Windows PowerShell Get-Help on Cmdlet 'New-AzFirewall'

PS:\>Get-HELP New-AzFirewall -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

New-AzFirewall

SYNOPSIS

Creates a new Firewall in a resource group.

SYNTAX

New-AzFirewall [-AllowActiveFTP] [-ApplicationRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallApplicationRuleCollection[]>] [-AsJob] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-DnsServer <System.String[]>] [-EnableDnsProxy] [-EnableFatFlowLogging] [-EnableUDPLogOptimization] [-FirewallPolicyId <System.String>] [-Force] [-HubIPAddress <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallHubIpAddresses>] -Location <System.String> [-ManagementPublicIpAddress <Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress>] -Name <System.String> [-NatRuleCollection

<Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNatRuleCollection[]>] [-NetworkRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNetworkRuleCollection[]>] [-PrivateRange <System.String[]>] [-PublicIpAddress <Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress[]>] -ResourceGroupName <System.String> [-RouteServerId <System.String>] [-SkuName {AZFW_Hub | AZFW_VNet}] [-SkuTier {Standard | Premium | Basic}] [-Tag <System.Collections.Hashtable>] [-ThreatIntelMode {Alert | Deny | Off}] [-ThreatIntelWhitelist <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallThreatIntelWhitelist>] [-VirtualHubId <System.String>] -VirtualNetwork <Microsoft.Azure.Commands.Network.Models.PSVirtualNetwork> [-Zone <System.String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

New-AzFirewall [-AllowActiveFTP] [-ApplicationRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallApplicationRuleCollection[]>] [-AsJob] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-DnsServer <System.String[]>] [-EnableDnsProxy] [-EnableFatFlowLogging] [-EnableUDPLogOptimization] [-FirewallPolicyId <System.String>] [-Force] [-HubIPAddress <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallHubIpAddresses>] -Location <System.String> -Name <System.String> [-NatRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNatRuleCollection[]>] [-NetworkRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNetworkRuleCollection[]>] [-PrivateRange <System.String[]>] [-PublicIpName <System.String>] -ResourceGroupName <System.String> [-RouteServerId <System.String>] [-SkuName {AZFW_Hub | AZFW_VNet}] [-SkuTier {Standard | Premium | Basic}] [-Tag <System.Collections.Hashtable>] [-ThreatIntelMode {Alert | Deny | Off}] [-ThreatIntelWhitelist <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallThreatIntelWhitelist>] [-VirtualHubId <System.String>] -VirtualNetworkName <System.String> [-Zone <System.String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The New-AzFirewall cmdlet creates an Azure Firewall.

PARAMETERS

-AllowActiveFTP <System.Management.Automation.SwitchParameter>

Allows Active FTP on the Firewall. By default it is disabled.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-ApplicationRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallApplicationRuleCollection[]>

Specifies the collections of application rules for the new Firewall.

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

-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

-DnsServer <System.String[]>

The list of DNS Servers to be used for DNS resolution,

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-EnableDnsProxy <System.Management.Automation.SwitchParameter>

Enable DNS Proxy. By default it is disabled.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnableFatFlowLogging <System.Management.Automation.SwitchParameter>

Enable Fat Flow Logging. By default it is false.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EnableUDPLogOptimization <System.Management.Automation.SwitchParameter>

Enable UDP Log Optimization. By default it is false.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-FirewallPolicyId <System.String>

The firewall policy attached to the firewall

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Force <System.Management.Automation.SwitchParameter>

Forces the command to run without asking for user confirmation.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-HubIPAddress <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallHubIpAddresses>

The ip addresses for the firewall attached to a virtual hub

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Location <System.String>

Specifies the region for the Firewall.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ManagementPublicIpAddress <Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress>

One or more Public IP Addresses to use for management traffic. The Public IP addresses must use Standard SKU and must belong to the same resource group as the

Firewall.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Name <System.String>

Specifies the name of the Azure Firewall that this cmdlet creates.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-NatRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNatRuleCollection[]>

The list of AzureFirewallNatRuleCollections

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-NetworkRuleCollection <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallNetworkRuleCollection[]>

The list of AzureFirewallNetworkRuleCollections

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PrivateRange <System.String[]>

The private IP ranges to which traffic won't be SNAT'ed

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PublicIpAddress <Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress[]>

One or more Public IP Addresses. The Public IP addresses must use Standard SKU and must belong to the same resource group as the Firewall. No input needed for

Forced Tunneling Firewalls.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-PublicIpName <System.String>

Public Ip Name. The Public IP must use Standard SKU and must belong to the same resource group as the Firewall.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ResourceGroupName <System.String>

Specifies the name of a resource group to contain the Firewall.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-RouteServerId <System.String>

The Route Server Id for the firewall

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-SkuName <System.String>

The sku name for firewall

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-SkuTier <System.String>

The sku tier for firewall

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Tag <System.Collections.Hashtable>

Key-value pairs in the form of a hash table. For example:

@{key0="value0";key1=$null;key2="value2"}

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ThreatIntelMode <System.String>

Specifies the operation mode for Threat Intelligence. Default mode is Alert, not Off.

Required? false

Position? named

Default value Alert

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-ThreatIntelWhitelist <Microsoft.Azure.Commands.Network.Models.PSAzureFirewallThreatIntelWhitelist>

The allowlist for Threat Intelligence

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-VirtualHubId <System.String>

The virtual hub that a firewall is attached to

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

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

Virtual Network

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-VirtualNetworkName <System.String>

Specifies the name of the virtual network for which the Firewall will be deployed. Virtual network and Firewall must belong to the same resource group.

Required? true

Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Zone <System.String[]>

A list of availability zones denoting where the firewall needs to come from.

Required? false

Position? named

Default value None

Accept pipeline input? False

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

System.String

Microsoft.Azure.Commands.Network.Models.PSVirtualNetwork

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

Microsoft.Azure.Commands.Network.Models.PSPublicIpAddress

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

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

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

System.Collections.Hashtable

OUTPUTS

Microsoft.Azure.Commands.Network.Models.PSAzureFirewall

NOTES

-- Example 1: Create a Firewall attached to a virtual network --

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip
```

This example creates a Firewall attached to virtual network "vnet" in the same resource group as the firewall. Since no rules were specified, the firewall will block

all traffic (default behavior). Threat Intel will also run in default mode - Alert - which means malicious traffic will be logged, but not denied.

- Example 2: Create a Firewall which allows all HTTPS traffic -

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"

$rule = New-AzFirewallApplicationRule -Name R1 -Protocol "https:443" -TargetFqdn "*"
$ruleCollection = New-AzFirewallApplicationRuleCollection -Name RC1 -Priority 100 -Rule $rule -ActionType "Allow"
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ApplicationRuleCollection $ruleCollection
```

This example creates a Firewall which allows all HTTPS traffic on port 443. Threat Intel will run in default mode - Alert - which means malicious traffic will be

logged, but not denied.

Example 3: DNAT - redirect traffic destined to 10.1.2.3:80 to 10.2.3.4:8080

```
$rule = New-AzFirewallNatRule -Name "natRule" -Protocol "TCP" -SourceAddress "*" -DestinationAddress "10.1.2.3" -DestinationPort "80" -TranslatedAddress "10.2.3.4"
-TranslatedPort "8080"
$ruleCollection = New-AzFirewallNatRuleCollection -Name "NatRuleCollection" -Priority 1000 -Rule $rule
New-AzFirewall -Name "azFw" -ResourceGroupName "rg" -Location centralus -NatRuleCollection $ruleCollection -ThreatIntelMode Off
```

This example created a Firewall which translated the destination IP and port of all packets destined to 10.1.2.3:80 to 10.2.3.4:8080 Threat Intel is turned off in

this example.

Example 4: Create a Firewall with no rules and with Threat Intel in Alert mode

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ThreatIntelMode Alert
```

This example creates a Firewall which blocks all traffic (default behavior) and has Threat Intel running in Alert mode. This means alerting logs are emitted for

malicious traffic before applying the other rules (in this case just the default rule - Deny All)

Example 5: Create a Firewall which allows all HTTP traffic on port 8080, but blocks malicious domains identified by Threat Intel

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"

$rule = New-AzFirewallApplicationRule -Name R1 -Protocol "http:8080" -TargetFqdn "*"
$ruleCollection = New-AzFirewallApplicationRuleCollection -Name RC1 -Priority 100 -Rule $rule -ActionType "Allow"
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ApplicationRuleCollection $ruleCollection
-ThreatIntelMode Deny
```

This example creates a Firewall which allows all HTTP traffic on port 8080 unless it is considered malicious by Threat Intel. When running in Deny mode, unlike Alert,
traffic considered malicious by Threat Intel is not just logged, but also blocked.

Example 6: Create a Firewall with no rules and with availability zones

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
 New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetworkName $vnet.Name -PublicIpName $pip.Name -Zone 1,2,3
```

This example creates a Firewall with all available availability zones.

Example 7: Create a Firewall with two or more Public IP Addresses

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -Name "vnet" -ResourceGroupName $rgName
```

$pip1 = New-AzPublicIpAddress -Name "AzFwPublicIp1" -ResourceGroupName "rg" -Sku "Basic" -Tier "Regional" -Location "centralus" -AllocationMethod Static

$pip2 = New-AzPublicIpAddress -Name "AzFwPublicIp2" -ResourceGroupName "rg" -Sku "Basic" -Tier "Regional" -Location "centralus" -AllocationMethod Static

New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress @($pip1, $pip2)

This example creates a Firewall attached to virtual network "vnet" with two public IP addresses.

Example 8: Create a Firewall which allows MSSQL traffic to specific SQL database

$rgName = "resourceGroupName"

$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"

$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"

$rule = New-AzFirewallApplicationRule -Name R1 -Protocol "mssql:1433" -TargetFqdn "sql1.database.windows.net"

$ruleCollection = New-AzFirewallApplicationRuleCollection -Name RC1 -Priority 100 -Rule $rule -ActionType "Allow"

New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ApplicationRuleCollection $ruleCollection

-ThreatIntelMode Deny

This example creates a Firewall which allows MSSQL traffic on standard port 1433 to SQL database sql1.database.windows.net.

---- Example 9: Create a Firewall attached to a virtual hub ----

$rgName = "resourceGroupName"

$fp = Get-AzFirewallPolicy -ResourceGroupName $rgName -Name "fp"

$fpId = $fp.Id

$vHub = Get-AzVirtualHub -Name "hub"

$vHubId = $vHub.Id

New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -SkuName AZFW_Hub -VirtualHubId $vHubId -FirewallPolicyId -$fpId

This example creates a Firewall attached to virtual hub "vHub". A firewall policy $fp will be attached to the firewall. This firewall allows/denies the traffic based

on the rules mentioned in the firewall policy $fp. The virtual hub and the firewall should be in the same regions.

Example 10: Create a Firewall with threat intelligence allowlist setup

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"

$tiWhitelist = New-AzFirewallThreatIntelWhitelist -FQDN @("www.microsoft.com") -IpAddress @("8.8.8.8")
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ThreatIntelWhitelist $tiWhitelist
```

This example creates a Firewall that allowlists "www.microsoft.com" and "8.8.8.8" from threat intelligence

Example 11: Create a Firewall with customized private range setup

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"

    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -PrivateRange @("99.99.99.0/24", "66.66.0.0/16")
```

This example creates a Firewall that treats "99.99.99.0/24" and "66.66.0.0/16" as private ip ranges and won't snat traffic to those addresses

Example 12: Create a Firewall with a management subnet and Public IP address

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
$mgmtPip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "managementPublicIpName"

New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -ManagementPublicIpAddress $mgmtPip
```

This example creates a Firewall attached to virtual network "vnet" in the same resource group as the firewall. Since no rules were specified, the firewall will block

all traffic (default behavior). Threat Intel will also run in default mode - Alert - which means malicious traffic will be logged, but not denied.

To support "forced tunneling" scenarios, this firewall will use the subnet "AzureFirewallManagementSubnet" and the management public IP address for its management

traffic

Example 13: Create a Firewall with Firewall Policy attached to a virtual network

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
$fp = Get-AzFirewallPolicy -ResourceGroupName $rgName -Name "fp"
New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -FirewallPolicyId $fp
```

This example creates a Firewall attached to virtual network "vnet" in the same resource group as the firewall. The rules and threat intelligence that will be applied

to the firewall will be taken from the firewall policy

- Example 14: Create a Firewall with DNS Proxy and DNS Servers -

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -DnsServer @("10.10.10.1", "20.20.20.2")
```

This example creates a Firewall attached to virtual network "vnet" in the same resource group as the firewall. DNS Proxy is enabled for this firewall and 2 DNS

Servers are provided. Also Require DNS Proxy for Network rules is set so if there are any Network rules with FQDNs then DNS proxy will be used for them too.

Example 15: Create a Firewall with multiple IPs. The Firewall can be associated with the Virtual Hub

```
$rgName = "resourceGroupName"
$vHub = Get-AzVirtualHub -Name "hub"
$vHubId = $vHub.Id
$fwpips = New-AzFirewallHubPublicIpAddress -Count 2
$hubIpAddresses = New-AzFirewallHubIpAddress -PublicIP $fwpips
    $fw=New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location westus -SkuName AZFW_Hub -HubIPAddress $hubIpAddresses -VirtualHubId $vHubId
```

This example creates a Firewall attached to virtual hub "hub" in the same resource group as the firewall. The Firewall will be assigned 2 public IPs that are created

implicitly.

---- Example 16: Create a Firewall with Allow Active FTP. ----

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$pip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "publicIpName"
    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -PublicIpAddress $pip -AllowActiveFTP
```

This example creates a Firewall with allow active FTP flag.

Example 17: Create a Firewall with a management subnet and no data Public IP address

```
$rgName = "resourceGroupName"
$vnet = Get-AzVirtualNetwork -ResourceGroupName $rgName -Name "vnet"
$mgmtPip = Get-AzPublicIpAddress -ResourceGroupName $rgName -Name "managementPublicIpName"

    New-AzFirewall -Name "azFw" -ResourceGroupName $rgName -Location centralus -VirtualNetwork $vnet -ManagementPublicIpAddress $mgmtPip
```

This example creates a "forced tunneling" Firewall that uses the subnet "AzureFirewallManagementSubnet" and the management public IP address for its management
traffic. In this scenario, users do not have to specify a data Public IP if they are only using firewall for private traffic only.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.network/new-azfirewall

Get-AzFirewall

Remove-AzFirewall

Set-AzFirewall

New-AzFirewallApplicationRuleCollection

New-AzFirewallNatRuleCollection

New-AzFirewallNetworkRuleCollection