Windows PowerShell Get-Help on Cmdlet 'Set-AzVirtualNetworkGateway'

PS:\>Get-HELP Set-AzVirtualNetworkGateway -Full

WARNING: The names of some imported commands from the module 'Microsoft.Azure.PowerShell.Cmdlets.Network' include unapproved verbs that might make them less discoverable.

To find the commands with unapproved verbs, run the Import-Module command again with the Verbose parameter. For a list of approved verbs, type Get-Verb.

NAME

Set-AzVirtualNetworkGateway

SYNOPSIS

Updates a virtual network gateway.

SYNTAX

Set-AzVirtualNetworkGateway [-AadAudienceId <System.String>] [-AadIssuerUri <System.String>] [-AadTenantUri <System.String>] [-AsJob] [-Asn <System.UInt32>]

[-BgpRouteTranslationForNat <System.Nullable`1[System.Boolean]>] [-ClientConnectionConfiguration

<Microsoft.Azure.Commands.Network.Models.PSClientConnectionConfiguration[]>] [-CustomRoute <System.String[]>] [-DefaultProfile

<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-DisableActiveActiveFeature] [-EnableActiveActiveFeature]

[-EnablePrivateIpAddress <System.Nullable`1[System.Boolean]>] [-GatewayDefaultSite

<Microsoft.Azure.Commands.Network.Models.PSLocalNetworkGateway>] [-GatewaySku
{Basic | Standard | HighPerformance | UltraPerformance | VpnGw1 | VpnGw2 | VpnGw3 | VpnGw4 | VpnGw5 |
VpnGw1AZ | VpnGw2AZ | VpnGw3AZ | VpnGw4AZ | VpnGw5AZ | ErGw1AZ |
ErGw2AZ | ErGw3AZ}] [-IpConfigurationBgpPeeringAddresses
<Microsoft.Azure.Commands.Network.Models.PSIpConfigurationBgpPeeringAddress[]>] [-NatRule
<Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGatewayNatRule[]>] [-PeerWeight <System.Int32>]
[-RadiusServerAddress <System.String>] [-RadiusServerList
<Microsoft.Azure.Commands.Network.Models.PSRadiusServer[]>] [-RadiusServerSecret
<System.Security.SecureString>] [-RemoveAadAuthentication] -Tag
<System.Collections.Hashtable> -VirtualNetworkGateway
<Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway> [-VirtualNetworkGatewayPolicyGroup
<Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGatewayPolicyGroup[]>] [-VpnAuthenticationType
{Certificate | Radius | AAD}] [-VpnClientAddressPool
<System.String[]>] [-VpnClientIpsecPolicy <Microsoft.Azure.Commands.Network.Models.PSIpsecPolicy[]>]
[-VpnClientProtocol {SSTP | IkeV2 | OpenVPN}]
[-VpnClientRevokedCertificates <Microsoft.Azure.Commands.Network.Models.PSVpnClientRevokedCertificate[]>]
[-VpnClientRootCertificates
<Microsoft.Azure.Commands.Network.Models.PSVpnClientRootCertificate[]>] [-Confirm] [-WhatIf]
[<CommonParameters>]

DESCRIPTION

The Set-AzVirtualNetworkGateway cmdlet updates a virtual network gateway.

PARAMETERS

-AadAudienceId <System.String>

P2S AAD authentication option:AadAudienceId.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-AadIssuerUri <System.String>

P2S AAD authentication option:AadIssuerUri.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-AadTenantUri <System.String>

P2S AAD authentication option:AadTenantUri.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Asn <System.UInt32>

The virtual network gateway's ASN, used to set up BGP sessions inside IPsec tunnels

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-BgpRouteTranslationForNat <System.Nullable`1[System.Boolean]>

This will enable and disable BgpRouteTranslationForNat on this VirtualNetworkGateway

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ClientConnectionConfiguration <Microsoft.Azure.Commands.Network.Models.PSClientConnectionConfiguration[]>

P2S Client Connection Configuration that assiociate between address and policy group

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-CustomRoute <System.String[]>

Custom routes AddressPool specified by customer

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with Azure.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DisableActiveActiveFeature <System.Management.Automation.SwitchParameter>

Flag to disable Active Active feature on virtual network gateway

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnableActiveActiveFeature <System.Management.Automation.SwitchParameter>

Flag to enable Active Active feature on virtual network gateway

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnablePrivateIpAddress <System.Nullable`1[System.Boolean]>

Flag to enable Active Active feature on virtual network gateway

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-GatewayDefaultSite <Microsoft.Azure.Commands.Network.Models.PSLocalNetworkGateway>

The default site to use for force tunneling. If a default site is specified, all internet traffic from the gateway's vnet is routed to that site.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-GatewaySku <System.String>

The virtual network gateway's SKU

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-IpConfigurationBgpPeeringAddresses <Microsoft.Azure.Commands.Network.Models.PSIpConfigurationBgpPeeringAddress[]>

The BgpPeeringAddresses for Virtual network gateway bgpsettings.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-NatRule <Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGatewayNatRule[]>

The NatRules for Virtual network gateway.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PeerWeight <System.Int32>

The weight added to routes learned over BGP from this virtual network gateway

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-RadiusServerAddress <System.String>

P2S External Radius server address.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-RadiusServerList <Microsoft.Azure.Commands.Network.Models.PSRadiusServer[]>

P2S multiple external Radius servers.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-RadiusServerSecret <System.Security.SecureString>

P2S External Radius server secret.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-RemoveAadAuthentication <System.Management.Automation.SwitchParameter>

Flag to remove AAD authentication for P2S client from virtual network gateway.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Tag <System.Collections.Hashtable>

P2S External Radius server address.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-VirtualNetworkGateway <Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway>

The virtual network gateway object to base modifications off of. This can be retrieved using Get-AzVirtualNetworkGateway

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-VirtualNetworkGatewayPolicyGroup <Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGatewayPolicyGroup[]>

P2S policy group added to this gateway

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnAuthenticationType <System.String[]>

The list of P2S VPN client authentication types.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnClientAddressPool <System.String[]>

The address space to allocate VPN client IP addresses from. This should not overlap with virtual network or on-premise ranges.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnClientIpsecPolicy <Microsoft.Azure.Commands.Network.Models.PSIpsecPolicy[]>

A list of IPSec policies for P2S VPN client tunneling protocols.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnClientProtocol <System.String[]>

A list of P2S VPN client tunneling protocols

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnClientRevokedCertificates <Microsoft.Azure.Commands.Network.Models.PSVpnClientRevokedCertificate[]>

A list of revoked VPN client certificates. A VPN client presenting a certificate that matches one of these will be told to go away.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-VpnClientRootCertificates <Microsoft.Azure.Commands.Network.Models.PSVpnClientRootCertificate[]>

A list of VPN client root certificates to use for VPN client authentication. Connecting VPN clients must present certificates generated from one of these root

certificates.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway

System.String

Microsoft.Azure.Commands.Network.Models.PSLocalNetworkGateway

System.String[]

Microsoft.Azure.Commands.Network.Models.PSVpnClientRootCertificate[]

Microsoft.Azure.Commands.Network.Models.PSVpnClientRevokedCertificate[]

Microsoft.Azure.Commands.Network.Models.PSIpsecPolicy[]

System.UInt32

System.Int32

Microsoft.Azure.Commands.Network.Models.PSIpConfigurationBgpPeeringAddress[]

System.Security.SecureString

Microsoft.Azure.Commands.Network.Models.PSRadiusServer[]

Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGatewayNatRule[]

OUTPUTS

Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway

NOTES

------ Example 1: Update a virtual network gateway's ASN ------

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -Asn 1337
```

The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command updates the virtual network gateway stored in variable $Gateway. The command also sets the ASN to 1337.

--- Example 2: Add IPsec policy to a virtual network gateway ---

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
  $vpnclientipsecpolicy = New-AzVpnClientIpsecPolicy -IpsecEncryption AES256 -IpsecIntegrity SHA256 -SALifeTime 86472 -SADataSize 429497 -IkeEncryption AES256
-IkeIntegrity SHA256 -DhGroup DHGroup2 -PfsGroup None
```

Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientIpsecPolicy $vpnclientipsecpolicy

The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command creates the Vpn ipsec policy object as per specified ipsec parameters. The third command updates the virtual network gateway stored in variable

$Gateway. The command also sets the custom vpn ipsec policy specified in the $vpnclientipsecpolicy object on Virtual network gateway.

Example 3: Add/Update Tags to an existing virtual network gateway

$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"

Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -Tag @{ testtagKey="SomeTagKey"; testtagValue="SomeKeyValue" }

Name : Gateway001

ResourceGroupName : ResourceGroup001

Location : westus

Id :

/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGateways/Gateway001

Etag : W/"00000000-0000-0000-0000-000000000000"

ResourceGuid : 00000000-0000-0000-0000-000000000000

ProvisioningState : Succeeded

Tags :

Name Value

========== ==========

testtagValue SomeKeyValue

testtagKey SomeTagKey

IpConfigurations : [

```
                    {
                      "PrivateIpAllocationMethod": "Dynamic",
                      "Subnet": {
                        "Id":

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Networ
k/virtualNetworks/MyVnet/subnets/GatewaySubnet"
                      },
                      "PublicIpAddress": {
                        "Id":

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Networ
k/publicIPAddresses/Gateway001Ip"
                      },
                      "Name": "vng1ipConfig",
                      "Etag": "W/\"00000000-0000-0000-0000-000000000000\"",
                                                                                                    "Id":
"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Networ
k/virtualNetworkGatew
  ays/Gateway001/ipConfigurations/Gateway001IpConfig"
                    }
                  ]
  GatewayType           : Vpn
  VpnType               : RouteBased
  EnableBgp             : False
  ActiveActive          : False
  GatewayDefaultSite    : null
  Sku                   : {
                    "Capacity": 2,
                    "Name": "VpnGw1",
                    "Tier": "VpnGw1"
                  }
  VpnClientConfiguration : null
```

```
BgpSettings          : {
                         "Asn": 65515,
                         "BgpPeeringAddress": "1.2.3.4",
                         "PeerWeight": 0
                       }
```

The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command updates the virtual network gateway Gateway01 with the tags @{ testtagKey="SomeTagKey"; testtagValue="SomeKeyValue" }.

Example 4: Add/Update AAD authentication configuration for VpnClient of an existing virtual network gateway

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
                                Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -AadTenantUri
"https://login.microsoftonline.com/0ab2c4f4-81e6-44cc-a0b2-b3a47a1443f4" -AadIssuerUri
                                        "https://sts.windows.net/0ab2c4f4-81e6-44cc-a0b2-b3a47a1443f4/" -AadAudienceId
"a21fce82-76af-45e6-8583-a08cb3b956f9"

Name                 : Gateway001
ResourceGroupName    : ResourceGroup001
Location             : westus
Id                   :

/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
virtualNetworkGateways/Gateway001
Etag                 : W/"00000000-0000-0000-0000-000000000000"
ResourceGuid         : 00000000-0000-0000-0000-000000000000
ProvisioningState    : Succeeded
Tags                 :
                       Name         Value
                       ============ ============
```

testtagValue SomeKeyValue

testtagKey SomeTagKey

IpConfigurations : [

{

"PrivateIpAllocationMethod": "Dynamic",

"Subnet": {

"Id":

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworks/MyVnet/subnets/GatewaySubnet"

},

"PublicIpAddress": {

"Id":

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Network/publicIPAddresses/Gateway001Ip"

},

"Name": "vng1ipConfig",

"Etag": "W/\"00000000-0000-0000-0000-000000000000\"",

"Id":

"/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGateways/Gateway001/ipConfigurations/Gateway001IpConfig"

}

]

GatewayType : Vpn

VpnType : RouteBased

EnableBgp : False

ActiveActive : False

GatewayDefaultSite : null

Sku : {

"Capacity": 2,

```
                    "Name": "VpnGw1",
                    "Tier": "VpnGw1"
                  }
  vpnClientConfiguration : {
                "vpnClientProtocols": [
    "OpenVPN"
    ],

                "vpnClientAddressPool": {
                "addressPrefixes": [
                   "101.10.0.0/16"
                ]
              },
    "vpnClientRootCertificates": "",
                "vpnClientRevokedCertificates": "",

                "radiusServerAddress": "",
                "radiusServerSecret": "",
    "aadTenantUri": "https://login.microsoftonline.com/0ab2c4f4-81e6-44cc-a0b2-b3a47a1443f4\",
    "aadAudienceId": "a21fce82-76af-45e6-8583-a08cb3b956g9\",
    "aadIssuerUri": "https://sts.windows.net/0ab2c4f4-81e6-44cc-a0b2-b3a47a1443f4/\"
            },
  BgpSettings          : {
                    "Asn": 65515,
                    "BgpPeeringAddress": "1.2.3.4",
                    "PeerWeight": 0
                  }
```


Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientRootCertificates $rootCert -RemoveAadAuthentication

The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command updates the virtual network gateway Gateway01 with the AAD authentication configurations params:aadTenantUri, aadAudienceId, aadIssuerUri for

VpnClient. The third command removes the AAD authentication configuration from VpnClient of virtual network gateway.

Example 5: Add/Update IpConfigurationBgpPeeringAddresses to an existing virtual network gateway

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
                                                                    $ipconfigurationId1 =
'/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
virtualNetworkGateways/Gateway00
1/ipConfigurations/default'
$addresslist1 = @('169.254.21.25')
        $gw1ipconfBgp1 = New-AzIpConfigurationBgpPeeringAddressObject -IpConfigurationId $ipconfigurationId1
-CustomAddress $addresslist1
          Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -IpConfigurationBgpPeeringAddresses
$gw1ipconfBgp1

Name                 : Gateway001
ResourceGroupName    : ResourceGroup001
Location             : westcentralus
Id                   :

/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/v
irtualNetworkGateways/Gateway001
Etag                 : W/"a08f13d3-6106-44e0-9127-e35e6f9793d5"
ResourceGuid         : 30993429-a1ed-42ca-9862-9156b013626e
ProvisioningState    : Succeeded
Tags                 :
IpConfigurations     : [
                       {
                         "PrivateIpAllocationMethod": "Dynamic",
                         "Subnet": {
```

```
                    "Id":
"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworks/newApipaNet/subnets/GatewaySubnet"
                    },
                    "PublicIpAddress": {
                      "Id":
"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/publicIPAddresses/newapipaip"
                    },
                    "Name": "default",
                    "Etag": "W/\"a08f13d3-6106-44e0-9127-e35e6f9793d5\"",
                                                                                    "Id":
"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGatew
  ays/Gateway001/ipConfigurations/default"
                  }
                ]
  GatewayType           : Vpn
  VpnType               : RouteBased
  EnableBgp             : False
  ActiveActive          : False
  GatewayDefaultSite    : null
  Sku                 : {
                    "Capacity": 2,
                    "Name": "VpnGw1",
                    "Tier": "VpnGw1"
                  }
  VpnClientConfiguration : null
  BgpSettings           : {
                    "Asn": 65515,
                    "BgpPeeringAddress": "10.1.255.30",
```

```
    "PeerWeight": 0,
    "BgpPeeringAddresses": [
      {
                                                                                    "IpconfigurationId":
"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
vi
  rtualNetworkGateways/Gateway001/ipConfigurations/default",
        "DefaultBgpIpAddresses": [
          "10.1.255.30"
        ],
        "CustomBgpIpAddresses": [
          "169.254.21.55"
        ],
        "TunnelIpAddresses": [
          "13.78.146.151"
        ]
      }
    ]
  }
```


The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command assigns the value of virtual network gateway Gateway01 IpConfiguration Id into variable ipconfigurationId1. The third command assigns the address list

into addresslist1. The fourth command created a PSIpConfigurationBgpPeeringAddress object. The fifth command set this new created PSIpConfigurationBgpPeeringAddress

to IpConfigurationBgpPeeringAddresses and update the gateway.

Example 6: Update/Remove CustomAddress to an existing IpConfigurationBgpPeeringAddresses of virtual network gateway

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
```

$ipconfigurationId1 =
'/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
virtualNetworkGateways/Gateway00
1/ipConfigurations/default'
$addresslist1 = @()
$gw1ipconfBgp1 = New-AzIpConfigurationBgpPeeringAddressObject -IpConfigurationId $ipconfigurationId1
-CustomAddress $addresslist1
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -IpConfigurationBgpPeeringAddresses
$gw1ipconfBgp1

Name : Gateway001
ResourceGroupName : ResourceGroup001
Location : westcentralus
Id :

/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/v
irtualNetworkGateways/Gateway001
Etag : W/"a08f13d3-6106-44e0-9127-e35e6f9793d5"
ResourceGuid : 30993429-a1ed-42ca-9862-9156b013626e
ProvisioningState : Succeeded
Tags :
IpConfigurations : [
{
"PrivateIpAllocationMethod": "Dynamic",
"Subnet": {
"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
virtualNetworks/newApipaNet/subnets/GatewaySubnet"
},
"PublicIpAddress": {
"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/publicIPAddresses/newapipaip"
},
"Name": "default",
"Etag": "W/\"a08f13d3-6106-44e0-9127-e35e6f9793d5\"",
"Id": "/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGatew
ays/Gateway001/ipConfigurations/default"
}
]
GatewayType : Vpn
VpnType : RouteBased
EnableBgp : False
ActiveActive : False
GatewayDefaultSite : null
Sku : {
"Capacity": 2,
"Name": "VpnGw1",
"Tier": "VpnGw1"
}
VpnClientConfiguration : null
BgpSettings : {
"Asn": 65515,
"BgpPeeringAddress": "10.1.255.30",
"PeerWeight": 0,
"BgpPeeringAddresses": [
{
"IpconfigurationId": "/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/vi
rtualNetworkGateways/Gateway001/ipConfigurations/default",
"DefaultBgpIpAddresses": [

```
      "10.1.255.30"
    ],
    "CustomBgpIpAddresses": [],
    "TunnelIpAddresses": [
      "13.78.146.151"
    ]
   }
  ]
}
```


The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command assigns the value of virtual network gateway Gateway01 IpConfiguration Id into variable ipconfigurationId1. The third command assigns the address list

into addresslist1. The fourth command created a PSIpConfigurationBgpPeeringAddress object. The fifth command set this new created PSIpConfigurationBgpPeeringAddress

to IpConfigurationBgpPeeringAddresses and update the gateway.

Example 7: Add/Update NatRules to an existing virtual network gateway

```
$Gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"
$vngNatRules = $Gateway.NatRules
$natRule = New-AzVirtualNetworkGatewayNatRule -Name "natRule1" -Type "Static" -Mode "IngressSnat" -InternalMapping @("25.0.0.0/16") -ExternalMapping @("30.0.0.0/16")
$vngNatRules.Add($natrule)
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -NatRule $vngNatRules.NatRules -BgpRouteTranslationForNat $true

Name                 : Gateway001
ResourceGroupName    : ResourceGroup001
Location             : westcentralus
Id                   :
```

/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGateways/Gateway001

Etag : W/"a08f13d3-6106-44e0-9127-e35e6f9793d5"

ResourceGuid : 30993429-a1ed-42ca-9862-9156b013626e

ProvisioningState : Succeeded

Tags :

IpConfigurations : [

{

"PrivateIpAllocationMethod": "Dynamic",

"Subnet": {

"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworks/newApipaNet/subnets/GatewaySubnet"

},

"PublicIpAddress": {

"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/publicIPAddresses/newapipaip"

},

"Name": "default",

"Etag": "W/\"a08f13d3-6106-44e0-9127-e35e6f9793d5\"",

"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/virtualNetworkGatew

ays/Gateway001/ipConfigurations/default"

}

]

GatewayType : Vpn

VpnType : RouteBased

EnableBgp : False

```
  ActiveActive          : False
  GatewayDefaultSite    : null
  Sku                   : {
                          "Capacity": 2,
                          "Name": "VpnGw1",
                          "Tier": "VpnGw1"
                        }
  VpnClientConfiguration : null
  BgpSettings           : {
                          "Asn": 65515,
                          "BgpPeeringAddress": "10.1.255.30",
                          "PeerWeight": 0,
                          "BgpPeeringAddresses": [
                            {
                                                                                "IpconfigurationId":
"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
vi
  rtualNetworkGateways/Gateway001/ipConfigurations/default",
                              "DefaultBgpIpAddresses": [
                                "10.1.255.30"
                              ],
                              "CustomBgpIpAddresses": [
                                "169.254.21.55"
                              ],
                              "TunnelIpAddresses": [
                                "13.78.146.151"
                              ]
                            }
                          ]
                        }
  NatRules              : [
                          {
                            "VirtualNetworkGatewayNatRulePropertiesType": "Static",
```

```
"Mode": "IngressSnat",
"InternalMappings": [
  {
    "AddressSpace": "25.0.0.0/16"
  }
],
"ExternalMappings": [
  {
    "AddressSpace": "30.0.0.0/16"
  }
],
"ProvisioningState": "Succeeded",
"Name": "natRule1",
"Etag": "W/\"5150d788-e165-42ba-99c4-8138a545fce9\"",
"Id":

"/subscriptions/59ac12a6-f2b7-46d4-af3d-98ba9d9dbd92/resourceGroups/ResourceGroup001/providers/Microsoft.Network/
virtualNetworkGateways/Gateway001/natRules/natRule1"
  }
]
EnableBgpRouteTranslationForNat : True
```


The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command assigns the existing natrules into variable vngNatRules. The third command assigns the value newly created PSVirtualNetworkGatewayNatRule object

natrule into variable natRule. The fourth command add this PSVirtualNetworkGatewayNatRule object into vngNatRules list. The fifth command set this new created

PSVirtualNetworkGatewayNatRule to NatRules of gateway and update the gateway.

Example 8: Delete multiple expired VpnClientRootCertificates of an existing virtual network gateway

```
$Gateway=Get-AzVirtualNetworkGateway -ResourceGroupName "ResourceGroup001" -Name "Gateway001"

$rootCerts=$Gateway.VpnClientConfiguration.VpnClientRootCertificates

$rootCerts.Count
$rootCerts[0]
$rootCerts[1]
$rootCerts.Remove($rootCerts[1])

$Gateway1 = Set-AzVirtualNetworkGateway -VirtualNetworkGateway $Gateway -VpnClientRootCertificates $rootCerts
```

The first command gets a virtual network gateway named Gateway01 that belongs to resource group ResourceGroup001 and stores it to the variable named $Gateway The

second command gets all the root certificates on VirtualNetworkGateway and save it to another variable $rootCerts The third command shows total existing root certs on

VirtualNetworkGateway. The forth & fifth commands print root certificates at those corresponding indices for customer to see which ones they want to delete. The

sixth command removes expired root certificate by using that index e.g. here 1. Repeat same steps to remove multiple expired certificates from variable: $rootCerts

The seventh command updates VirtualNetworkGateway to set valid root certificates i.e. certificates that exists in variable: $rootCerts

Example 9: Configure an ExpressRoute virtual network gateway to allow communication over ExpressRoute with other ExpressRoute virtual network gateways in Virtual Wan

networks.

```
# Option 1 - Retrieve the gateway object, modify the property and save the changes.
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
$gateway.AllowVirtualWanTraffic = $true
$gateway = Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway

# Option 2 - Use the cmdlet switch
```

```
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway -AllowVirtualWanTraffic $true
```

In both cases, the first command retrieves the gateway. You may then either modify the property directly on the object and persist it, or you may use the switch on
the Set-AzVirtualNetworkGateway cmdlet.

Example 10: Configure an ExpressRoute virtual network gateway to block communication over ExpressRoute with other ExpressRoute virtual network gateways in Virtual Wan
networks.

```
# Option 1 - Retrieve the gateway object, modify the property and save the changes.
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
$gateway.AllowVirtualWanTraffic = $false
$gateway = Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway

# Option 2 - Use the cmdlet switch
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway -AllowVirtualWanTraffic $false
```

In both cases, the first command retrieves the gateway. You may then either modify the property directly on the object and persist it, or you may use the switch on
the Set-AzVirtualNetworkGateway cmdlet.

Example 11: Configure an ExpressRoute virtual network gateway to allow communication over ExpressRoute with other ExpressRoute virtual network gateways in other VNets.

```
# Option 1 - Retrieve the gateway object, modify the property and save the changes.
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
$gateway.AllowRemoteVnetTraffic = $true
$gateway = Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway
```

```
# Option 2 - Use the cmdlet switch
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway -AllowRemoteVnetTraffic $true
```

In both cases, the first command retrieves the gateway. You may then either modify the property directly on the object and persist it, or you may use the switch on
the Set-AzVirtualNetworkGateway cmdlet.

Example 12: Configure an ExpressRoute virtual network gateway to block communication over ExpressRoute with other virtual network gateways in other VNets.

```
# Option 1 - Retrieve the gateway object, modify the property and save the changes.
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
$gateway.AllowRemoteVnetTraffic = $false
$gateway = Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway

# Option 2 - Use the cmdlet switch
$gateway = Get-AzVirtualNetworkGateway -ResourceGroupName "resourceGroup001" -Name "gateway001"
Set-AzVirtualNetworkGateway -VirtualNetworkGateway $gateway -AllowRemoteVnetTraffic $false
```

In both cases, the first command retrieves the gateway. You may then either modify the property directly on the object and persist it, or you may use the switch on
the Set-AzVirtualNetworkGateway cmdlet.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.network/set-azvirtualnetworkgateway