



How to Configure Azure Load Balancer for HA Clusters using PowerShell and ARM

For incoming traffic to automatically fail over to the secondary HA firewall in case of a failover, you must configure an Azure load balancer. The load balancer has its own public IP address that can be allocated either dynamically or statically. For each service that is made available through the F-Series Firewall HA cluster, you must create a load balancer rule. At least one health probe must be configured for the load balancer to be able to determine which firewall is currently active. Depending on the probing interval, failover times may vary because the load balancer take several seconds. The load balancer only controls the incoming connections from the Internet. For internal VMs to use the secondary firewall as the default gateway, you must configure UDR routing. [How to Configure Azure Route Tables \(UDR\) using PowerShell and ARM](#) or [How to Configure Azure Route Tables \(UDR\) using Azure Portal and ARM](#).

Example Load Balancer Deployment Script

You can combine the PowerShell commandlets to customize the deployment of the Azure Load Balancer. See below for an example script. This script assumes that you already deployed your HA cluster and that you are logged in to your Azure Account from the PowerShell.

[Click here to show the example Load Balancer script](#)

```
$VerbosePreference = 'Continue' $ErrorActionPreference = 'Stop' $LBdomainnamelabel =
'YOUR_DOMAIN_LABEL' $LBResourceGroupName = 'RESOURCE_GROUP_FOR_LOADBALANCER'
$LBPublicIPName = 'NGF-LB-IP' $LBName =#NGF-LB# $location = 'West Europe' $nic1Name
= 'DOC-NGF1-NIC1' # Name of the NIC on the primary firewall $nic2Name = 'DOC-NGF2-
NIC1' # Name of the NIC on the secondary firewall # Resource Group name the
firewalls are in. Adapt the script if the primary and secondary firewall are in
different resource groups $nicResourceGroupName = 'DOC-NGF' # create a new resource
group for the load balance New-AzureRmResourceGroup -Name $LBResourceGroupName -
Location $location # Create a static public IP # $LBpublicIP = New-
AzureRmPublicIpAddress -Name #NGF-LB-PIP# -ResourceGroupName $LBResourceGroupName -
Location $location -AllocationMethod Static -DomainNameLabel $LBdomainnamelabel
$LBpublicIP = New-AzureRmPublicIpAddress -Name #NGF-LB-PIP# -ResourceGroupName
$LBResourceGroupName -Location $location -AllocationMethod Static # Add the public
IP address as the frontend IP address $LBfrontend = New-
AzureRmLoadBalancerFrontendIpConfig -Name #NGF-LB-Frontend# -PublicIpAddress
$LBpublicIP # Create an address pool for the primary and secondary NextGen Firewall
F VM $LBbackend= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name #NGF-LB-
backend# ##### ## For each service
create a probe and a load balancing rule. Example probe and rule for TINA VPN and
SSL VPN below: # Create a probe for each service. $LBprobe1 = New-
AzureRmLoadBalancerProbeConfig -Name #TINA-Probe# -Protocol Tcp -Port 691 -
IntervalInSeconds 5 -ProbeCount 2 $LBprobe2 = New-AzureRmLoadBalancerProbeConfig -
Name #SSLVPN-Probe# -Protocol Http -Port 443 -IntervalInSeconds 5 -ProbeCount 2 -
RequestPath #/_formauth/login.html# # Create a Load Balancing rules for each service
$LBrule1 = New-AzureRmLoadBalancerRuleConfig -Name #TINA_TCP# -
FrontendIpConfiguration $LBfrontend -BackendAddressPool $LBbackend -Protocol TCP -
FrontendPort 691 -BackendPort 691 -Probe $LBprobe1 $LBrule2 = New-
AzureRmLoadBalancerRuleConfig -Name #SSLVPN# -FrontendIpConfiguration $LBfrontend -
BackendAddressPool $LBbackend -Protocol Tcp -FrontendPort 443 -BackendPort 443 -
Probe $LBprobe2 $LBrule3 = New-AzureRmLoadBalancerRuleConfig -Name #SSH# -
FrontendIpConfiguration $LBfrontend -BackendAddressPool $LBbackend -Protocol Tcp -
FrontendPort 22 -BackendPort 22 -Probe $LBprobe1 #####
```



```
# Create the Load Balancer # NOTE: Add the additional probes and load balancer rules
as necessary $LB = New-AzureRmLoadBalancer -ResourceGroupName $LBResourceGroupName -
Name $LBName -Location $location -FrontendIpConfiguration $LBfrontend -
LoadBalancingRule $LBrule1,$LBrule2,$LBrule3 -BackendAddressPool $LBbackend -Probe
$LBprobe1,$LBprobe2 # Add the NIC of the primary and secondary Barracuda NextGen
Firewall F VM to the load balancer and update the NIC configuration $nic1 = Get-
AzureRmNetworkInterface -ResourceGroupName $nicResourceGroupName -Name $nic1Name
$nic2 = Get-AzureRmNetworkInterface -ResourceGroupName $nicResourceGroupName -Name
$nic2Name
$nic1.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($LB.BackendAddressPool
s[0]);
$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($LB.BackendAddressPool
s[0]); $nic1 | Set-AzureRmNetworkInterface $nic2 | Set-AzureRmNetworkInterface
```

Before You Begin

- Look up the resource group and NIC name for the primary and secondary firewall.
- Install Azure PowerShell version 2.0.1 or higher.

Step 1. (optional) Create a Resource Group for the Load Balancer

Create a resource group for the load balancer, or add it to an existing resource group.

1. Open Azure PowerShell.
2. Create a resource group:

```
New-AzureRmResourceGroup -Name NETWORK_RESOURCE_GROUP_NAME -Location
YOUR_LOCATION
```

Step 2. Create a Public IP Address for the Load Balancer

The public IP address that is associated with the load balancer can either be allocated dynamically, or configured to be static. You can also add a DNS label and reverse domain lookup.

1. Open Azure PowerShell.
2. Create a static Azure Public IP. The domain name label is optional.

```
$PublicLoadBalancerIP = New-AzureRmPublicIpAddress -ResourceGroupName
RESOURCE_GROUP_NAME -Location LOCATION -Name PUBLIC_IP_NAME -DomainNameLabel
DOMAIN_NAME -AllocationMethod Static
```

Step 3. Create the Load Balancer Configuration

Create the configuration for the load balancer backend and frontend.

1. Open Azure PowerShell.
2. Create frontend and backend address pool configurations:

```
$LBfrontend = New-AzureRmLoadBalancerFrontendIpConfig -Name LB_FRONTEND_NAME -
PublicIpAddress $PublicLoadBalancerIP $LBbackend = New-
AzureRmLoadBalancerBackendAddressPoolConfig -Name LB_BACKEND_NAME
```



Step 4. Create the Load Balancer Rules

For each service, create a rule for the load balancer. Select one service and create the health monitor probe. When the probe fails, the VM is removed from the load balancer backend pool.

1. Open Azure PowerShell.
2. Create the health probes used to decide whether a service is healthy:

```
$LBprobe1 = New-AzureRmLoadBalancerProbeConfig -Name #LB-Probe# -Protocol Tcp -
Port 691 -IntervalInSeconds 5 -ProbeCount 2 $LBprobe2 = New-
AzureRmLoadBalancerProbeConfig -Name #SSLVPN-LB-Probe# -Protocol Tcp -Port 443
-IntervalInSeconds 5 -ProbeCount 2
```

3. For each service, create a load balancer rule and, optionally, a load balancer probe:

```
$LBrule1 = New-AzureRmLoadBalancerRuleConfig -Name #TINA-TCP# -
FrontendIpConfiguration $LBfrontend -BackendAddressPool $LBbackend -Protocol
Tcp -FrontendPort 691 -BackendPort 691 -Probe $LBprobe1 $LBrule2 = New-
AzureRmLoadBalancerRuleConfig -Name #SSL-VPN# -FrontendIpConfiguration
$LBfrontend -BackendAddressPool $LBbackend -Protocol Tcp -FrontendPort 443 -
BackendPort 443 -Probe $LBprobe2
```

Step 5. Create the Azure Load Balancer

Use the configuration and the rules from steps 3 and 4 to create the load balancer.

1. Open Azure PowerShell.
2. Create the load balancer with the frontend and backend configuration as well as the comma-separated lists for the Loadbalancer rules created in steps 3 and 4:

```
$LB = New-AzureRmLoadBalancer -ResourceGroupName $LBResourceGroupName -Name
$LBName -Location $location -FrontendIpConfiguration $LBfrontend -
LoadBalancingRule $LBrule1,$LBrule2 -BackendAddressPool $LBbackend -Probe
$LBprobe1,$LBprobe2
```

Step 6. Add the NICs to the Load Balancer Backend Pool

1. Open Azure PowerShell.
2. Store both NICs you want to add to variables:

```
$nic1 = Get-AzureRmNetworkInterface -ResourceGroupName NGF_RESOURCE_GROUP -Name
NIC1_NAME $nic2 = Get-AzureRmNetworkInterface -ResourceGroupName
NGF_RESOURCE_GROUP -Name NIC2_NAME
```

3. Add the NICs to the backend pool and update the NIC configuration:

```
$nic1.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($LB.BackendAdres
sPools[0]);
$nic2.IpConfigurations[0].LoadBalancerBackendAddressPools.Add($LB.BackendAdres
sPools[0]); $nic1 | Set-AzureRmNetworkInterface $nic2 | Set-
AzureRmNetworkInterface
```

