Configuring Automated UDR Failover

For Palo Alto Networks VM-Series Firewalls on Microsoft Azure

This solution provides automated failover of User-Defined Route Table entries on Microsoft Azure for Palo Alto Networks VM-Series Firewalls. The automated failover code is deployed to an [Azure Function App](https://docs.microsoft.com/en-us/azure/azure-functions/) within the same Azure subscription as the VM-Series Firewall deployment.

# Pre-requisites

Prior to implementing the solution in this document, an existing deployment of two Palo Alto Networks VM-Series Firewalls in an Availability Set on an Azure VNET is required. If you do not yet have VM-Series firewalls deployed, use the deployment template linked below to deploy on an existing VNET:

* <https://github.com/robotechredmond/ha-nva>

# Create Service Principal in Azure Active Directory

The Azure Function App used in this solution will authenticate to Azure Active Directory using Service Principal credentials. Use the steps below to create a new Service Principal:

* <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-create-service-principal-portal#create-an-azure-active-directory-application>

# Assign Permissions to Service Principal for Azure Resources

The Service Principal with which the Azure Function App authenticates requires specific role-based access control (RBAC) permissions to Azure resources as follows:

* Resource Group containing VM-Series Firewall VMs: Reader role
* Resource Group(s) containing Route Table resources: Contributor role

Use the steps below to assign these RBAC permissions to the Service Principal for each Resource Group:

* <https://docs.microsoft.com/en-us/azure/active-directory/role-based-access-control-configure>

Note that the Resource Group(s) containing Route Table resources can exist in the same Azure subscription as where the VM-Series firewalls are deployed or a different Azure subscription, as long as all Azure subscriptions exist under a common Azure AD Tenant.

# Apply Resource Tag to Azure Route Table resources

The Azure Function App deployed in this solution will automate the failover of user-defined route table entries with a next hop address set to the IP address of an interface on the first VM-Series firewall. A specific Resource Tag Name and Value will need to be set for each Route Table resource for which the Azure Function App will manage automated failover, as follows:

* Resource Tag name: pan\_ha\_udr
* Resource Value: <case-sensitive user defined value for this deployment>

Use the steps below to configure this Resource Tag Name and Value on each Route Table resource:

* <https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-using-tags#portal>

Note that these Route Table resources can exist in the same Azure subscription as where the VM-Series firewalls are deployed or a different Azure subscription, as long as all Azure subscriptions exist under a common Azure AD Tenant.

# Gather Information for Function App Configuration

Before configuring the Function App settings, locate values for the following information specific to this deployment:

* Application ID of Service Principal (defined in prior step)
* Key Value of Service Principal (copied in prior step)
* Azure Active Directory Tenant ID (copied in prior step)
* Azure Subscription ID for Azure subscription in which VM-Series Firewalls are deployed
* Azure VM Name of first VM-Series Firewall
* Azure VM Name of second VM-Series Firewall
* Azure Resource Group Name for the Resource Group containing the VM-Series Firewall resources
* Resource Tag Value for the value assigned to the “pan\_ha\_udr” Resource Tag on each Resource Group containing Route Table Resources (defined in prior step)

# Create Azure Function App

Use the steps below to create a new Azure Function App in the same Azure subscription as where the VM-Series Firewalls are deployed:

* <https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-azure-function#create-a-function-app>

Using these steps, only create the Azure Function App – do not attempt to create or test Function code at this point.

# Configure Azure Function App

In the Azure Portal, navigate to the newly created Function App and then click [*Platform Features*](https://docs.microsoft.com/en-us/azure/azure-functions/functions-how-to-use-azure-function-app-settings#platform-features-tab), and then [*Application Settings*](https://docs.microsoft.com/en-us/azure/azure-functions/functions-how-to-use-azure-function-app-settings#settings) to add the following Application Setting variable values:

* SP\_USERNAME: *< Application ID of Service Principal >*
* SP\_PASSWORD: *< Key Value of Service Principal >*
* TENANTID: *< Azure Active Directory Tenant ID >*
* SUBSCRIPTIONID: *< Azure Subscription ID >*
* AZURECLOUD: *< Either “AzureCloud” or “AzureUSGovernment” >*
* FW1NAME: *< Azure VM Name of first VM-Series Firewall >*
* FW2NAME: *< Azure VM Name of second VM-Series Firewall >*
* FWRGNAME: *< Name of Azure Resource Group containing the VM-Series Firewall resources >*
* FWUDRTAG: *< Resource Tag Value >*
* FWTRIES: 3 (ie., 3 retries for checking firewall health before returning “Down” status)
* FWDELAY: 2 (ie., 2 seconds between retries)
* FWMONITOR: < Either “VMStatus” or “TCPPort” >

If setting FWMONITOR to the value of “TCPPort”, also add the following additional Application Setting variable values:

* FW1FQDN: < Publicly accessible FQDN or IP Address for first VM-Series firewall >
* FW1PORT: < TCP Port on which the first VM-Series firewall is listening >
* FW2FQDN: < Publicly accessible FQDN or IP Address for second VM-Series firewall >
* FW2PORT: < TCP Port on which the second VM-Series firewall is listening >

# Deploy Azure Function App code

In the Azure Portal, navigate to the newly created Function App, and then click [*Platform Features*](https://docs.microsoft.com/en-us/azure/azure-functions/functions-how-to-use-azure-function-app-settings#platform-features-tab), and then *Deployment Options*. Click the *Setup* button to setup a new deployment option with the following configuration:

* Choose Source: *External Repository*
* Repository URL: [*https://github.com/robotechredmond/ha-nva-fo*](https://github.com/robotechredmond/ha-nva-fo)
* Branch: *master*
* Repository Type: *Git*

After setup of this deployment option, click the *Sync* button to deploy the Function code.

**Note:** After deployment and initial testing is completed, click the *Disconnect* button to disconnect the Azure Function App from the deployment repository so that future public repository updates are not automatically deployed.