



Windows PowerShell Get-Help on Cmdlet 'Add-AzVirtualNetworkPeering'

PS: \>Get-HELP Add-AzVirtualNetworkPeering -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

Add-AzVirtualNetworkPeering

SYNOPSIS

Creates a peering between two virtual networks.

SYNTAX

```
Add-AzVirtualNetworkPeering [-AllowForwardedTraffic] [-AllowGatewayTransit] [-AsJob] [-BlockVirtualNetworkAccess]
[-DefaultProfile
    <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] -Name
<System.String> -RemoteVirtualNetworkId <System.String>
    [-UseRemoteGateways] -VirtualNetwork <Microsoft.Azure.Commands.Network.Models.PSVirtualNetwork>
[<CommonParameters>]
```

DESCRIPTION

The Add-AzVirtualNetworkPeering cmdlet creates a peering between two virtual networks.

PARAMETERS

-AllowForwardedTraffic <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet allows the forwarded traffic from the virtual machines in the remote virtual network.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-AllowGatewayTransit <System.Management.Automation.SwitchParameter>

Flag to allow gatewayLinks be used in remote virtual network's link to this virtual network

Required? false

Position? named

Default value False

Accept pipeline input? False

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

-BlockVirtualNetworkAccess <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet blocks the virtual machines in the linked virtual network space to access all the virtual machines in local virtual network space.

Required?	false
Position?	named
Default value	False
Accept pipeline input?	False
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

-Name <System.String>

Specifies the name of the virtual network peering.

Required?	true
Position?	named
Default value	None
Accept pipeline input?	False
Accept wildcard characters?	false

-RemoteVirtualNetworkId <System.String>

Specifies the ID of the remote virtual network.

Required?	true
Position?	named
Default value	None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-UseRemoteGateways <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet allows remote gateways on this virtual network.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-VirtualNetwork <Microsoft.Azure.Commands.Network.Models.PSVirtualNetwork>

Specifies the parent virtual network.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

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.PSVirtualNetwork

System.String

OUTPUTS

Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkPeering

NOTES

Example 1: Create a peering between two virtual networks in the same region

Variables for common values used throughout the script.

\$rgName='myResourceGroup'

\$location='eastus'

Create a resource group.

New-AzResourceGroup -Name \$rgName -Location \$location

Create virtual network 1.

\$vnet1 = New-AzVirtualNetwork -ResourceGroupName \$rgName -Name 'myVnet1' -AddressPrefix '10.0.0.0/16' -Location
\$location

Create virtual network 2.

\$vnet2 = New-AzVirtualNetwork -ResourceGroupName \$rgName -Name 'myVnet2' -AddressPrefix '10.1.0.0/16' -Location
\$location

Peer VNet1 to VNet2.

Add-AzVirtualNetworkPeering -Name 'myVnet1ToMyVnet2' -VirtualNetwork \$vnet1 -RemoteVirtualNetworkId \$vnet2.Id

Peer VNet2 to VNet1.

```
Add-AzVirtualNetworkPeering -Name 'myVnet2ToMyVnet1' -VirtualNetwork $vnet2 -RemoteVirtualNetworkId $vnet1.Id
```

Note that a peering link must be created from vnet1 to vnet2 and vice versa in order for peering to work.

Example 2: Create a peering between two virtual networks in different regions

```
# Variables for common values used throughout the script.
```

```
$rgName='myResourceGroup'
```

```
# Create a resource group.
```

```
New-AzResourceGroup -Name $rgName -Location westcentralus
```

```
# Create virtual network 1.
```

```
$vnet1 = New-AzVirtualNetwork -ResourceGroupName $rgName -Name 'myVnet1' -AddressPrefix '10.0.0.0/16' -Location  
westcentralus
```

```
# Create virtual network 2.
```

```
$vnet2 = New-AzVirtualNetwork -ResourceGroupName $rgName -Name 'myVnet2' -AddressPrefix '10.1.0.0/16' -Location  
canadacentral
```

```
# Peer VNet1 to VNet2.
```

```
Add-AzVirtualNetworkPeering -Name 'myVnet1ToMyVnet2' -VirtualNetwork $vnet1 -RemoteVirtualNetworkId $vnet2.Id
```

```
# Peer VNet2 to VNet1.
```

```
Add-AzVirtualNetworkPeering -Name 'myVnet2ToMyVnet1' -VirtualNetwork $vnet2 -RemoteVirtualNetworkId $vnet1.Id
```

Here 'myVnet1' in US West Central is peered with 'myVnet2' in Canada Central.

RELATED LINKS

Get-AzVirtualNetwork

Get-AzVirtualNetworkPeering

Remove-AzVirtualNetworkPeering

Set-AzVirtualNetworkPeering

Sync-AzVirtualNetworkPeering