How to Configure Azure Cloud Integration using ARM

https://campus.barracuda.com/doc/79463432/

Azure Cloud Integration allows the firewall to connect directly to the Azure service fabric in order to rewrite Azure user-defined routes and to monitor the IP forwarding setting of the NIC of your firewall VM. On the Azure side, an Azure AD application is created. Certificate authentication is used to authenticate the firewall when accessing the Azure API endpoints. The certificate must be valid for at least 1 year. The end date of the certificate is used by the setup script to also determine the end date for the Azure AD application. When the certificate or the Azure AD application expires, the firewall can no longer use Azure Cloud Integration features until the Azure AD application and the corresponding certificate have been replaced. If a global HTTP proxy is configured, all calls to the Azure REST API are sent via the proxy.

Cloud Integration is required for the following features:

- Barracuda Firewall Admin dashboard cloudinfo element
- UDR route rewriting for CloudGen Firewall high availability clusters
- IP forward protection

Cloud Integration Script for Azure PowerShell 4.2.1

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 4.2.1

```
###########################################
# Cloud Integration for NextGen Firewall F
###########################################

$pathToCERfile = 'PATH_TO\arm.cer'
$ADAppName = 'NGF'
# Set the resource group the Azure Route Table is in
$resourceGroupName = 'RESOURCE_GROUP_NAME'
# your subscription ID - the subscription ID must be entered with the dashes,
as displayed by the Login-AzureRmAccount commandlet
$subscriptionID = '/subscriptions/YOURSUBSCRIPTIONID'

# the identifier and role name must both be unique
$identifier = 'http://localhost'
$roleName = 'NGF Role'
```

```
# if required uncomment the following line to be prompted to log in
#Login-AzureRmAccount

############################################################
# You should not have to change settings after this line
############################################################

# stop script on error
$ErrorActionPreference = 'Stop'

#create custom role
$role = Get-AzureRmRoleDefinition "Virtual Machine Contributor"
$role.Id = $null
$role.Name = $roleName
$role.Description = "Barracuda NextGen Firewall Cloud Integration and UDR
route rewriting"
$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)
# Extract the expiration date from the certificate. Must be valid at least
one year
$endDate = [System.DateTime]::Parse($cert.GetExpirationDateString())

## subtract a day to ensure valid EndDate value
$validNumDays = 1
$timespan = New-TimeSpan -Days $validNumDays
$endDate = $endDate - $timespan

# convert the certificate to a base64 encoded string
$key = [System.Convert]::ToBase64String($cert.GetRawCertData())

# Create the Azure AD Application
$app = New-AzureRmADApplication -DisplayName $ADAppName -HomePage $identifier
-IdentifierUris $identifier -CertValue $key -EndDate $endDate
Write-Host ('Application ID created')

#Create Azure AD Service Principal
$princ = New-AzureRmADServicePrincipal -ApplicationId $app.ApplicationId -
```

```
Verbose
Write-Host ('ServicePrincipalName created...')

Start-Sleep -Seconds 30

# if this step fails increase the Start-Sleep to 120 seconds or just execute
the last command again after waiting a bit
New-AzureRmRoleAssignment -RoleDefinitionName $firewallRole.Name -
ServicePrincipalName $princ.ServicePrincipalNames[0]

# Get the required IDs to configure the service on the firewall
Write-Host ('Use the following information to configure Azure Cloud
Integration on your NextGen Firewall F:')
Write-Host ('Subscription ID ''{0}''' -f (Get-AzureRmSubscription).Id)
Write-Host ('Tenant ID ''{0}''' -f (Get-AzureRmSubscription).TenantId)
Write-Host ('Application ID ''{0}''' -f $app.ApplicationId)
```

Cloud Integration Script 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
```

Cloud Integration 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
```

Cloud Integration 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

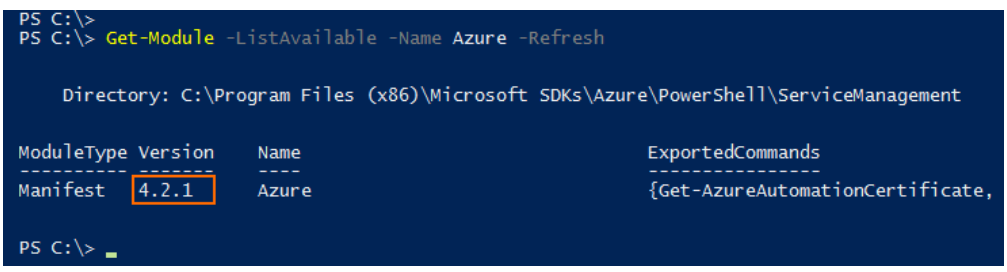
- Deploy your CloudGen Firewall, and configure Azure UDR using the **Azure Resource Manager** (ARM).
- Verify that you are using **Azure PowerShell 4.2.1** or higher.
- Verify that a DNS server is configured. For more information, see How to Configure DNS Settings.
- Log into your Azure account using `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*.

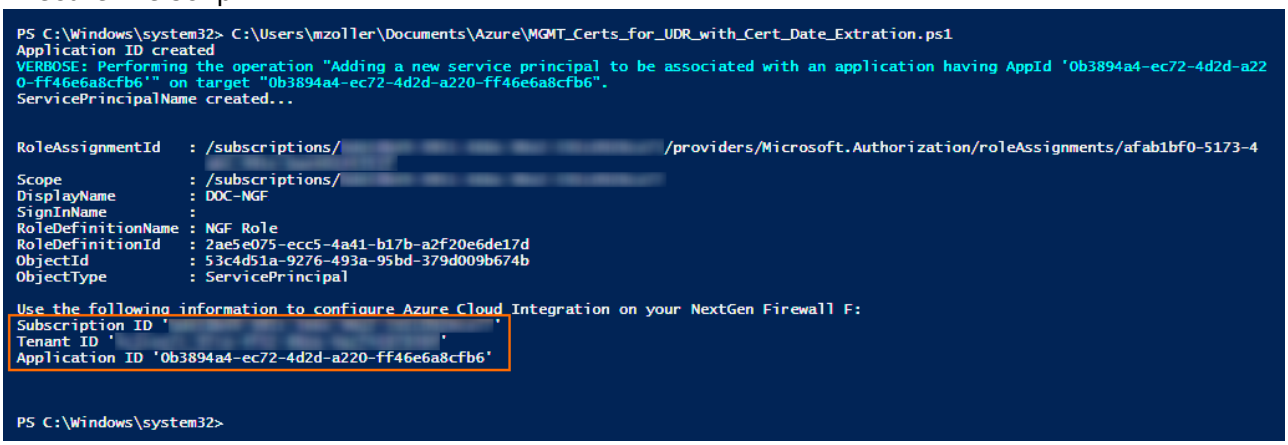
Step 3. Upload the Azure Management Certificate via Azure PowerShell

The values **role name, identifierURis** and **HomePage** URLs must be unique. The **subscription ID** must be entered in the following format: `"/subscriptions/abcdefg1234567891011212"`.

1. Edit the Cloud Integration PowerShell script matching your Azure PowerShell version to set the following variables:
   - **$pathToCERfile** – Enter the path to the certificates created in Step 2. E.g., `'c:\Azure\certs\arm.cer'`
   - **$ADAppName** – Enter a unique name for the ADAppName.
   - $**resourceGroupName** – Enter the name of the Azure resource group containing the VNET.
   - **$subscriptionID** – Enter the Azure SubscriptionID in the following format: `'/subscriptions/YOURSUBSCRIPTIONID'`. Use **Get-AzureRmSubscriptionID** to get your Azure subscription ID.

- **$identifier** – Enter an identifier in the format `'http://localhost'`. The identifier must be unique.
- **$roleName** – Enter a unique name for the role.

2. Execute the script.



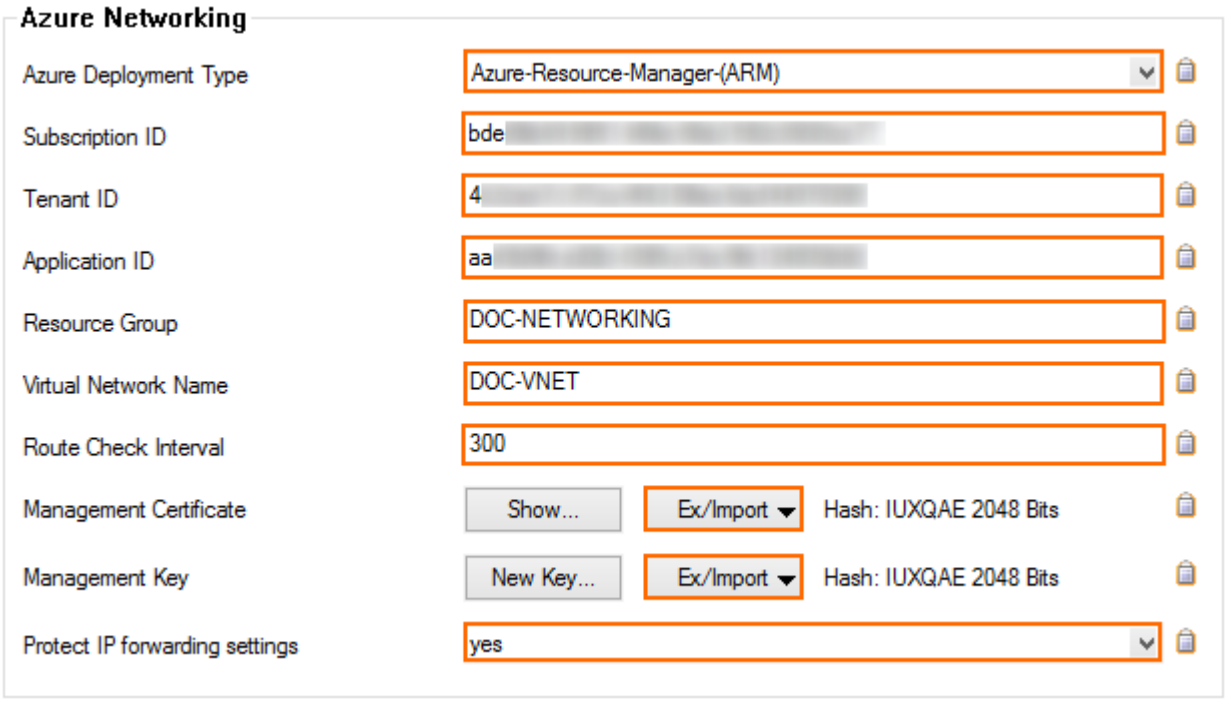
Write down the Subscription ID (without the **/subscription/** prefix), the Tenant ID, and Application ID for use in the firewall configuration.

Step 4. Configure User-Defined Routing and IP Forward Protection on the Firewall

You must enter your Azure ARM IDs and upload the management certificate created in Step 2 and 3 to allow the 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**.
   Only enter the id, do not include the `/subscription/` prefix.
6. Enter your Azure **Tenant ID**.
7. Enter your Azure **Application ID**.
8. Enter the **Resource Group** name the VNET is in. This is the same resource group as entered in the script in Step 3.
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**.

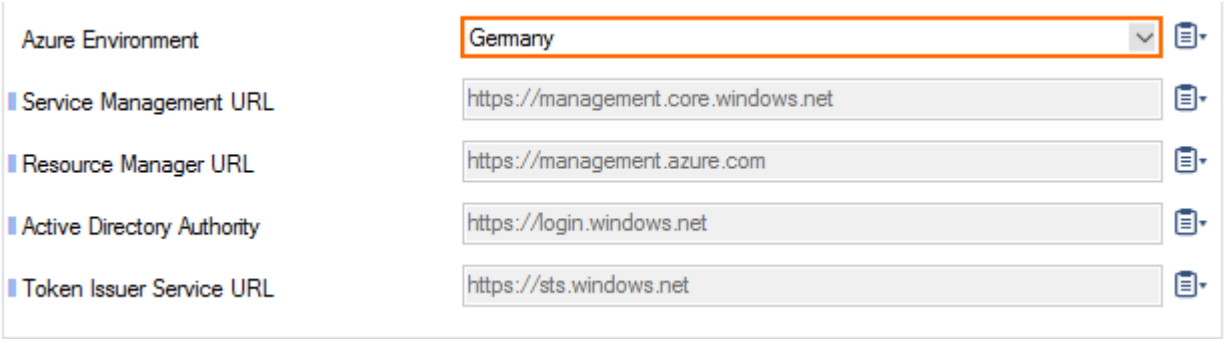
The Azure routing table is now updated every time the virtual server fails over.

Step 5. (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**.

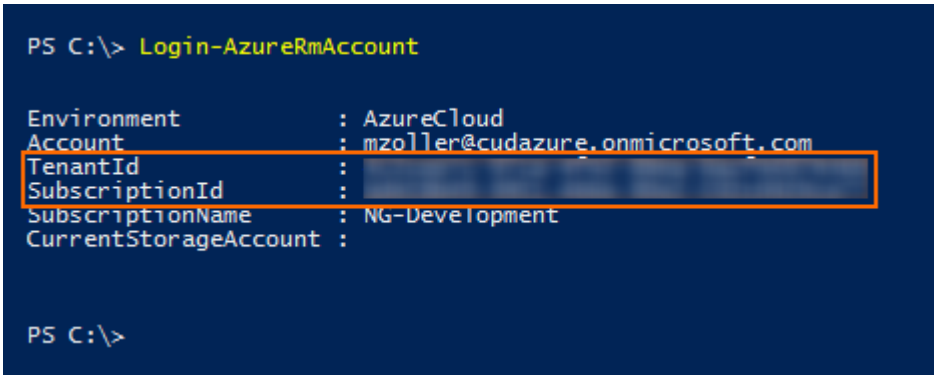
Next steps

For managed high availability clusters repeat Step 4 and, optionally, Step 5 for the other firewall VM in the high availability cluster, or use a repository entry to share the configuration across the cluster. For standalone high availability clusters, the settings are propagated automatically.

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.

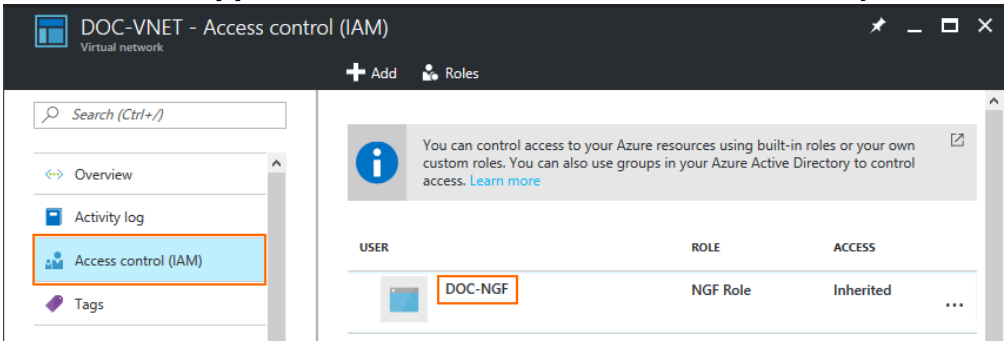
1. Launch Azure PowerShell.
2. The **SubscriptionId** and **TenantId** are listed after logging in via the **Login-AzureRmAccount** commandlet.



Getting the Application ID for Existing Setups

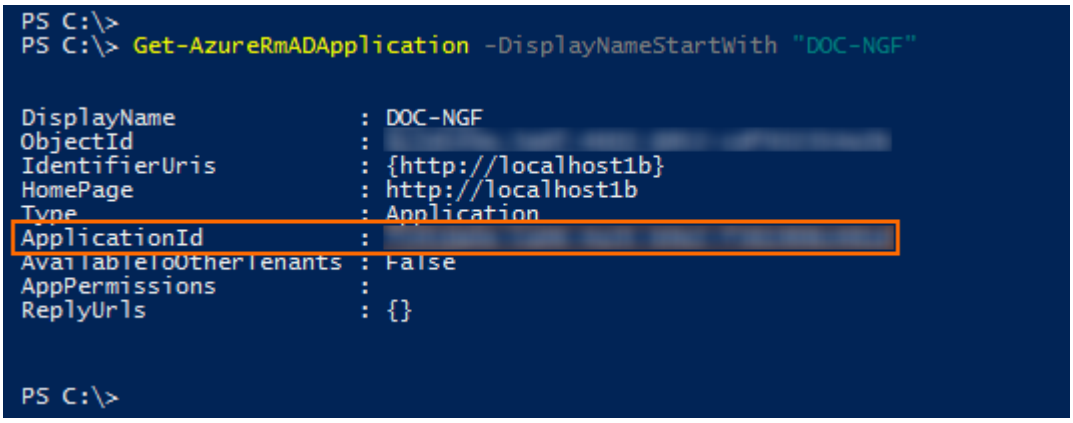
It might take a couple of minutes for the user to be propagated in Azure AD.

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.



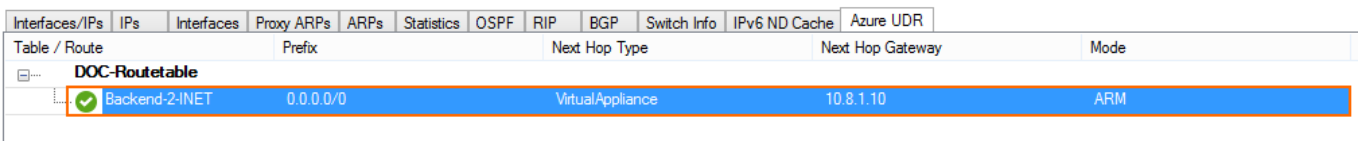
3. Launch Azure PowerShell.
4. Retrieve the **ADApplication** using the username:

```
Get-AzureRmADApplication -DisplayNameStartWith "YOUR_ADAPPNAME"
```

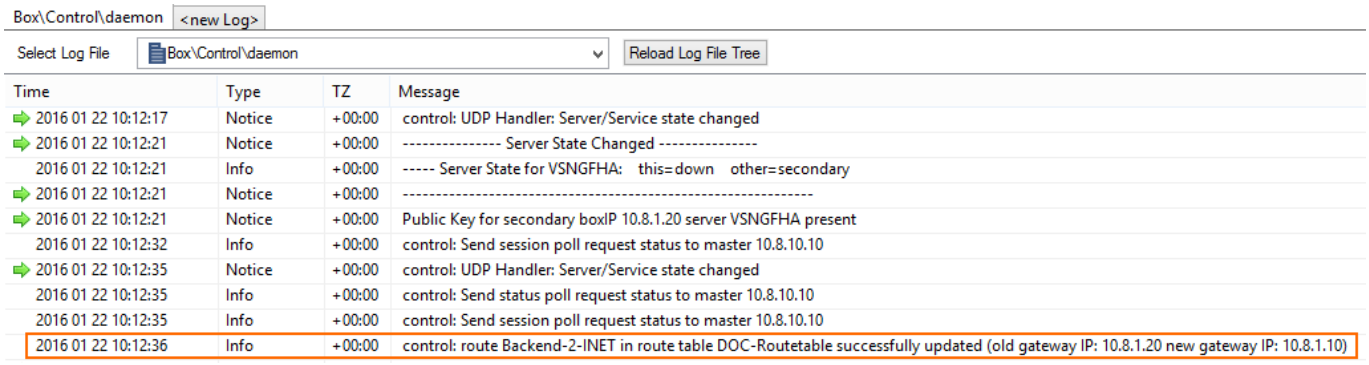


Monitoring

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 next hop 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. cert_upload_script_output.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