How to Configure Azure Cloud Integration using ARM

https://campus.barracuda.com/doc/48202630/

Azure Cloud integration allows the firewall to connect directly to the Azure service fabric to rewrite Azure User Defined Routes and to monitor the IP Forwarding setting of the NIC of your firewall VM. Azure User Defined Routing allows you to use the Firewall F-Series high availability cluster in the public subnet as the default gateway for all your VMs running in the backend networks. You must enable IP forwarding for the firewall VMs and create and apply an Azure routing table to the backend networks. Using a management certificate and the Azure subscriber ID, the firewall VMs can change the Azure routing table on the fly when the virtual server fails over from one VM to the other. Azure route table rewriting must be configured on the primary and secondary F-Series Firewall. If a global HTTP proxy is configured, all REST API calls are sent via the proxy. The Azure AD Application and the management certificate must be valid for the same length of time. By default the Azure AD application is valid for exactly one year. This value can be extended when creating the Azure Application to match the expiration date of the certificate.

Example script for Azure PowerShell 3.5

Create the certificates according to the steps in the article. Use the example script below to configure Cloud Integration without having to enter the PowerShell commands one-by-one. Set the variables in the script to match your setup.

Use this script if you are using Azure PowerShell version 3.5

```
$pathToCERfile = 'PATH_TO\arm.cer' $ADAppName = 'NGF' # Set the resource
group the Azure Route Table is in $resourceGroupName = 'RESOURCE_GROUP_NAME'
# your subscription ID $subscriptionID = '/subscriptions/YOURSUBSCRIPTIONID'
# the identifier and role name must both be unique $identifier =
'http://localhost' $roleName = 'NGF Role' # Number of days until the AD
application expires. # Must match the expiration date of the certificate 730
= 2 years $validNumDays = 730 # Select the Azure subscription Select-
AzureRmSubscription -SubscriptionId $subscriptionID # Create a custom role
for NGF Cloud Integration. An existing role is cloned, all rights removed and
then assigned proper privileges $role = Get-AzureRmRoleDefinition "Virtual
Machine Contributor" $role.Id = $null $role.Name = $roleName
$role.Description = "Barracuda NextGen Firewall Cloud Integration"
$role.Actions.Clear() # Add role definitions to the empty role
$role.Actions.Add("Microsoft.Compute/virtualMachines/*")
$role.Actions.Add("Microsoft.Network/*") $role.AssignableScopes.Clear()
$role.AssignableScopes.Add($subscriptionID) $firewallRole = New-
AzureRmRoleDefinition -Role $role # convert date $endDate =
[System.DateTime]::Parse((date).ToString("yyyy.MM.dd")) $timespan = New-
TimeSpan -Days $validNumDays $endDate = $endDate + $timespan # convert and
```

```
upload the authentication certificate $cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate($pathToCERfile)
$key = [System.Convert]::ToBase64String($cert.GetRawCertData()) $app = New-
AzureRmADApplication -DisplayName $ADAppName -HomePage $identifier -
IdentifierUris $identifier -CertValue $key -EndDate $endDate $princ = New-
AzureRmADServicePrincipal -ApplicationId $app.ApplicationId -Verbose #wait
for the service principal to be created Start-Sleep -Seconds 30 New-
AzureRmRoleAssignment -RoleDefinitionName $firewallRole.Name -
ServicePrincipalName $princ.ServicePrincipalNames[0]
```

Example scripts for older Azure PowerShell versions

It is recommended to update to the latest PowerShell version to be able to use the newest version of this script. If this is not possible, use the example scripts below that match your Azure PowerShell version. Custom firewall role definitions are not supported for older Azure PowerShell versions. The scripts for older Azure PowerShell versions create an Azure AD application valid for one year. To find out which Azure PowerShell version you are using, enter the following PowerShell command:

```
Get-Module -ListAvailable -Name Azure -Refresh
```

Example Script for Azure PowerShell 1.0.1 and 1.1.0

Use this script if you are using Azure PowerShell version 1.0.1 or 1.1.0:

```
$pathToCERfile = 'PATH_TO\arm.cer' $ADAppName = 'NGFUDR' $roleDefName =
'owner' # Set the resource group the Azure Route Table is in
$resourceGroupName = 'RESOURCE_GROUP_NAME' # the identifier must be unique
$identifier = 'http://localhost' $cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate($pathToCERfile)
$key = [System.Convert]::ToBase64String($cert.GetRawCertData()) $app = New-
AzureRmADApplication -DisplayName $ADAppName -HomePage $identifier -
IdentifierUris $identifier -KeyValue $key -KeyType AsymmetricX509Cert New-
AzureRmADServicePrincipal -ApplicationId $app.ApplicationId -Verbose New-
AzureRmRoleAssignment -RoleDefinitionName $roleDefName -ServicePrincipalName
$app.ApplicationId -ResourceGroupName $resourceGroupName
```

Example Script for Azure PowerShell 2.1

Use this script if you are using Azure PowerShell version 2.1:

```
$pathToCERfile = 'PATH_TO\arm.cer' $ADAppName = 'NGFUDR' $roleDefName =
'Network Contributor' # Set the resource group the Azure Route Table is in
```

```
$resourceGroupName = 'RESOURCE_GROUP_NAME' # your subscription ID
$subscriptionID = '/subscriptions/YOURSUBSCRIPTIONID' # the identifier must
be unique $identifier = 'http://localhost' # the identifier and role name
must both be unique $identifier = 'http://localhost' $roleName = 'NGF Role' #
Select the Azure subscription Select-AzureRmSubscription -SubscriptionId
$subscriptionID # Create a custom role for NGF Cloud Integration. An existing
role is cloned, all rights removed and then assigned proper privileges $role
= Get-AzureRmRoleDefinition "Virtual Machine Contributor" $role.Id = $null
$role.Name = $roleName $role.Description = "Barracuda NextGen Firewall Cloud
Integration" $role.Actions.Clear() # Add role definitions to the empty role
$role.Actions.Add("Microsoft.Compute/virtualMachines/*")
$role.Actions.Add("Microsoft.Network/*") $role.AssignableScopes.Clear()
$role.AssignableScopes.Add($subscriptionID) $firewallRole = New-
AzureRmRoleDefinition -Role $role $cert = New-Object
System.Security.Cryptography.X509Certificates.X509Certificate($pathToCERfile)
$key = [System.Convert]::ToBase64String($cert.GetRawCertData()) $app = New-
AzureRmADApplication -DisplayName $ADAppName -HomePage $identifier -
IdentifierUris $identifier -CertValue $key New-AzureRmADServicePrincipal -
ApplicationId $app.ApplicationId -Verbose #wait for the service principal to
be created Start-Sleep -Seconds 30 New-AzureRmRoleAssignment -
RoleDefinitionName $firewallRole -ServicePrincipalName $app.ApplicationId
```

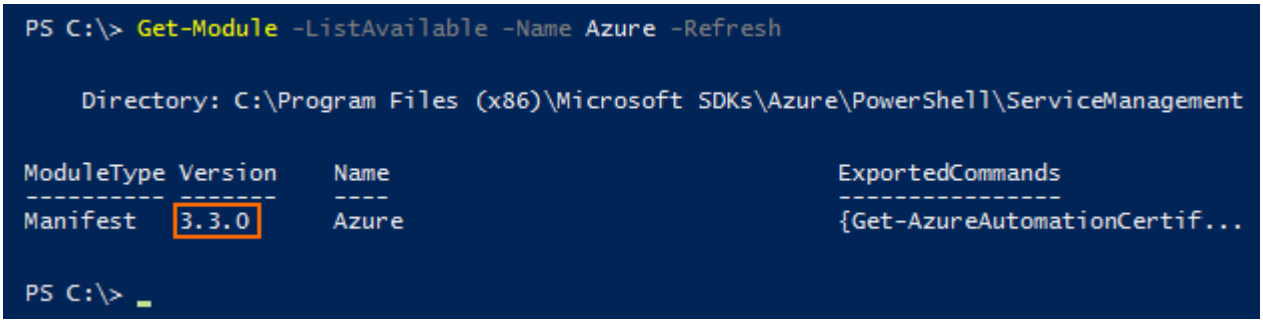
Before you begin

- Deploy your Barracuda NextGen F-Series Firewall, and configure Azure UDR using the **Azure Resource Manager** (ARM).
- Verify that you are using **Azure PowerShell 3.5.0** or higher.
- Verify that a DNS server is configured. For more information, see How to Configure DNS Settings.
- Log into your Azure account using `Login-AzureRmAccount`

Step 1. Verify the Azure PowerShell version

Verify that you are using the required Azure PowerShell version (see **Before you begin**). If you must use an older version, use the example scripts above that match your version.

1. Launch Azure PowerShell.
2. Get the Azure PowerShell version:
   `Get-Module -ListAvailable -Name Azure -Refresh`



3. If needed, update Azure PowerShell to match the required version.

Step 2. Create the Azure management certificate

For the firewall to be able to connect to the Azure backend, you must create and upload a management certificate. The certificate must be valid for at least two years.

1. Log into the firewall via ssh.
2. Create the certificate:
   `openssl req -x509 -nodes -days 1095 -newkey rsa:2048 -keyout arm.pem -out arm.pem`
3. Answer the questions at the prompt. The **Common Name** is used to identify this certificate in the Azure web interface.
4. Convert the certificate to CER, as required by Azure:
   `openssl x509 -inform pem -in arm.pem -outform der -out arm.cer`
5. Extract the RSA key:
   `openssl rsa -in arm.pem -out arm.key.pem`

You now have three certificates: *arm.pem, arm.key.pem* and *arm.cer*.

Step 3. Create a custom Azure access control role for Cloud Integration

Create a custom role to use with Cloud Integration.

> The **role name** must be unique.

1. Launch Azure PowerShell.
2. Create a new role by cloning an existing role. Clear all privileges and then add only the privileges needed for Cloud Integration. The subscription ID must be entered in the following format: `"/subscriptions/abcdefg1234567891011212"`.
   ```
   # Create a custom role for NGF Cloud Integration. An existing role is
   ```

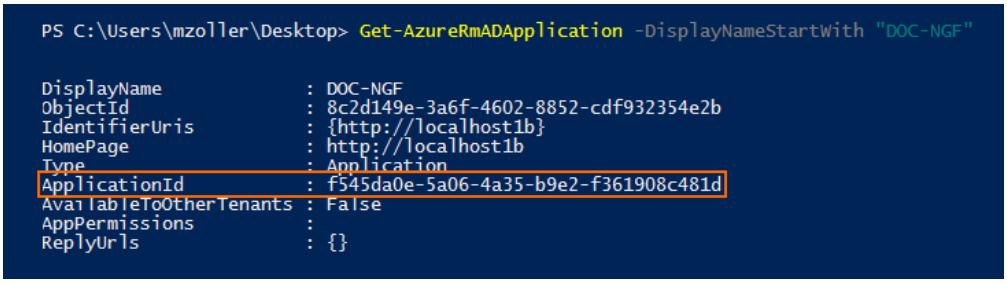
```
cloned, all rights removed and then assigned proper privileges $role =
Get-AzureRmRoleDefinition "Virtual Machine Contributor" $role.Id = $null
$role.Name = 'NGF Role' $role.Description = "Barracuda NextGen Firewall
Cloud Integration" $role.Actions.Clear() # Add role definitions to the
empty role $role.Actions.Add("Microsoft.Compute/virtualMachines/*")
$role.Actions.Add("Microsoft.Network/*") $role.AssignableScopes.Clear()
$role.AssignableScopes.Add(YOUR_SUBSCRIPTION_ID) $firewallRole = New-
AzureRmRoleDefinition -Role $role
```

Step 4. Upload the Azure management certificate via Azure PowerShell

The Azure AD application and the certificate must be valid for the same length of time. Otherwise Authentication errors will occur.

> The **identifierURis** must be unique.

1. Launch Azure PowerShell.
2. Store the **EndDate** in a variable. it is recommended to set the **EndDate** or the Azure AD application and the expiration date of the certificate to the same value.
   ```
   $endDate = [System.DateTime]::Parse((date).ToString("yyyy.MM.dd"))
   $timespan = New-TimeSpan -Days VALID_FOR_NUM_DAYS $endDate = $endDate +
   $timespan
   ```
3. Execute the following commands to import *arm.cer* as a management certificate:
   ```
   Login-AzureRmAccount $cert = New-Object
   System.Security.Cryptography.X509Certificates.X509Certificate("PATH_TO_C
   ER_FILE") $key =
   [System.Convert]::ToBase64String($cert.GetRawCertData()) $app = New-
   AzureRmADApplication -DisplayName "DISPLAY_NAME" -HomePage
   "http://localhost" -IdentifierUris "http://localhost" -CertValue $key -
   EndDate $endDate # write down the application ID ($app.ApplicationID is
   the "client ID" in NextGen Admin) $princ = New-AzureRmADServicePrincipal
   -ApplicationId $app.ApplicationId # wait for the service principal to be
   created Start-Sleep -Seconds 30 # in the command below, you can use "-
   Scope" to restrict permissions to specific resource groups New-
   AzureRmRoleAssignment -RoleDefinitionName $firewallRole.Name -
   ServicePrincipalName $princ.ServicePrincipalNames[0]
   ```
4. Get the Application ID.
   ```
   Get-AzureRmADApplication -DisplayNameStartWith "YOUR_ADAPPNAME"
   ```

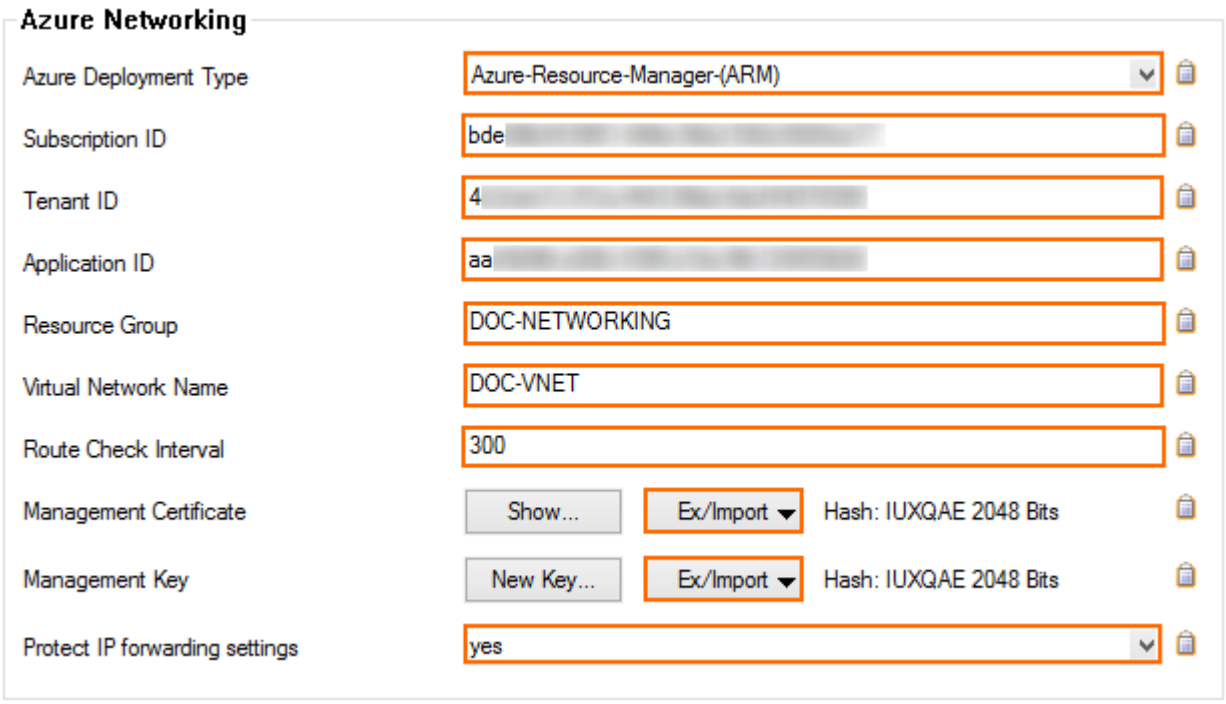


Write down the **ApplicationId** for step 5.

Step 5. Configure User Defined Routing and IP Forward Protection

You must enter your Azure ARM IDs and upload the management certificate created in Step 1 to allow the F-Series Firewall to change the Azure User Defined Routing Table and to monitor the IP Forwarding setting via ARM.

1. Go to **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > Cloud Integration**.
2. Click **Lock**.
3. In the left menu, click **Azure Networking**.
4. Select **Azure Resource Manager (ARM)** from the **Azure Deployment Type** drop-down list.
5. Enter your Azure **Subscription ID**.
6. Enter your Azure **Tenant ID**.
7. Enter your Azure **Application ID**.
8. Enter the **Resource Group** name.
9. Enter the **Virtual Network Name**. E.g., `DOC-VNET`
10. Enter the **Route Check Interval**. Default: `300`
11. Next to **Management Certificate** click **Ex/Import** and select **Import from PEM File**. The **File browser** window opens.
12. Select the *arm.pem* certificate created in Step 1, and click **Open**.
13. Next to **Management Key** click **Ex/Import** and select **Import from File**. The **File browser** window opens.
14. Select the *arm.key.pem* certificate created in Step 1, and click **Open**.
15. From the **Protect IP Forwarding Settings** select **yes** to monitor the **IP Forwarding** setting of the NIC.



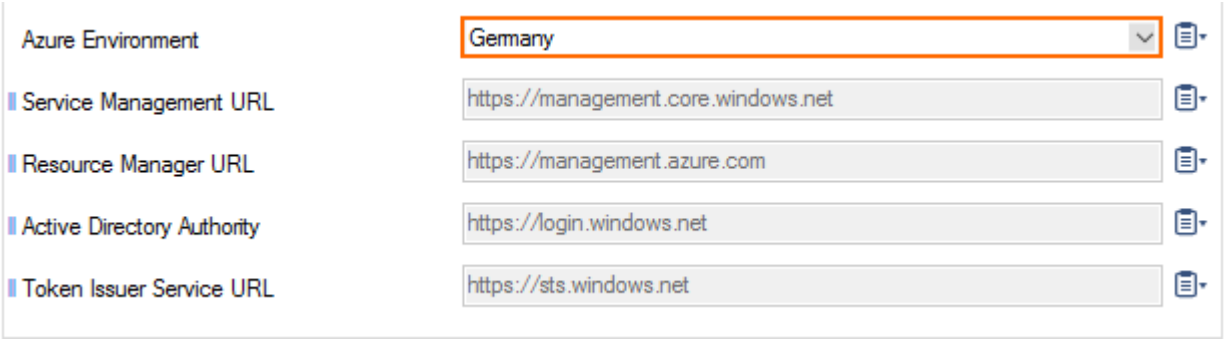
16. Click **Send Changes** and **Activate**.

The Azure routing table is now updated every time the virtual server fails over.

Step 6. (optional) Set the Azure environment

If you are running your firewall in a non-default Azure environment, such as Azure Germany, govcloud, Azure China, or Azure Stack, you must configure the Azure environment.

1. Go to **CONFIGURATION > Configuration Tree > Box > Advanced Configuration > Cloud Integration**.
2. Click **Lock**.
3. In the left menu, click **Azure Networking**.
4. Select the **Azure Environment** from the list. If your Azure environment is not in the list, select **Explicit**.
5. (Explicit only) In the left menu, expand the **Configuration Mode** section and click **Switch to Advanced View**.
6. (Explicit only) Enter the following setting for your Azure environment:
   - **Service Manager URL**
   - **Resource Manager URL**
   - **Active Directory Authority**
   - **Token Issuer Service URL**



7. Click **Send Changes** and **Activate**.

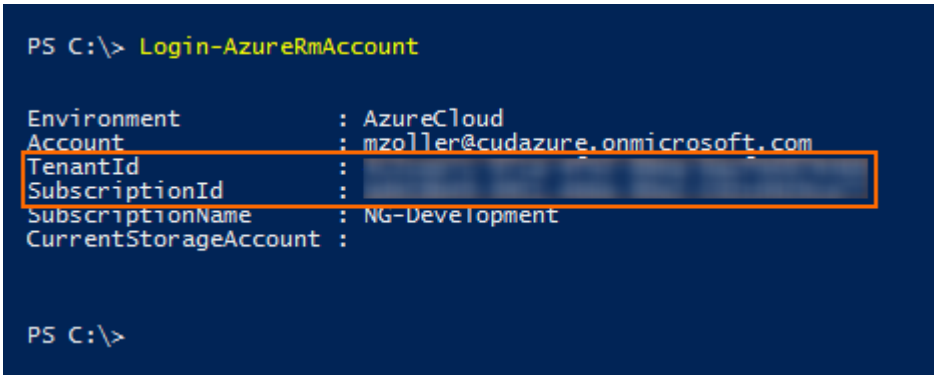
Next steps

Repeat steps 3 and 4 of this configuration for the other firewall VM in the HA cluster.

Getting tenant ID and subscription ID for existing setups

It might take a couple of minutes for the user to be propagated in Azure AD.

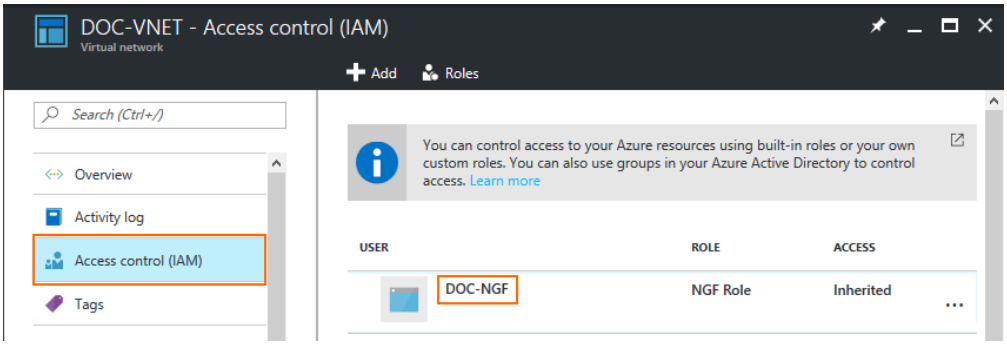
1. Launch Azure PowerShell.
2. The **SubscriptionId** and **TenantId** are listed after logging in via the **Login-AzureRmAccount** commandlet.



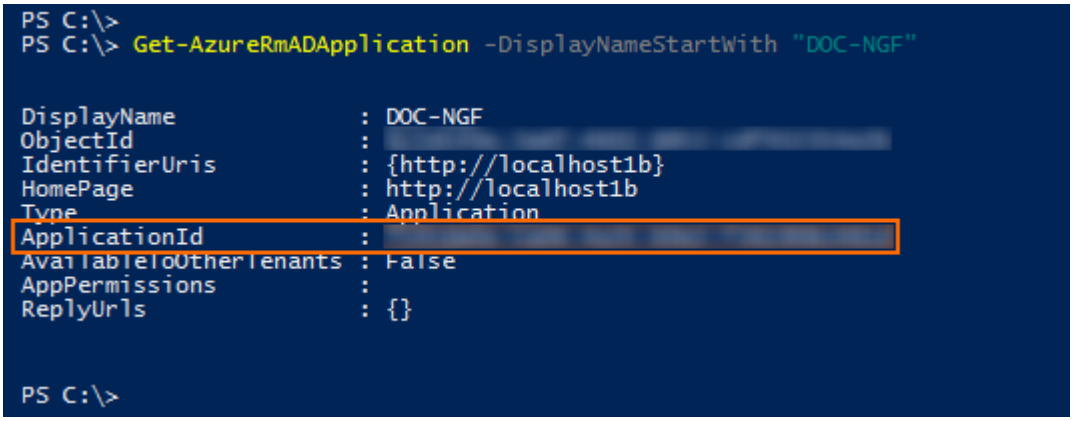
Getting application ID for existing setups

It might take a couple of minutes for the user to be propagated in Azure AD.

1. Go to the **Access control (IAM)** settings of your **virtual network**.
2. Locate the **ADAppname** in the **User** column of the custom role you created for your firewall.

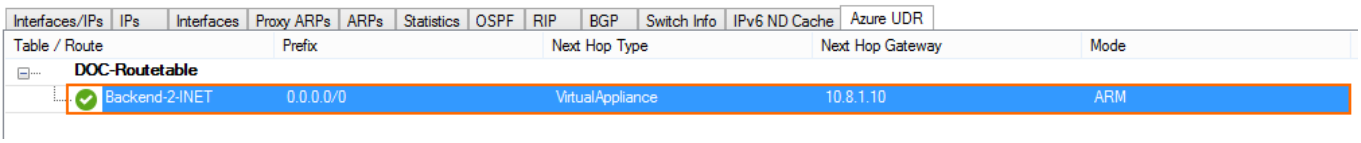


3. Launch Azure PowerShell.
4. Retrieve the **ADApplication** using the username:
   `Get-AzureRmADApplication -DisplayNameStartWith "YOUR_ADAPPNAME"`

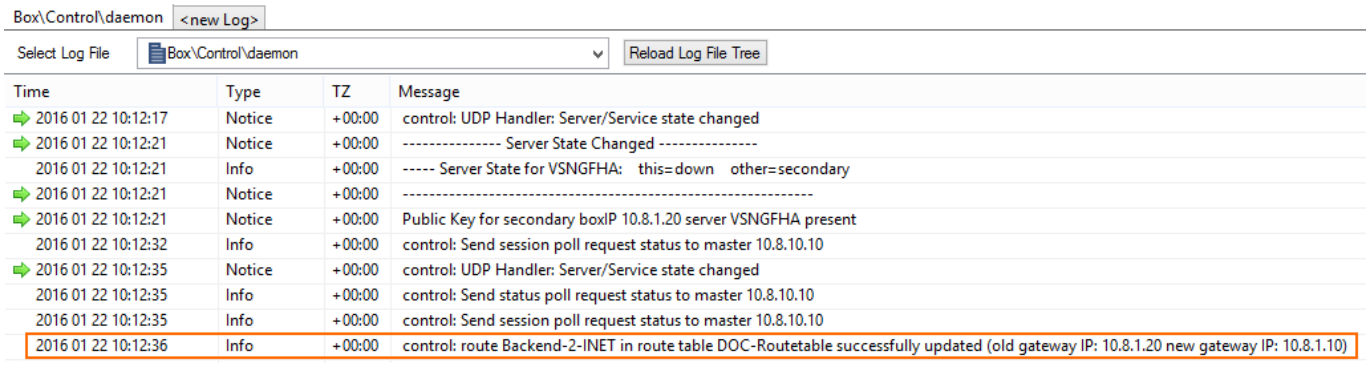


Monitoring

Go to **NETWORK > Azure UDR** to see the UDR routing table for all subnets in the firewall's VNET. Routes using the firewall VM as the nexthop are marked with a green icon. This icon changes to red during the UDR HA failover process.



All activity is logged to the **Box\Control\daemon** log file


Box\Control\daemon
<new Log>
Select Log File
Box\Control\daemon
Reload Log File Tree
Time
Type
TZ
Message
2016 01 22 10:12:17 Notice +00:00 control: UDP Handler: Server/Service state changed
2016 01 22 10:12:21 Notice +00:00 ---------- Server State Changed ----------
2016 01 22 10:12:21 Info +00:00 ----- Server State for VSNGFHA: this=down other=secondary
2016 01 22 10:12:21 Notice +00:00 ----------
2016 01 22 10:12:21 Notice +00:00 Public Key for secondary boxIP 10.8.1.20 server VSNGFHA present
2016 01 22 10:12:32 Info +00:00 control: Send session poll request status to master 10.8.10.10
2016 01 22 10:12:35 Notice +00:00 control: UDP Handler: Server/Service state changed
2016 01 22 10:12:35 Info +00:00 control: Send status poll request status to master 10.8.10.10
2016 01 22 10:12:35 Info +00:00 control: Send session poll request status to master 10.8.10.10
2016 01 22 10:12:36 Info +00:00 control: route Backend-2-INET in route table DOC-Routetable successfully updated (old gateway IP: 10.8.1.20 new gateway IP: 10.8.1.10)

Figures

1. get_azure_powershell_version.png
2. Azure_get_applicationID.png
3. UDR_HA_ARM.png
4. azure_environment_01.png
5. udr_get_subscription_tenantID.png
6. udr_get_user_name.png
7. udr_get_applicationID.png
8. ARM-UDR_01.png
9. ARM-UDR_02.png