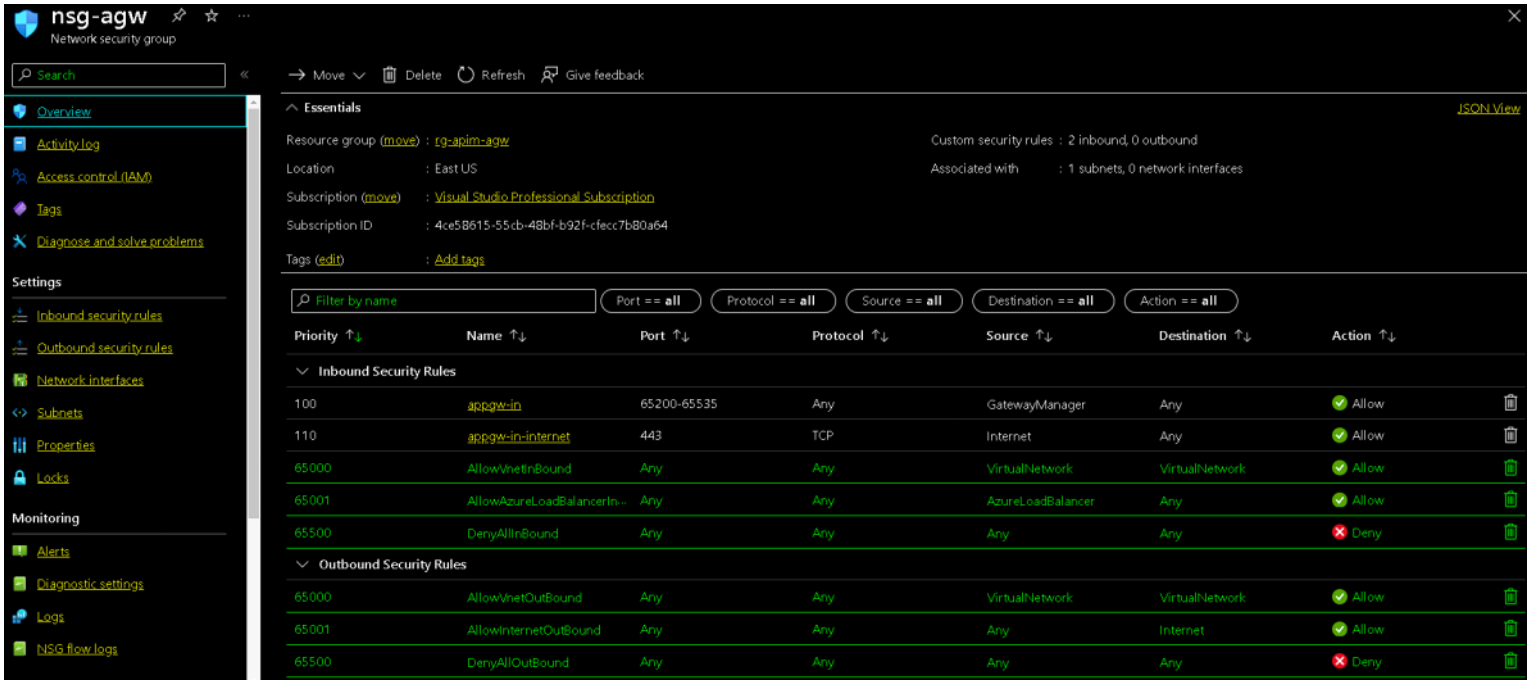
ALL RESOURCES RECORDED
==========

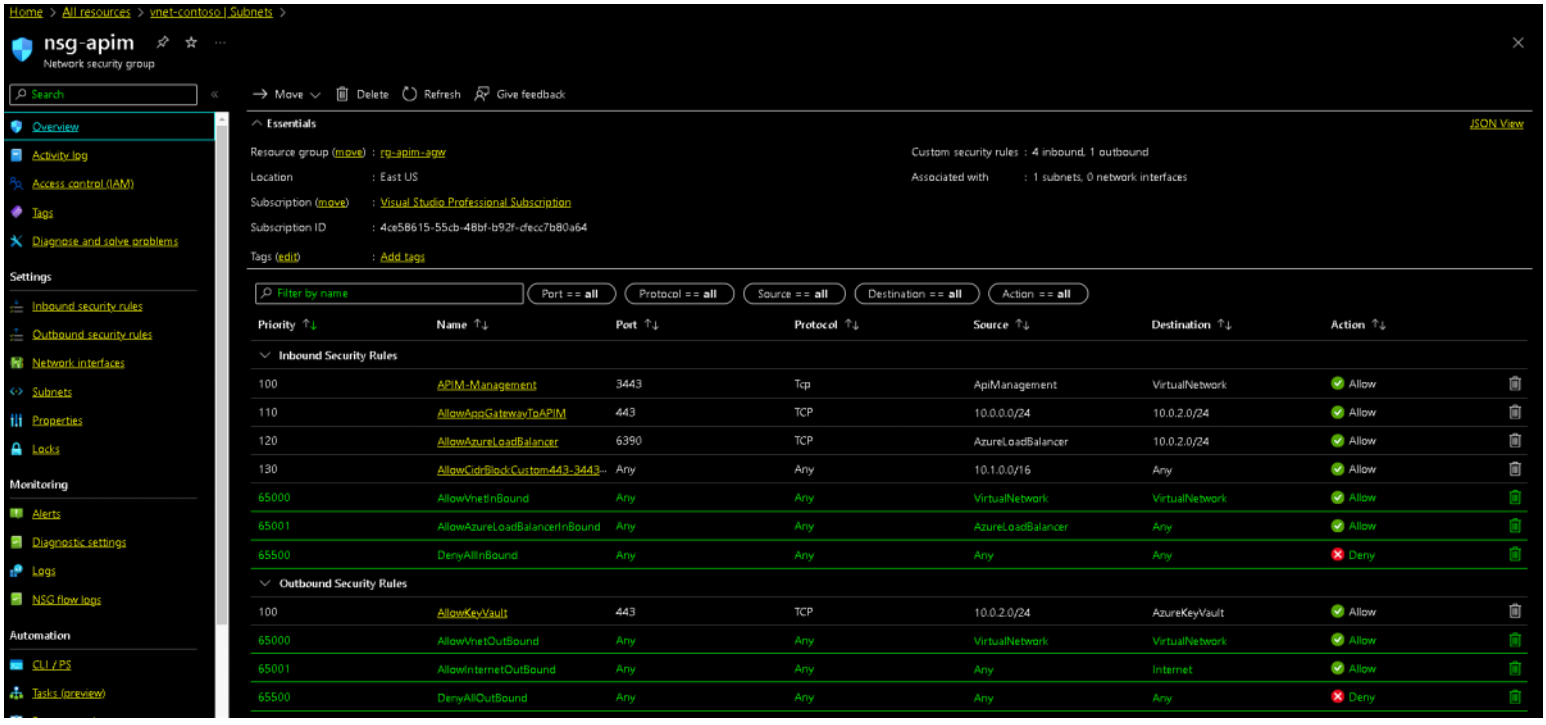
agw-contoso-001	Application gateway	rg-apim-agw	East US	Visual Studio Professional Subscription	•••
apim-contoso-004	API Management service	apimtest	East US	Visual Studio Professional Subscription	•••
contoso.net	Private DNS zone	rg-apim-agw	Global	Visual Studio Professional Subscription	•••
myVm	Virtual machine	apimtest	East US	Visual Studio Professional Subscription	•••
myVm-ip	Public IP address	apimtest	East US	Visual Studio Professional Subscription	•••
myVm-vnet	Virtual network	apimtest	East US	Visual Studio Professional Subscription	•••
myvm128	Network Interface	apimtest	East US	Visual Studio Professional Subscription	•••
myVm_OsDisk_1_b728fe1a777f4584997507d4c4c3db75	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	•••
pip-apim	Public IP address	rg-apim-agw	East US	Visual Studio Professional Subscription	•••
pip-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	•••

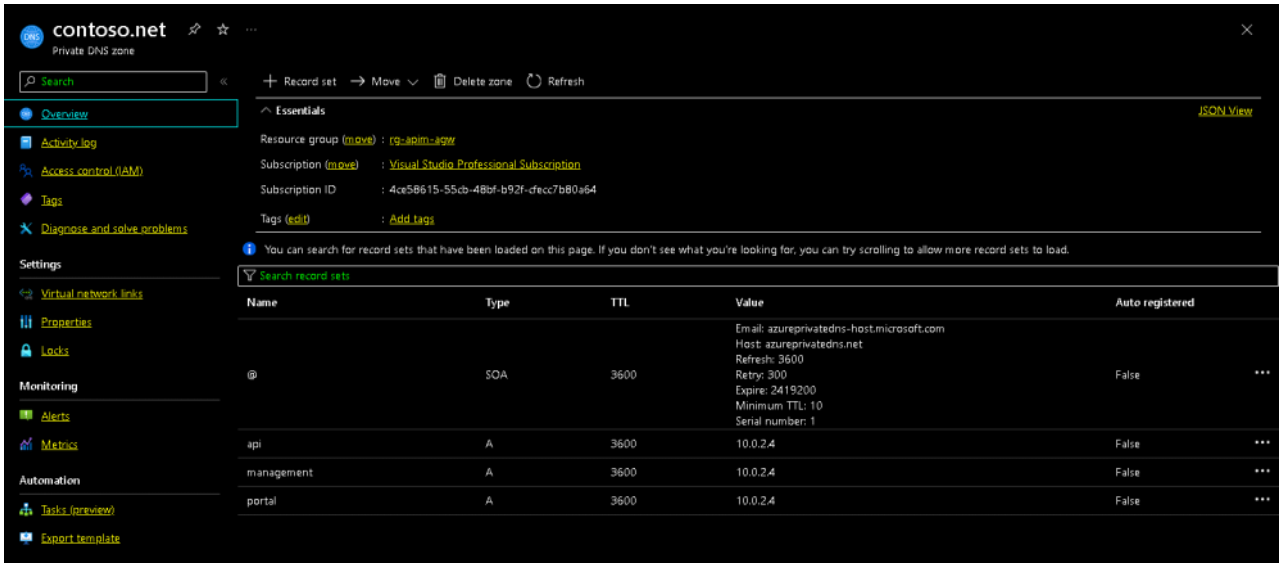
TRUSTED ROOT CERTIFICATES

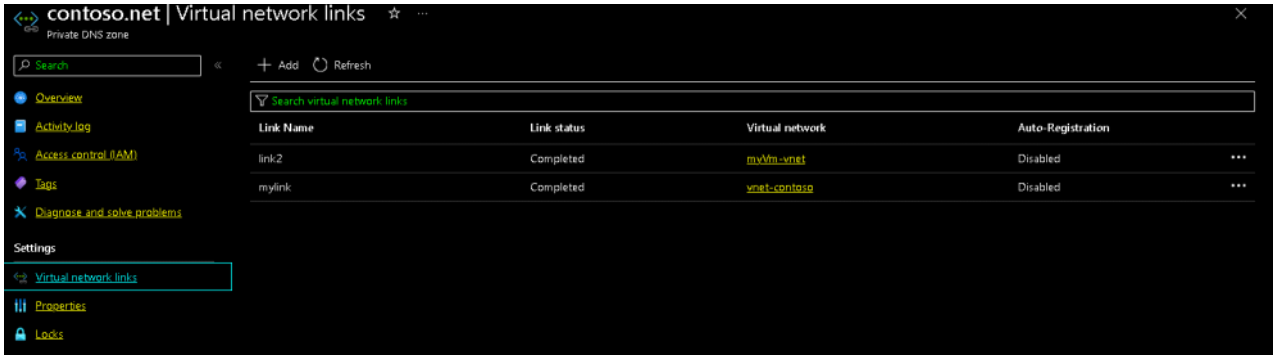
VIRTUAL NETWORK SUBNET , AZURE PRIVATE DNS ZONE AND NSG CONFIGURATION



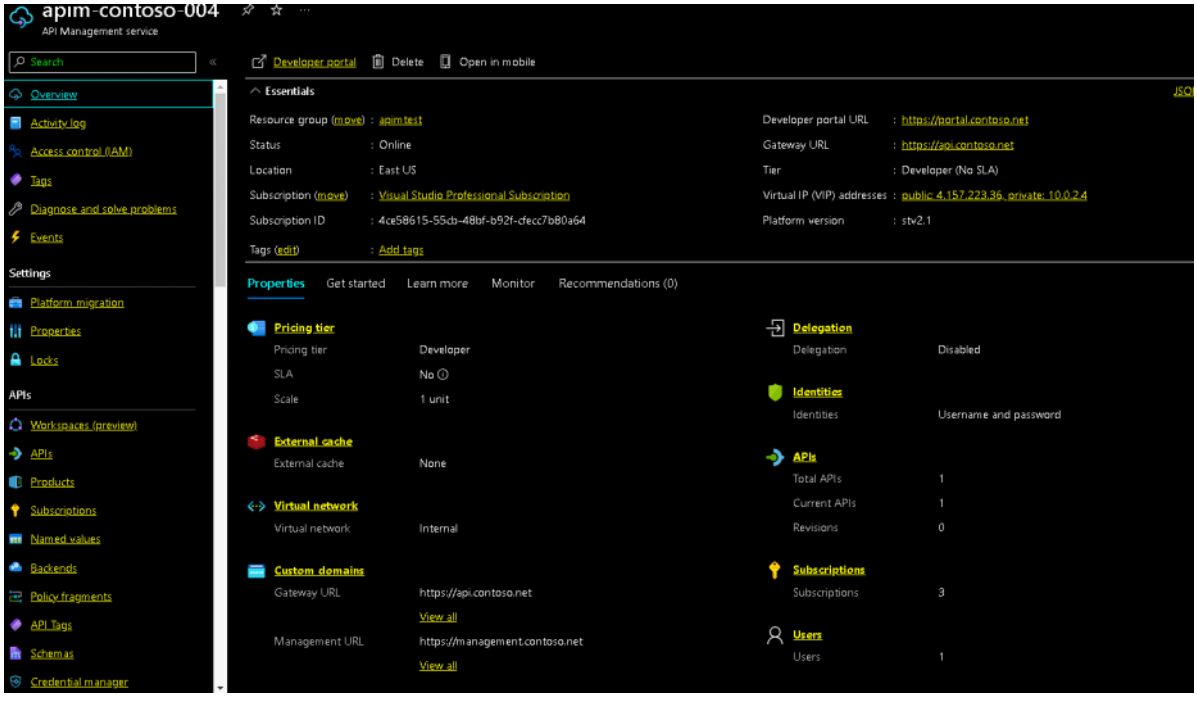

nsg-agw
Network security group
Search
Move
Delete
Refresh
Give feedback
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Settings
Inbound security rules
Outbound security rules
Network interfaces
Subnets
Properties
Locks
Monitoring
Alerts
Diagnostic settings
Logs
NSG flow logs
Essentials
JSON View
Resource group (move) : rg-apim-agw
Location : East US
Subscription (move) : Visual Studio Professional Subscription
Subscription ID : 4ce58615-55cb-48bf-b92f-cfecc7b80a64
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


Home > All resources > vnet-contoso | Subnets >
nsg-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 (preview)
Essentials
JSON View
Resource group (move) : rg-apim-agw
Location : East US
Subscription (move) : Visual Studio Professional Subscription
Subscription ID : 4ce58615-55cb-48bf-b92f-cfecc7b80a64
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





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 *
vnet-contoso

Subnet *
apimsubnet

Management public IP address
pip-apim

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-api.net subdomain (for example, contoso.azure-api.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-api.net	Built-in				•••
Gateway	api.contoso.net	Custom	Expiry: 3/5/2026, thumbprint:		✓	•••

Home > All resources > apim-contoso-004 | Custom domains >

Developer portal
API Management service

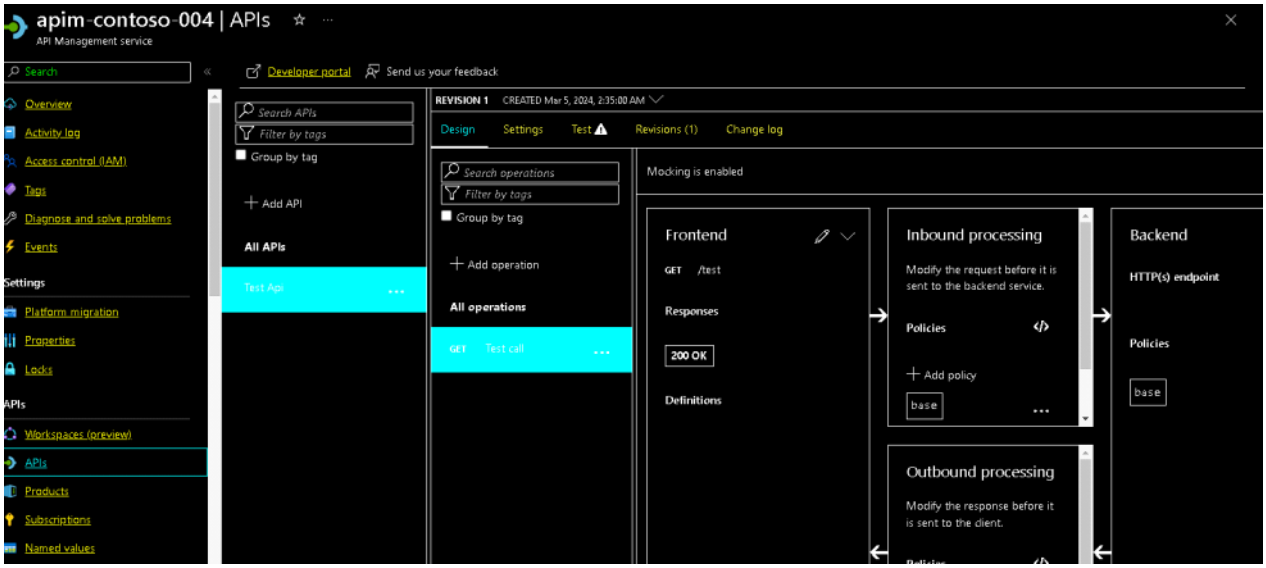
Type
Developer portal

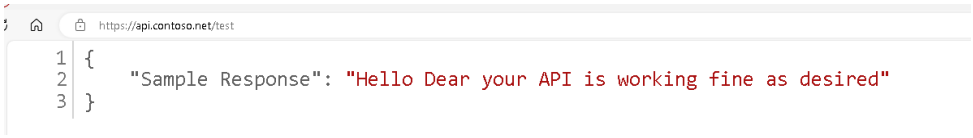
Hostname *
portal.contoso.net

Certificate
Custom

Certificate
Expiry: 3/5/2026, thumbprint: 5567ECAB8EA21453EB72CD8164E23683638B5719

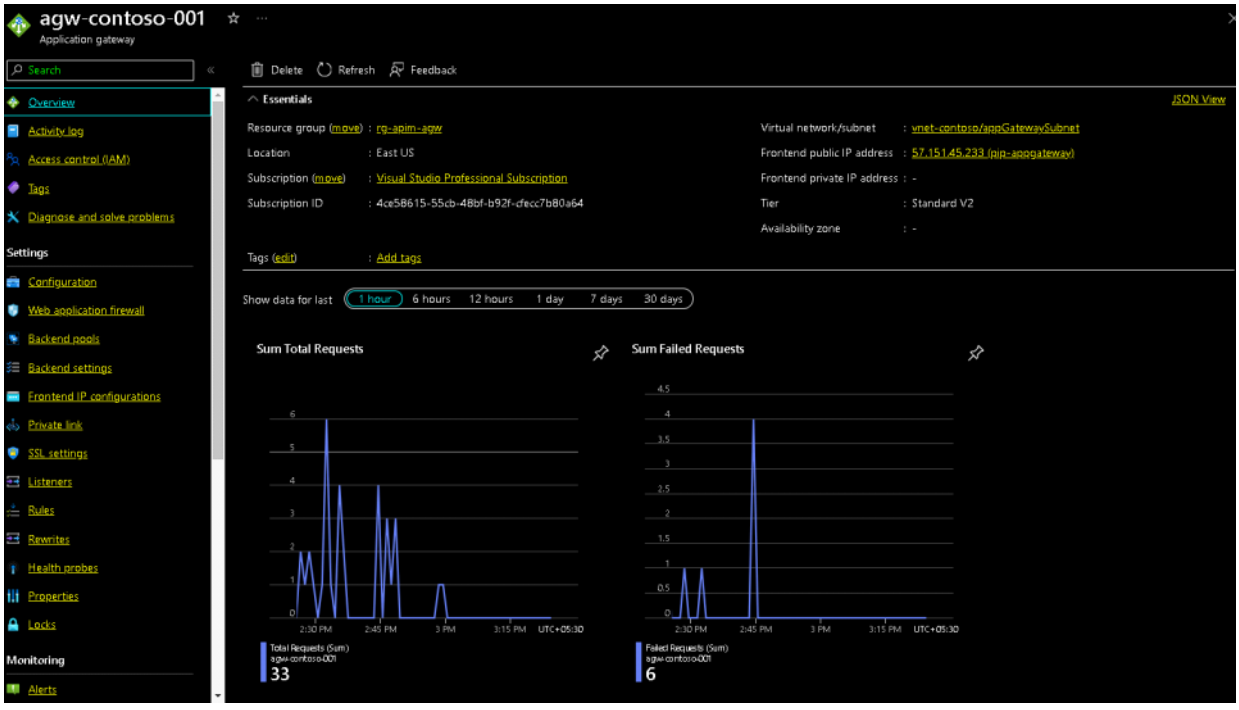
Change

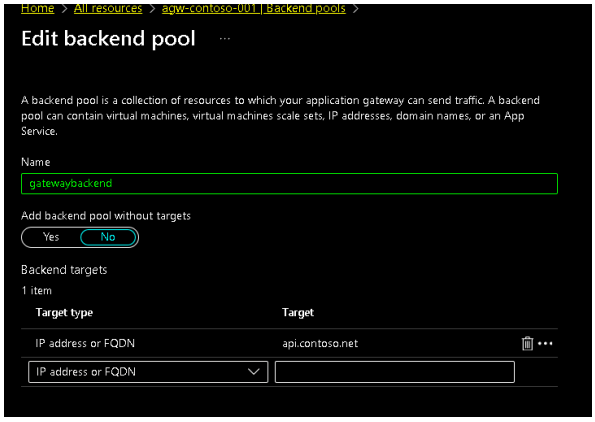




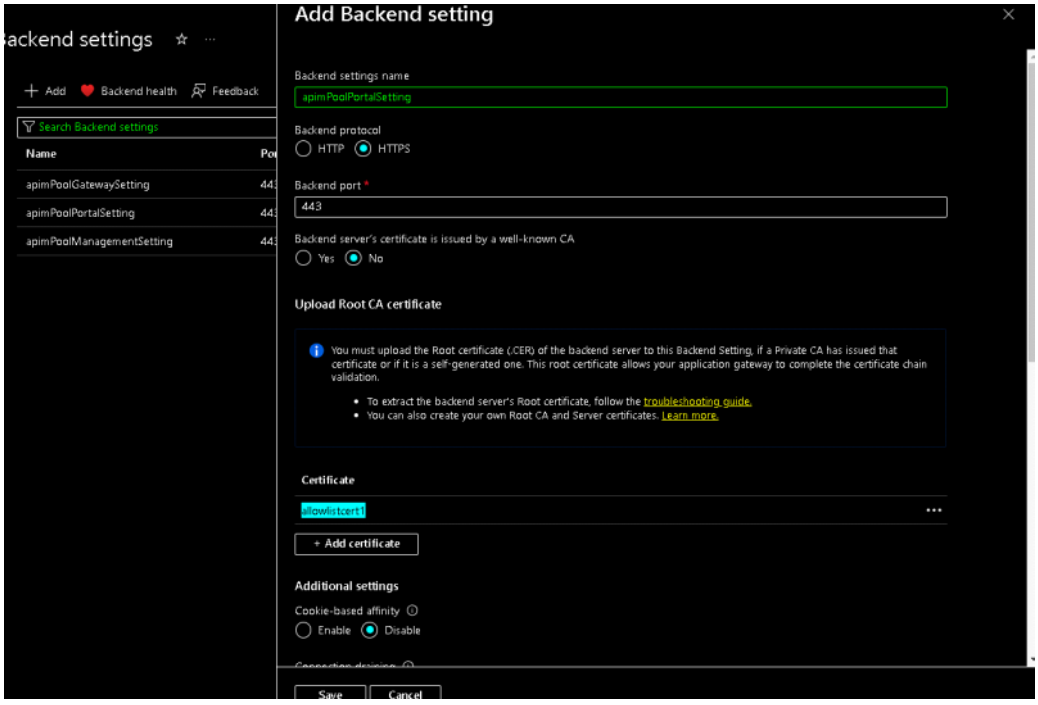

```
{
    "Sample Response": "Hello Dear your API is working fine as desired"
}
```

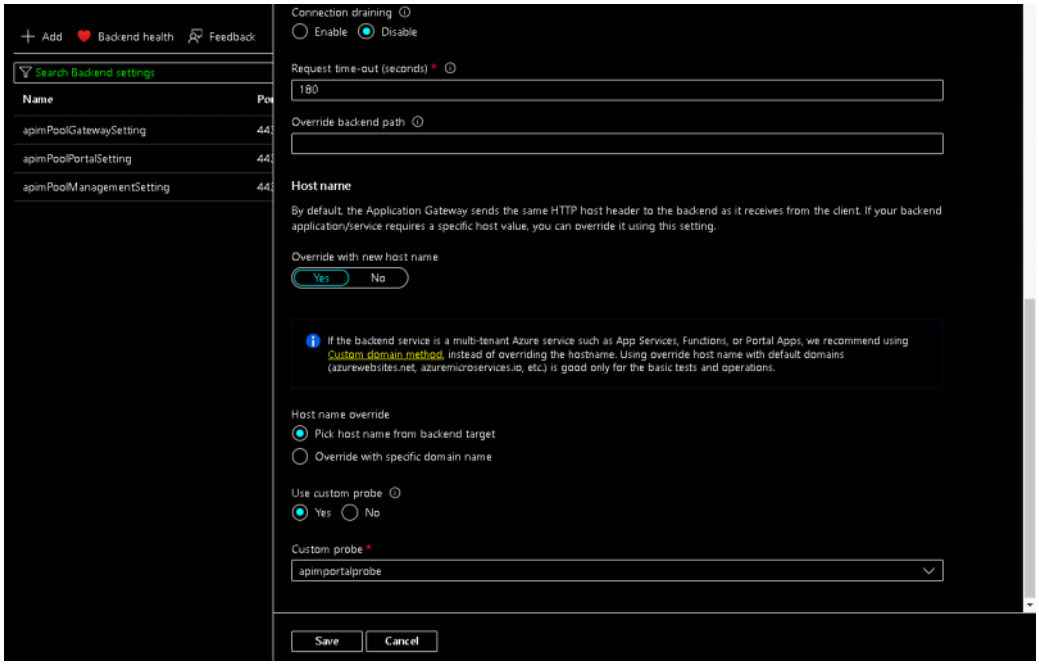
APPLICATION GATEWAY CONFIGURATION



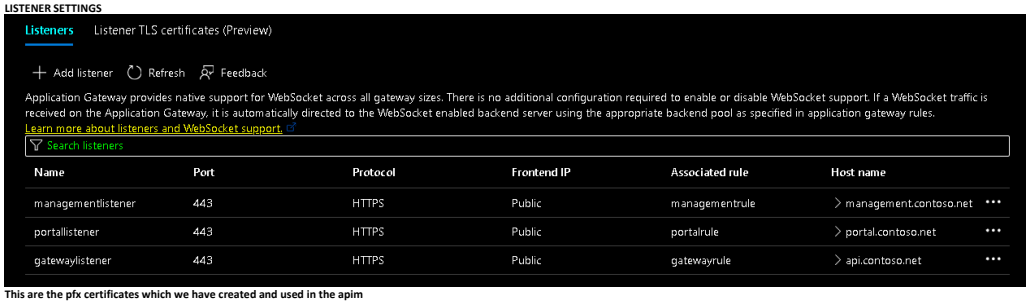


Here we have to refer to trusted root certificate and same settings we have to do for all three backend settings

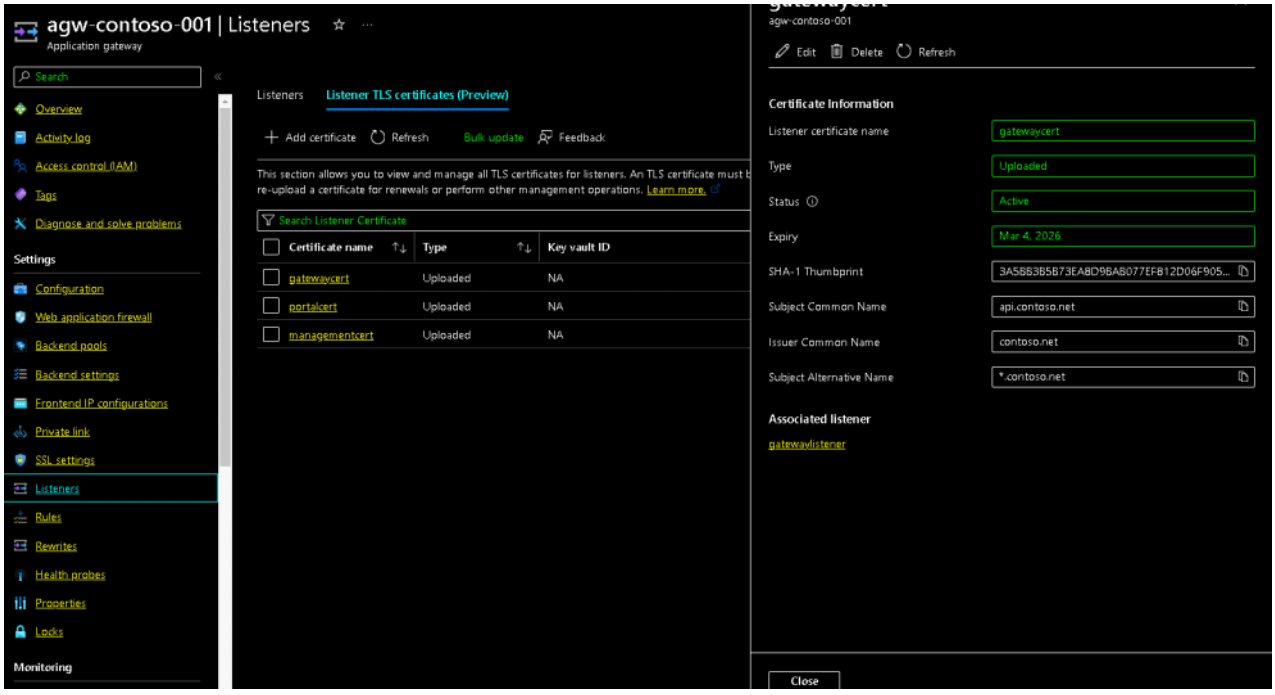


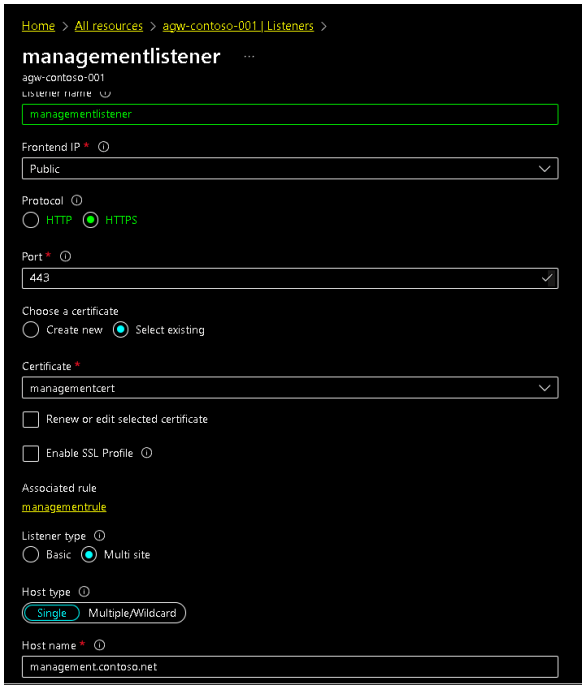


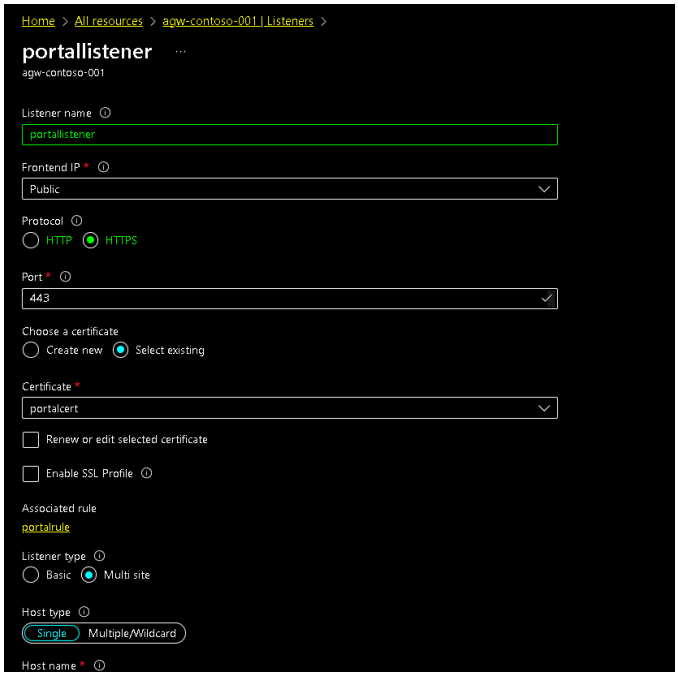
LISTENER SETTINGS



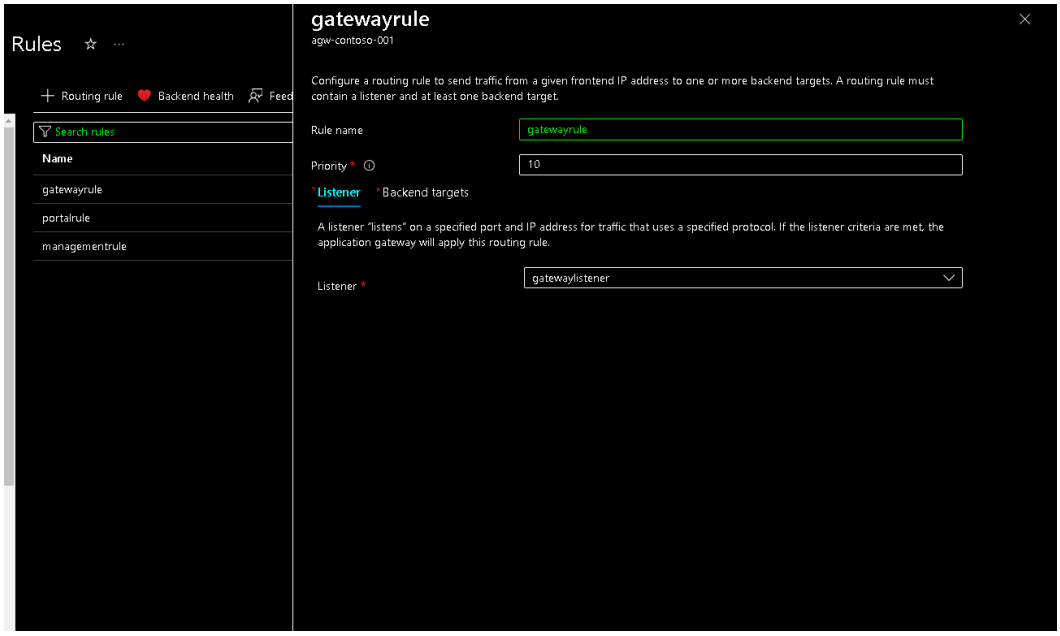
This are the pfx certificates which we have created and used in the apim

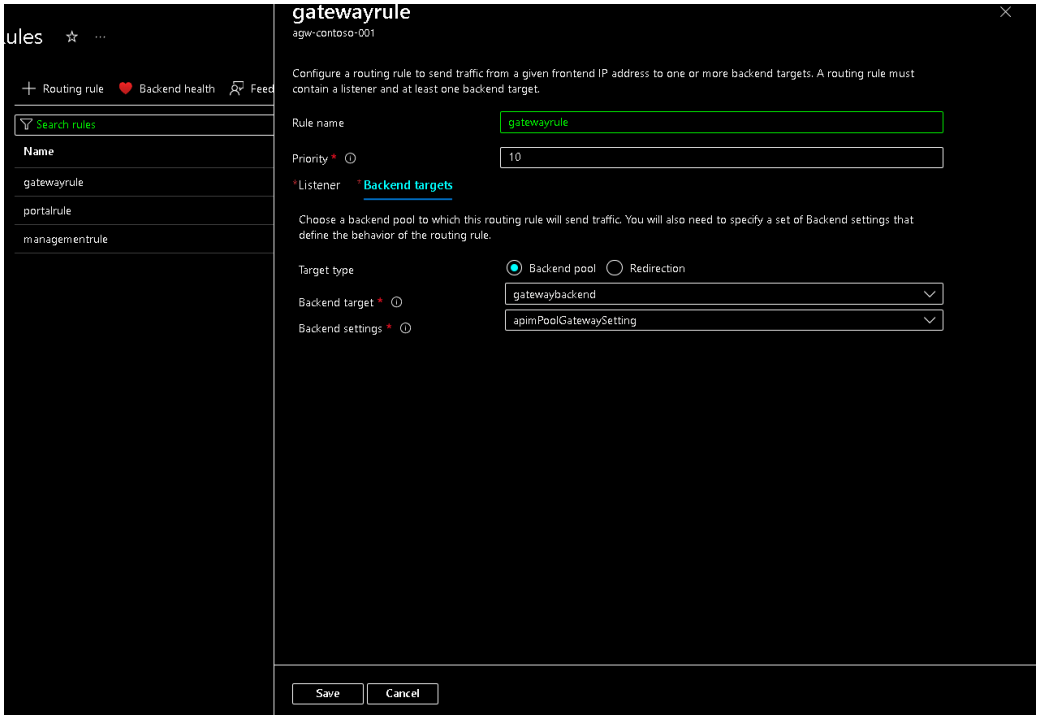






ROUTING RULES





Health Probe

apimmanagementprobe

agw-contoso-001

Name	apimmanagementprobe
Protocol *	○ HTTP ◉ HTTPS
Pick host name from backend settings	○ Yes ◉ No
Host *	management.contoso.net
Pick port from backend settings	◉ Yes ○ No
Path *	/ServiceStatus
Interval (seconds) *	60
Timeout (seconds) *	300
Unhealthy threshold *	8
Use probe matching conditions	○ Yes ◉ No
Backend settings	apimPoolManagementSetting

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

Test Discard

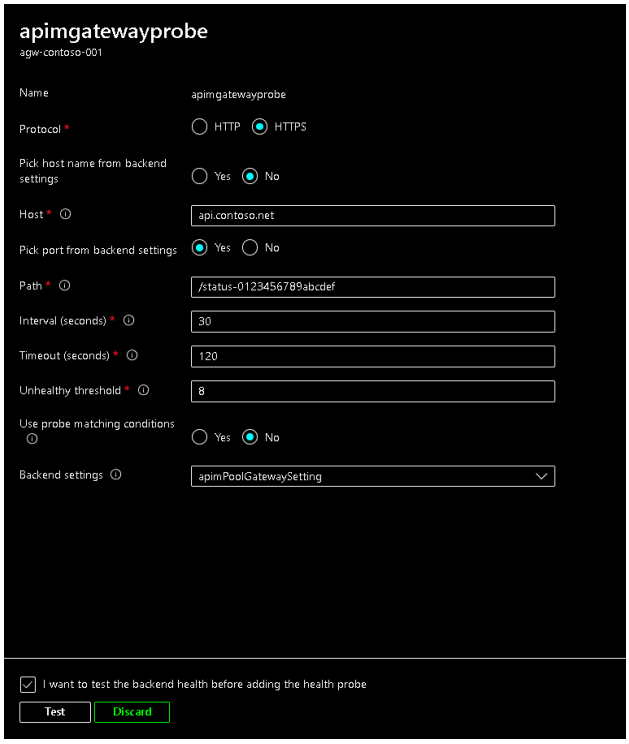
apimportalprobe

agw-contoso-001

Name	apimportalprobe
Protocol *	○ HTTP ◉ HTTPS
Pick host name from backend settings	○ Yes ◉ No
Host *	portal.contoso.net
Pick port from backend settings	◉ Yes ○ No
Path *	/signin
Interval (seconds) *	60
Timeout (seconds) *	300
Unhealthy threshold *	8
Use probe matching conditions	○ Yes ◉ No
Backend settings	apimPoolPortalSetting

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

Test Discard



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