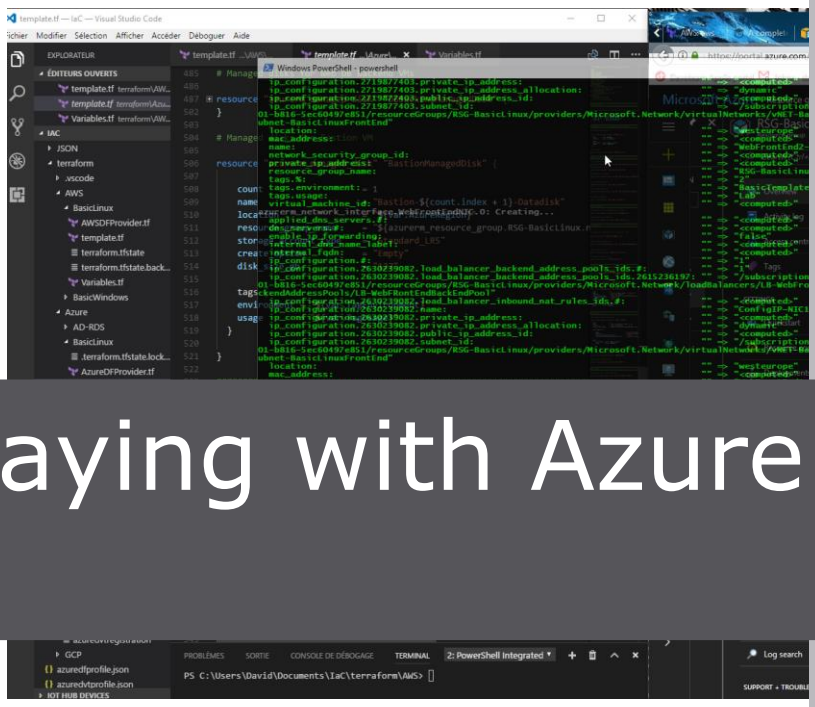


Teknews.cloud

Playing with Azure Network Watcher

David Frappart

21/01/2018

Contributor

Name	Title	Email
David Frappart	Cloud Architect	david@dfitcons.info

Version Control

Version	Date	State
1.0	2018/01/21	Final

Table of Contents

1 Introduction

In this article we will take a look on the Azure Network Watcher service. The aim is mainly to have a first look on the avaialble functionnalities and also on the options / requirements to deploy and use the service.

2 Discovering Azure Network Watcher capabilities

As the name implies, Network watcher is a managed service providing capabilities to montor the Azure Network environment. Microsoft documentation here provides all the required information. To summarize, we have the following options:

Functionnalities	Description
Topology	View the network topology on a specified resource group
Packet capture	Allow packet capture on Azure VMs
IP flow verify	Check if a packet is allowed in the Azure environment
Next hop	Determines the next hop for packets being routed in the Azure Netwok fabric
Security group view	View the applied rules on a VM
NSG Flow logging	View theflow log on NSG (in JSON format)
Virtual Network Gateway and Connection troubleshooting	Tooling for troubleshooting
Network subscription limits	Provide a single pane summarizing the remaining Network resources available in a subscription
Configuring Diagnostics Log	Provides a single pane to enable or disable Diagnostics logs for network resources in a resource group
Connectivity (Preview)	Verifies the possibility of establishing a direct TCP connection from a virtual machine to a given endpoint

That's make quite a few. Also some interesting use case combining Azure services are explored in the documentation, one including the use of Network watcher and Azure function to launch packet capture under specified trigger. Not on the agenda in this article but clearly interesting for an Ops oriented guy wanting to start playing with the functions.

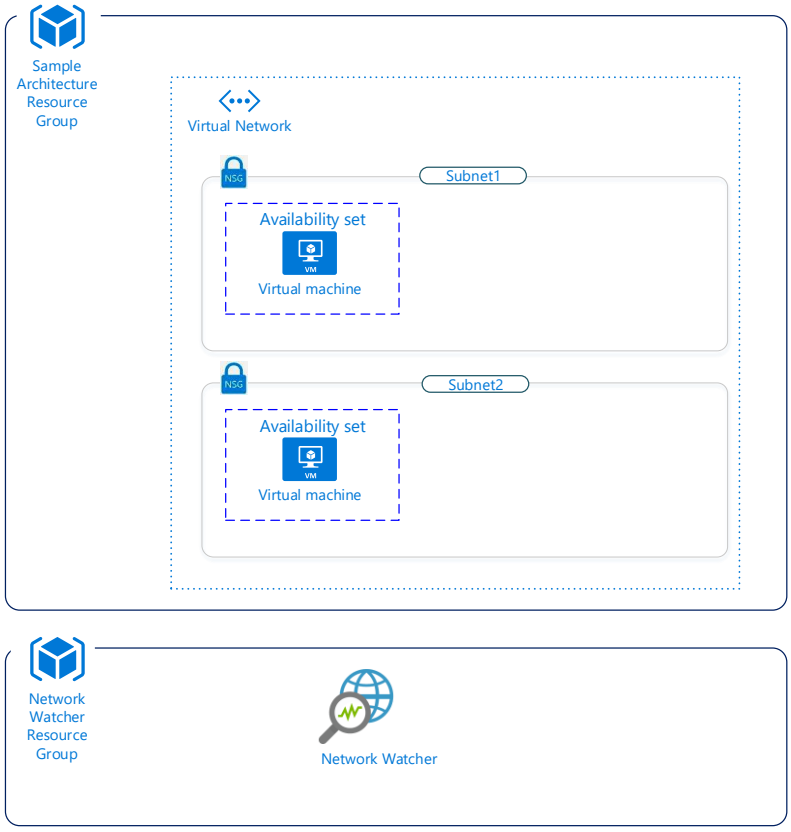
In terms of service scope, the Network watcher is deployed in a resource group and can be activated on a per region basis. The pricing model is based on the volume of Network logs ingested and the number of checks on the Network Diagnostic tools.
The table below, from Microsoft documentation, display the cost associated to Networkwatcher:

FEATURE	MONTHLY ALLOTMENT	OVERAGE CHARGE
Network Logs Ingested	5 GB	€0.422 per GB
Network Diagnostic Tools	1,000 checks	€0.844 per 1,000 checks

3 Deploying a sample architecture with Network Watcher

3.1 Sample architecture Schema

A quite simple architecture is used as a sample. Two subnets included in one vNet. One VM in each subnet. All of those resources deployedin one resource group. Another resource group is created and contains only the Network watcher instance.



3.2 Network flows

A simple Network filtering configuration is put into place with only SSH and RDP from Internet to respectively Subnet1 which contains a Linux VM and Subnet2 which contains a Windows VM. Below are the details for each Network Security Groups:

Network Security Group Subnet 1				
Source IP range	Destination IP range	Protocol	Source port Range	Destination port range
Internet	Subnet1 IP Range	TCP	*	22
Subnet1 IP Range	Subnet 2 IP Range	*	*	*

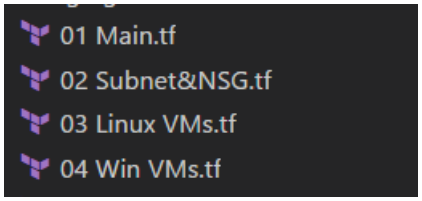
Subnet2 IP Range	Subnet1 IP Range	*	*	*
Subnet1 IP Range	Internet	*	*	*

Network Security Group Subnet 2				
Source IP range	Destination IP range	Protocol	Source port Range	Destination port range
Internet	Subnet2 IP Range	TCP	*	3389
Subnet2 IP Range	Subnet1 IP Range	*	*	*
Subnet1 IP Range	Subnet2 IP Range	*	*	*
Subnet2 IP Range	Internet	*	*	*

3.3 Terraform template for the sample architecture

3.3.1 Template organization

As usual, we separate the coded resources in different files. We have the following files:



- The main file defines the basics resources such as the Resource Groups, the virtual network and also the storage accounts used to store the log files and to provides a share file if needed. Also the Network watcher object is declared in this file.
- The Subnet and NSG file defines the 2 subnets described in the architecture with the 2 associated Network Security Groups.
- Then the 2 remaining files describe the provision of the 2 VMs, with the corresponding NSG rules and the associated VMs extensions.

3.3.2 Terraform modules

Except for a few exceptions, modules coded in other templates are reused. We describe those modules in the following sections.

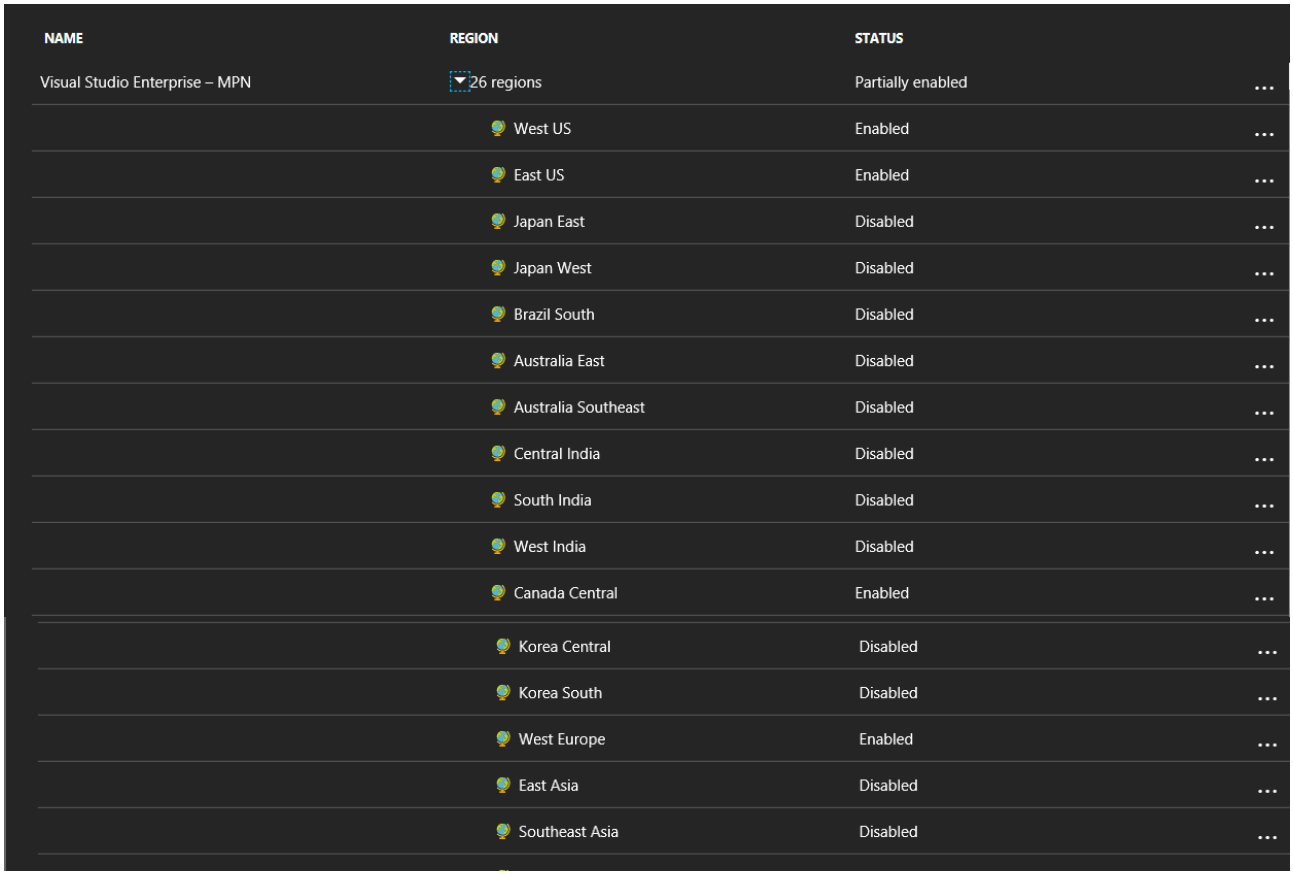
3.3.2.1 Network watcher instance

The Network watcher has been made available in Terraform in version 0.11.x. The module is as follow:

```
#####################################################################
#This module allows the creation of a Network Watcher
#####################################################################

#Variable declaration for Module

#The NW name
variable "NWName" {
  type    = "string"

}

#The RG in which the AS is attached to
variable "RGName" {
  type    = "string"

}

#The location in which the AS is attached to
variable "NWLocation" {
  type    = "string"

}




#Tag value to help identify the resource.
#Required tag are EnvironmentTAg defining the type of
#environment and
#environment Tag usage specifying the use case of the environment

variable "EnvironmentTag" {
  type    = "string"
  default = "Poc"
}

variable "EnvironmentUsageTag" {
  type    = "string"
  default = "Poc usage only"
}



# Availability Set Creation
```

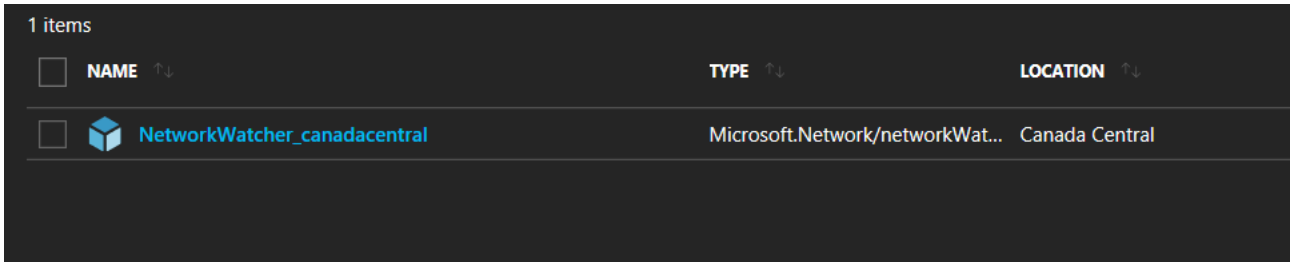
```
resource "azurerm_network_watcher" "Terra_NW" {

    name                    = "${var.NWName}"
    location                = "${var.NWLocation}"
    resource_group_name     = "${var.RGName}"

    tags {
        environment = "${var.EnvironmentTag}"
        usage       = "${var.EnvironmentUsageTag}"
  }
}

#Output


output "Name" {

  value = "${azurerm_network_watcher.Terra_NW.name}"
}

output "Id" {

  value = "${azurerm_network_watcher.Terra_NW.id}"
}
```

The Network watcher instance is deployed in the targeted Resource Group. The location (Azure Region) of this Resource Group determines which region is activated for Network Watcher scope. Currently, Terraform does not allow us to activate more than one region at the creation. Also, in Microsoft documentation, it is mentioned that one Network Watcher instance only is available per region per subscription. It can be confusing then when accessing Azure Portal to see in the Network Watcher blade that only one Network Watcher service appears, with the details on which region the service is activated.



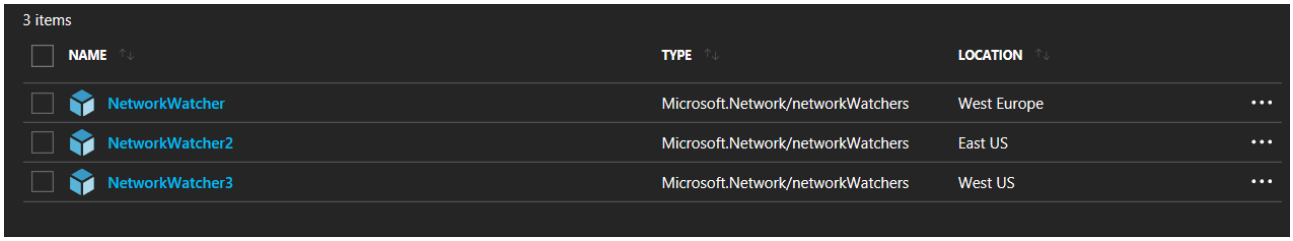
NAME	REGION	STATUS
Visual Studio Enterprise – MPN	26 regions	Partially enabled
	West US	Enabled
	East US	Enabled
	Japan East	Disabled
	Japan West	Disabled
	Brazil South	Disabled
	Australia East	Disabled
	Australia Southeast	Disabled
	Central India	Disabled
	South India	Disabled
	West India	Disabled
	Canada Central	Enabled
	Korea Central	Disabled
	Korea South	Disabled
	West Europe	Enabled
	East Asia	Disabled
	Southeast Asia	Disabled

On the print screen, the Network watcher service is activated in West US, East US, Canada Central and West Europe. The West Europe service was activated using Terraform and specifying the Network Watcher location in a Resource Group in West Europe.
However, the Canada Central was activated through the portal and an additional dedicated Resource Group was created. On the other hand, the West US, East US were activated through PowerShell. Below are displayed the Network Watcher objects for each region, located in a dedicated Resource Group for the Canada Central and in the Terraform provisioned Resource group for all other regional instance of the Network Watcher service



1 items

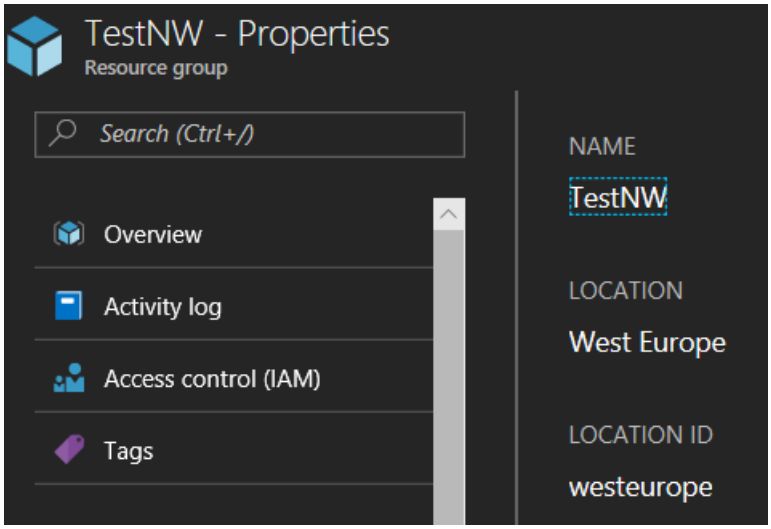
NAME	TYPE	LOCATION
NetworkWatcher_canadacentral	Microsoft.Network/networkWat...	Canada Central



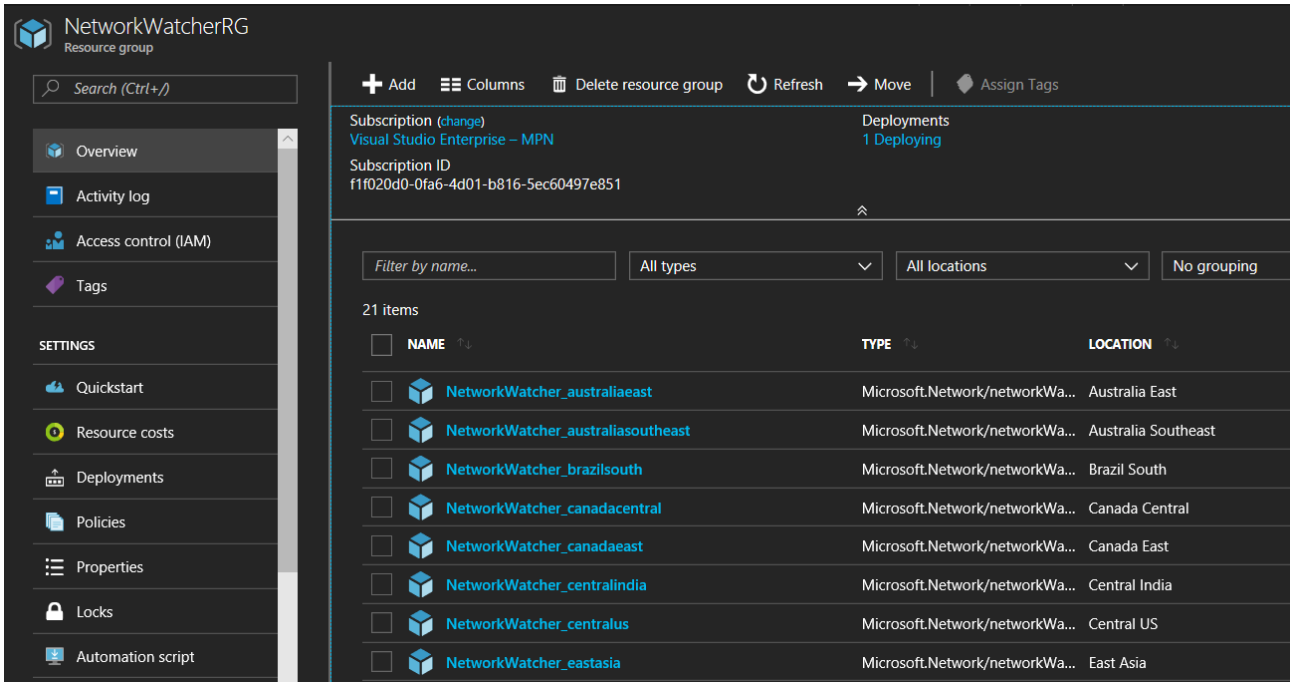
We used the following PowerShell command to activate the service on other region:

```
New-AzureRmNetworkWatcher -ResourceGroupName <Resource_Group_Name> -Name
<NetworkWatcherName> -Location <NetworkWatcherLocation>
```

In the print screen we can see that we have three Network Watcher Object in 3 different Region. However, we created the object in one unique Resource Group. The Resource Group, as you may know is also associated to a Region, as displayed below:



It means that the resource group in this case contains resource from location different than the one it is associated to.
In the same logic, from the portal, in the Network Watcher blade, it is possible to activate the service on all region at the same time. The result of this action is that Network Watcher object for each region are created in the same Resource Group that was created when activating the first Network Object from the portal. The result is displayed below:



Last, when trying to create a Network Watcher in a region for which it already exists, we encounter an error, in accordance to the limit of one instance per region per subscription described in the Azure documentation. The error is displayed below:

```
PS C:\Users\User1\Documents> New-AzureRmNetworkWatcher -ResourceGroupName testnw -
Name NetworkWatcher5 -Location westus
New-AzureRmNetworkWatcher : Cannot create more than 1 network watchers for this
subscription in this region.
StatusCode: 400
ReasonPhrase: Bad Request
OperationID : 'dddc70a5-c6e7-4fa3-87a9-260384c17e53'
Au caractère Ligne:1 : 1
+ New-AzureRmNetworkWatcher -ResourceGroupName testnw -Name NetworkWatc ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    + CategoryInfo          : CloseError : (:) [New-AzureRmNetworkWatcher],
NetworkCloudException
    + FullyQualifiedErrorId :
Microsoft.Azure.Commands.Network.NewAzureNetworkWatcherCommand
New-AzureRmNetworkWatcher : Cannot create more than 1 network watchers for this
subscription in this region.
StatusCode: 400
ReasonPhrase: Bad Request
OperationID : 'dddc70a5-c6e7-4fa3-87a9-260384c17e53'
Au caractère Ligne:1 : 1
+ New-AzureRmNetworkWatcher -ResourceGroupName testnw -Name NetworkWatc ...
+ ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
    + CategoryInfo          : CloseError : (:) [New-AzureRmNetworkWatcher],
NetworkCloudException
    + FullyQualifiedErrorId :
Microsoft.Azure.Commands.Network.NewAzureNetworkWatcherCommand
```

While it does not seem to cause any issue, in my opinion, it may be better to separate the regional instance of Network Watcher in different resource groups located in the same region as the Network Watcher Instance. It would logically ease the delegation of access.
One last point that I am not currently sure about is the billing implication that have the different instances of the service for each region. Does it mean that we have for each region a different instance and thus a free service until 5 Gb of logs and the few 1000's check ? If so it would be quite an interesting product and not necessarily with a high cost associated.
Anyway this question needs to be inquired.

3.3.2.2 Windows VM

In the sample architecture, we deploy 1 VM Linux and 1 VM Windows. For the Linux VM, we deploy with a module detailed in a previous article here. For the Windows VM, we use a similar module. The code is displayed below:

```
##############################################################################################
#This module allows the creation of 1 Windows VM with 1 NIC
##############################################################################################

#Variable declaration for Module

#The VM count
variable "VMCount" {
  type    = "string"

}

#The VM name
variable "VMName" {
  type    = "string"

}


#The VM location
variable "VMLocation" {
  type    = "string"
```

```
}

#The RG in which the VMs are located
variable "VMRG" {
  type    = "string"

}

#The NIC to associate to the VM
variable "VMNICid" {
  type    = "list"

}

#The VM size
variable "VMSize" {
  type    = "string"
  default = "Standard_F1"

}


#The Availability set reference

variable "ASID" {
  type    = "string"

}

#The Managed Disk Storage tier

variable "VMStorageTier" {
  type    = "string"
  default = "Premium_LRS"

}

#The VM Admin Name

variable "VMAdminName" {
  type    = "string"
  default = "VMAdmin"

}
```

```
#The VM Admin Password

variable "VMAdminPassword" {
  type    = "string"

}

# Managed Data Disk reference

variable "DataDiskId" {
  type    = "list"


}

# Managed Data Disk Name

variable "DataDiskName" {
  type    = "list"


}

# Managed Data Disk size

variable "DataDiskSize" {
  type    = "list"

}

# VM images info
#get appropriate image info with the following command
#Get-AzureRMVMImagePublisher -location WestEurope
#Get-AzureRMVMImageOffer -location WestEurope -PublisherName <PublisherName>
#Get-AzureRmVMImageSku -Location westeurope -Offer <OfferName> -PublisherName
<PublisherName>

variable "VMPublisherName" {
  type    = "string"

}
```

```
variable "VMOffer" {
  type    = "string"

}

variable "VMsku" {
  type    = "string"

}

#The boot diagnostic storage uri

variable "DiagnosticDiskURI" {
  type    = "string"

}

#Tag info

variable "EnvironmentTag" {
  type    = "string"
  default = "Poc"
}

variable "EnvironmentUsageTag" {
  type    = "string"
  default = "Poc usage only"
}



#VM Creation


resource "azurerm_virtual_machine" "TerraVMwithCount" {

    count                   = "${var.VMCount}"
    name                    = "${var.VMName}${count.index+1}"
    location                = "${var.VMLocation}"
    resource_group_name     = "${var.VMRG}"
    network_interface_ids   = ["${element(var.VMNICid,count.index)}"]
    vm_size                 = "${var.VMSize}"
    availability_set_id     = "${var.ASID}"

    boot_diagnostics {
```

```
      enabled = "true"
      storage_uri = "${var.DiagnosticDiskURI}"

    }


    storage_image_reference {
        #get appropriate image info with the following command
        #Get-AzureRmVMImageSku -Location westeurope -Offer windowsserver -
PublisherName microsoftwindowsserver
        publisher    = "${var.VMPublisherName}"
        offer        = "${var.VMOffer}"
        sku          = "${var.VMsku}"
        version      = "latest"

    }

    storage_os_disk {

        name                  = "${var.VMName}${count.index+1}-OSDisk"
        caching               = "ReadWrite"
        create_option         = "FromImage"
        managed_disk_type     = "${var.VMStorageTier}"

    }

    storage_data_disk {

        name                  = "${element(var.DataDiskName,count.index)}"
        managed_disk_id       = "${element(var.DataDiskId,count.index)}"
        create_option         = "Attach"
        lun                   = 0
        disk_size_gb          = "${element(var.DataDiskSize,count.index)}"


    }

    os_profile {

        computer_name    = "${var.VMName}${count.index+1}"
        admin_username   = "${var.VMAdminName}"
        admin_password   = "${var.VMAdminPassword}"
```

```
    }

    os_profile_windows_config {

        provision_vm_agent = "true"
        enable_automatic_upgrades = "false"
    }

    tags {
    environment = "${var.EnvironmentTag}"
    usage       = "${var.EnvironmentUsageTag}"
    }


}

#Adding BGInfo to VM

resource "azurerm_virtual_machine_extension" "Terra-BGInfoAgent" {


  count                = "${var.VMCount}"
  name                 = "${var.VMName}${count.index+1}BGInfo"
  location             = "${var.VMLocation}"
  resource_group_name  = "${var.VMRG}"
  virtual_machine_name =
"${element(azurerm_virtual_machine.TerraVMwithCount.*.name,count.index)}"
  publisher            = "microsoft.compute"
  type                 = "BGInfo"
  type_handler_version = "2.1"

      settings = <<SETTINGS
        {

        "commandToExecute": ""
        }
SETTINGS

  tags {
    environment = "${var.EnvironmentTag}"
    usage       = "${var.EnvironmentUsageTag}"
  }
}
```

```
output "Name" {

  value = ["${azurerm_virtual_machine.TerraVMwithCount.*.name}"]
}

output "Id" {

  value = ["${azurerm_virtual_machine.TerraVMwithCount.*.id}"]
}
```

3.3.2.3 Network watcher agent

In order to give the Network Watcher capabilities to check flows on VMs, it requires an agent. We deploy this agent on our VMs with te following module in terraform:

```
#Variable declaration for Module

#The Agent count
variable "AgentCount" {
  type    = "string"

}

#The Agent Name
variable "AgentName" {
  type    = "string"

}

#The Agent Location (Azure Region)
variable "AgentLocation" {
  type    = "string"

}
```

```
#The RG in which the VM resides
variable "AgentRG" {
  type    = "string"

}

#The VM Name
variable "VMName" {
  type    = "list"

}

#Tag info

variable "EnvironmentTag" {
  type    = "string"
  default = "Poc"
}

variable "EnvironmentUsageTag" {
  type    = "string"
  default = "Poc usage only"
}

#Adding Networkwatcher agent

resource "azurerm_virtual_machine_extension" "Terra-NetworkWatcherAgentWin" {


  count                = "${var.AgentCount}"
  name                 = "${var.AgentName}${count.index+1}-NetworkWatcherAgentWin"
  location             = "${var.AgentLocation}"
  resource_group_name  = "${var.AgentRG}"
  virtual_machine_name = "${element(var.VMName,count.index)}"
  publisher            = "microsoft.azure.networkwatcher"
  type                 = "NetworkWatcherAgentWindows"
  type_handler_version = "1.4"

      settings = <<SETTINGS
        {

        "commandToExecute": ""
        }
```

```
SETTINGS

  tags {
    environment = "${var.EnvironmentTag}"
    usage       = "${var.EnvironmentUsageTag}"
  }
}
```

For Linux VMs, it is a little different:

```
resource "azurerm_virtual_machine_extension" "Terra-NEtworkWatcherLinuxAgent" {


  count                = "${var.AgentCount}"
  name                 = "${var.AgentName}${count.index+1}"
  location             = "${var.AgentLocation}"
  resource_group_name  = "${var.AgentRG}"
  virtual_machine_name = "${element(var.VMName,count.index)}"
  publisher            = "Microsoft.Azure.NetworkWatcher"
  type                 = "NetworkWatcherAgentLinux"
  type_handler_version = "1.4"
```

3.4 Configuring access to Network Watcher

3.4.1 Define the required access to Network Watcher service

Until now, access to the Network Watcher service was conducted through an Azure AD Account on which the owners access rights are configured for the Azure Subscription. However, it will probably not be the case in the real world. Microsoft provides in the Network Watcher documentation the required access on all the services for Network Watcher to work. We can then define a custom RBAC role in JSON as follow:

```
{
    "Name": "NetworkWatcherCustomRole",
    "Id": null,
    "IsCustom": true,
    "Description": " Custom Role for access to Network Watcher",
    "Actions": [
      "Microsoft.Storage/*/read",
      "Microsoft.Authorization/*/read",
```

```
      "Microsoft.Resources/subscriptions/resourceGroups/*/read",
      "Microsoft.Storage/storageAccounts/listServiceSas/*/Action",
      "Microsoft.Storage/storageAccounts/listAccountSas/*/Action",
      "Microsoft.Storage/storageAccounts/listKeys/*/Action",
      "Microsoft.Compute/virtualMachines/*/read",
      "Microsoft.Compute/virtualMachines/*/write",
      "Microsoft.Compute/virtualMachineScaleSets/*/read",
      "Microsoft.Compute/virtualMachineScaleSets/*/write",
      "Microsoft.Network/networkWatchers/packetCaptures/*/read",
      "Microsoft.Network/networkWatchers/packetCaptures/*/write",
      "Microsoft.Network/networkWatchers/packetCaptures/*/delete",
      "Microsoft.Network/networkWatchers/*/write",
      "Microsoft.Network/networkWatchers/*/read",
      "Microsoft.Insights/alertRules/*",
      "Microsoft.Support/*"
    ],
    "NotActions": [
    ],
    "AssignableScopes": [
      "/subscriptions/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
    ]
  }
```

3.4.2 Custom role for Network Watcher creation

To create the custom role, we can use PowerShell. In case the appropriate command is not known, it is possible to look for it throught the Get-Command cmdlet as follow:

```
PS C:\WINDOWS\system32> Get-Command *azurermrole*

CommandType     Name                                               Version
Source
-----------     ----                                               -------    -----
-
Cmdlet          Get-AzureRmRoleAssignment                          5.1.1
AzureRM.Resources
Cmdlet          Get-AzureRmRoleDefinition                          5.1.1
AzureRM.Resources
Cmdlet          New-AzureRmRoleAssignment                          5.1.1
AzureRM.Resources
Cmdlet          New-AzureRmRoleDefinition                          5.1.1
AzureRM.Resources
Cmdlet          Remove-AzureRmRoleAssignment                       5.1.1
AzureRM.Resources
```

```
Cmdlet          Remove-AzureRmRoleDefinition                       5.1.1
AzureRM.Resources
Cmdlet          Set-AzureRmRoleDefinition                          5.1.1
AzureRM.Resources
```

The output shows the available command. We proceed with the New-AzureRmRoleDefinition and the help command:

```
PS C:\WINDOWS\system32> help New-AzureRmRoleDefinition -Examples

NOM
    New-AzureRmRoleDefinition

RÉSUMÉ
    Creates a custom role in Azure RBAC. Provide either a JSON role definition file
or a PSRoleDefinition object as input. First, use the
    Get-AzureRmRoleDefinition command to generate a baseline role definition
object. Then, modify its properties as required. Finally, use
    this command to create a custom role using role definition.


    -------------------------  Create using PSRoleDefinitionObject  --------------
------------

    PS C:\> $role = Get-AzureRmRoleDefinition -Name "Virtual Machine Contributor"
              PS C:\> $role.Id = $null
              PS C:\> $role.Name = "Virtual Machine Operator"
              PS C:\> $role.Description = "Can monitor, start, and restart virtual
machines."
              PS C:\> $role.Actions.RemoveRange(0,$role.Actions.Count)
              PS C:\> $role.Actions.Add("Microsoft.Compute/*/read")
              PS C:\>
$role.Actions.Add("Microsoft.Compute/virtualMachines/start/action")
              PS C:\>
$role.Actions.Add("Microsoft.Compute/virtualMachines/restart/action")
              PS C:\>
$role.Actions.Add("Microsoft.Compute/virtualMachines/downloadRemoteDesktopConnectio
nFile/action")
              PS C:\> $role.Actions.Add("Microsoft.Network/*/read")
              PS C:\> $role.Actions.Add("Microsoft.Storage/*/read")
              PS C:\> $role.Actions.Add("Microsoft.Authorization/*/read")
              PS C:\>
$role.Actions.Add("Microsoft.Resources/subscriptions/resourceGroups/read")
```

```
              PS C:\>
$role.Actions.Add("Microsoft.Resources/subscriptions/resourceGroups/resources/read"
)
              PS C:\> $role.Actions.Add("Microsoft.Insights/alertRules/*")
              PS C:\> $role.Actions.Add("Microsoft.Support/*")
              PS C:\> $role.AssignableScopes.Clear()
              PS C:\> $role.AssignableScopes.Add("/subscriptions/ xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx")

              PS C:\> New-AzureRmRoleDefinition -Role $role




    -------------------------  Create using JSON file  -------------------------

    PS C:\> New-AzureRmRoleDefinition -InputFile C:\Temp\roleDefinition.json
```

Since the JSON file is already created, we proceed with the corresponding example:

```
PS C:\WINDOWS\system32> New-AzureRmRoleDefinition -InputFile
C:\Users\User1\Documents\NetworkWatcherCustomRole.json


Name             : NetworkWatcherCustomRole
Id               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
IsCustom         : True
Description      : Custom Role for access to Network Watcher
Actions          : {Microsoft.Storage/*/read, Microsoft.Authorization/*/read,
Microsoft.Resources/subscriptions/resourceGroups/*/read,
                   Microsoft.Storage/storageAccounts/listServiceSas/*/Action...}
NotActions       : {}
AssignableScopes : {/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

After creation, it is possible to check the existence of the RBAC custom role with the following command:

```
PS C:\WINDOWS\system32> Get-AzureRmRoleDefinition | ? {$_.iscustom -eq $true}


Name             : NetworkWatcherCustomRole
Id               : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
IsCustom         : True
Description      : Custom Role for access to Network Watcher
```

```
Actions          : {Microsoft.Storage/*/read, Microsoft.Authorization/*/read,
Microsoft.Resources/subscriptions/resourceGroups/*/read,
                   Microsoft.Storage/storageAccounts/listServiceSas/*/Action...}
NotActions       : {}
AssignableScopes : {/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

3.4.3 Scoping and assigning the custom role

Now only remains the association of the role to the subscription level, so that users or / and group assigned the role should have the required credentials to use Network Watcher. We perform this action with the cmdlet

```
PS C:\Users\User1> New-AzureRmRoleAssignment -ObjectId a5bebb79-6b55-4a17-b258-
8cfa67cb0817 -RoleDefinitionName NetworkWatcherCustomRole -S
ope "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/"


RoleAssignmentId   : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/providers/Microsoft.Authorization/roleAssignments/xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx
Scope              : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
DisplayName        : NetworkWatcherOperators
SignInName         :
RoleDefinitionName : NetworkWatcherCustomRole
RoleDefinitionId   : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
ObjectId           : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
ObjectType         : Group
CanDelegate        : False
```

4 Playing a little with Network Watcher

4.1 Test access from a user with the specific RBAC role assigned

In previous section we defined the Custom RBAC role with the required access right as described on MS documentation. Now we can try to access the Network Watcher with a dedicated user assigned to the custom RBAC role.
The truth here is that the Network topology and some other features of the Network Watcher are not accessible with the Custom Role built from Microsoft Documentation.
To troubleshoot the access, we add the additional rights:

```
Microsoft.Network/networkWatchers/*
Microsoft.Network/networkWatchers/*/Action
```

Then to modify the custom role we proceed the steps as defined below:

- Get the Custom role to modify and store it in a variable

```
PS C:\Users\User1> $role = Get-AzureRmRoleDefinition "NetworkWatcherCustomRole"
PS C:\Users\User1> $role


Name             : NetworkWatcherCustomRole
Id               : 15983214-2748-4202-b9eb-c372770c64f4
IsCustom         : True
Description      : Custom Role for access to Network Watcher
Actions          : {Microsoft.Storage/*/read, Microsoft.Authorization/*/read,
Microsoft.Resources/subscriptions/resourceGroups/*/read,
                   Microsoft.Storage/storageAccounts/listServiceSas/*/Action...}
NotActions       : {}
AssignableScopes : {/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

- List the role's associated actions

```
PS C:\Users\Usr1> $role.actions
Microsoft.Storage/*/read
Microsoft.Authorization/*/read
Microsoft.Resources/subscriptions/resourceGroups/*/read
Microsoft.Storage/storageAccounts/listServiceSas/*/Action
Microsoft.Storage/storageAccounts/listAccountSas/*/Action
Microsoft.Storage/storageAccounts/listKeys/*/Action
Microsoft.Compute/virtualMachines/*/read
Microsoft.Compute/virtualMachines/*/write
Microsoft.Compute/virtualMachineScaleSets/*/read
Microsoft.Compute/virtualMachineScaleSets/*/write
Microsoft.Network/networkWatchers/packetCaptures/*/read
Microsoft.Network/networkWatchers/packetCaptures/*/write
Microsoft.Network/networkWatchers/packetCaptures/*/delete
Microsoft.Network/networkWatchers/*/write
Microsoft.Network/networkWatchers/*/read
```

```
Microsoft.Insights/alertRules/*
Microsoft.Support/*
```

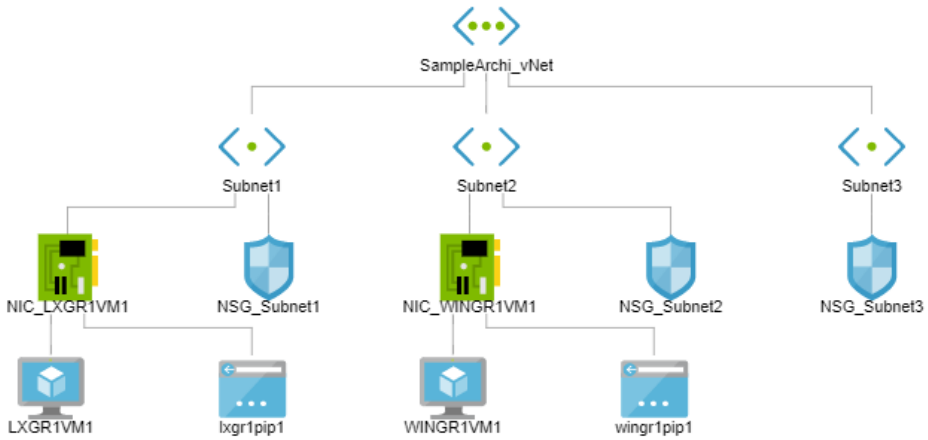
- Add the required new actions

```
PS C:\Users\User1> $role.Actions.Add("Microsoft.Network/networkWatchers/*")
PS C:\Users\User1> $role.Actions.Add("Microsoft.Network/networkWatchers/*/Action")
PS C:\Users\User1> Set-AzureRmRoleDefinition -Role $role


Name             : NetworkWatcherCustomRole
Id               : 15983214-2748-4202-b9eb-c372770c64f4
IsCustom         : True
Description      : Custom Role for access to Network Watcher
Actions          : {Microsoft.Storage/*/read, Microsoft.Authorization/*/read,
Microsoft.Resources/subscriptions/resourceGroups/*/read,
                   Microsoft.Storage/storageAccounts/listServiceSas/*/Action...}
NotActions       : {}
AssignableScopes : {/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}
```

After this the access to the Network Watcher is OK. The following section tests are performed with a dedicated Azure AD user assigned to the Custom RBAC role.

4.2 Get an Azure Environment Network topology

One of the coolest feature of Network watcher is the capability to create a Network topology diagram on a virtual Network. The feature is available in the portal and display something like this:



It is also possible to use PowerShell to get this topology. However we get a json output which then would need to be converted to a chart. The PowerShell command to get the topology is as follow:

```
PS C:\Users\David\Documents> $nw = Get-AzureRmNetworkWatcher -ResourceGroupName RG-
NetWorkWatcher
PS C:\Users\David\Documents> $topo = Get-AzureRmNetworkWatcherTopology -
TargetResourceGroupName rg-networkwatchertest -NetworkWatcher $nw
PS C:\Users\David\Documents> $topo


Id             : xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
CreatedDateTime : 26/01/2018 20:09:43
LastModified    : 26/01/2018 16:07:42
Resources       : [
                    {
                      "Name": "LXGR1VM1",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.C
                  ompute/virtualMachines/LXGR1VM1",
                      "Location": "westeurope",
                      "Associations": [
                        {
                          "AssociationType": "Contains",
                          "Name": "NIC_LXGR1VM1",
                          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                  /Microsoft.Network/networkInterfaces/NIC_LXGR1VM1"
                        }
                      ]
```

```
                    },
                    {
                      "Name": "WINGR1VM1",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.C
                  ompute/virtualMachines/WINGR1VM1",
                      "Location": "westeurope",
                      "Associations": [
                        {
                          "AssociationType": "Contains",
                          "Name": "NIC_WINGR1VM1",
                          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                  /Microsoft.Network/networkInterfaces/NIC_WINGR1VM1"
                        }
                      ]
                    },
                    {
                      "Name": "NIC_LXGR1VM1",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                  etwork/networkInterfaces/NIC_LXGR1VM1",
                      "Location": "westeurope",
                      "Associations": [
                        {
                          "AssociationType": "Associated",
                          "Name": "LXGR1VM1",
                          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                  /Microsoft.Compute/virtualMachines/LXGR1VM1"
                        },
                        {
                          "AssociationType": "Associated",
                          "Name": "Subnet1",
                          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet1"
                        },
                        {
                          "AssociationType": "Associated",
                          "Name": "lxgr1pip1",
                          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
```

```
                /Microsoft.Network/publicIPAddresses/lxgr1pip1"
                    }
                  ]
                },
                {
                  "Name": "NIC_WINGR1VM1",
                  "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/networkInterfaces/NIC_WINGR1VM1",
                  "Location": "westeurope",
                  "Associations": [
                    {
                      "AssociationType": "Associated",
                      "Name": "WINGR1VM1",
                      "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Compute/virtualMachines/WINGR1VM1"
                    },
                    {
                      "AssociationType": "Associated",
                      "Name": "Subnet2",
                      "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet2"
                    },
                    {
                      "AssociationType": "Associated",
                      "Name": "wingr1pip1",
                      "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Network/publicIPAddresses/wingr1pip1"
                    }
                  ]
                },
                {
                  "Name": "NSG_Subnet1",
                  "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/networkSecurityGroups/NSG_Subnet1",
                  "Location": "westeurope",
                  "Associations": [
                    {
                      "AssociationType": "Associated",
```

```
                        "Name": "Subnet1",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet1"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllSubnet1toInternetOut",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllSubnet1t
oInternetOut"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllFromSubnet1toSubnet2Out",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllFromSubn
et1toSubnet2Out"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllFromSubnet2toSubnet1In",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllFromSubn
et2toSubnet1In"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowSSHFromInternetSubnet1In",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet1/securityRules/AllowSSHFromInte
rnetSubnet1In"
                      }
                    ]
                  },
```

```
                    {
                      "Name": "AllowAllSubnet1toInternetOut",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllSubnet1toInternetOut
",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowAllFromSubnet1toSubnet2Out",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllFromSubnet1toSubnet2
Out",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowAllFromSubnet2toSubnet1In",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet1/securityRules/AllowAllFromSubnet2toSubnet1
In",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowSSHFromInternetSubnet1In",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet1/securityRules/AllowSSHFromInternetSubnet1I
n",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "NSG_Subnet2",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
```

```
                etwork/networkSecurityGroups/NSG_Subnet2",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "Subnet2",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet2"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowRDPromInternetSubnet2In",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet2/securityRules/AllowRDPromInter
netSubnet2In"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllFromSubnet2toSubnet1Out",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllFromSubn
et2toSubnet1Out"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllFromSubnet1toSubnet2In",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllFromSubn
et1toSubnet2In"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "AllowAllSubnet2toInternetOut",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
```

```
/Microsoft.Network/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllSubnet2t
oInternetOut"
                        }
                      ]
                    },
                    {
                      "Name": "AllowRDPromInternetSubnet2In",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet2/securityRules/AllowRDPromInternetSubnet2In
",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowAllFromSubnet2toSubnet1Out",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllFromSubnet2toSubnet1
Out",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowAllFromSubnet1toSubnet2In",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllFromSubnet1toSubnet2
In",
                      "Location": "westeurope",
                      "Associations": []
                    },
                    {
                      "Name": "AllowAllSubnet2toInternetOut",
                      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N

etwork/networkSecurityGroups/NSG_Subnet2/securityRules/AllowAllSubnet2toInternetOut
",
                      "Location": "westeurope",
```

```
                    "Associations": []
                  },
                  {
                    "Name": "NSG_Subnet3",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/networkSecurityGroups/NSG_Subnet3",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "Subnet3",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet3"
                      }
                    ]
                  },
                  {
                    "Name": "lxgr1pip1",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/publicIPAddresses/lxgr1pip1",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "NIC_LXGR1VM1",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Network/networkInterfaces/NIC_LXGR1VM1"
                      }
                    ]
                  },
                  {
                    "Name": "wingr1pip1",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/publicIPAddresses/wingr1pip1",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
```

```
                        "Name": "NIC_WINGR1VM1",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                  /Microsoft.Network/networkInterfaces/NIC_WINGR1VM1"
                      }
                    ]
                  },
                  {
                    "Name": "SampleArchi_vNet",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                  etwork/virtualNetworks/SampleArchi_vNet",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Contains",
                        "Name": "Subnet1",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet1"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "Subnet2",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet2"
                      },
                      {
                        "AssociationType": "Contains",
                        "Name": "Subnet3",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers

/Microsoft.Network/virtualNetworks/SampleArchi_vNet/subnets/Subnet3"
                      }
                    ]
                  },
                  {
                    "Name": "Subnet1",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
```

```
                etwork/virtualNetworks/SampleArchi_vNet/subnets/Subnet1",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "NSG_Subnet1",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Network/networkSecurityGroups/NSG_Subnet1"
                      }
                    ]
                  },
                  {
                    "Name": "Subnet2",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/virtualNetworks/SampleArchi_vNet/subnets/Subnet2",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "NSG_Subnet2",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Network/networkSecurityGroups/NSG_Subnet2"
                      }
                    ]
                  },
                  {
                    "Name": "Subnet3",
                    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.N
                etwork/virtualNetworks/SampleArchi_vNet/subnets/Subnet3",
                    "Location": "westeurope",
                    "Associations": [
                      {
                        "AssociationType": "Associated",
                        "Name": "NSG_Subnet3",
                        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers
                /Microsoft.Network/networkSecurityGroups/NSG_Subnet3"
                      }
                    ]
                  }
```

```
                ]
```

An interesting point of comparison, when comparing the topology in the portal and the topology obtained through PowerShell, we can see that the portal does not display the rules associated while the extract in PowerShell does. A tool to generate chart from this extract should be

4.3 Activating Logging on Network security group

We said in previous sections that Network Watcher is capable of logging flows on NSG. For this to be put in place in PowerShell, we use the following commands:

```
PS C:\Users\User1\Documents> Register-AzureRmResourceProvider -ProviderNamespace
Microsoft.Insights


ProviderNamespace : microsoft.insights
RegistrationState : Registered
ResourceTypes     : {components, webtests, queries, components/pricingPlans...}
Locations         : {East US, South Central US, North Europe, West Europe...}
```

This first command is required to activate the functionality of the NSG flow logging. We then store in variables the Network Watcher object, the network security groups and the target storage account for the logs. If we try to display the $nsglist variable, we get a detailed configuration of each nsg with the associated rules. To avoid a huge output, the use of | ft is preferred.

```
PS C:\Users\User1\Documents> $nw = Get-AzureRmNetworkWatcher -ResourceGroupName rg-
networkwatcher -Name networkwatcher

PS C:\Users\User1\Documents> $nsglist = Get-AzureRmNetworkSecurityGroup -
ResourceGroupName rg-networkwatchertest

PS C:\Users\User1\Documents> $stoalog = Get-AzureRmStorageAccount -
ResourceGroupName rg-networkwatchertest -name fswahdiaglogstorage



PS C:\Users\User1\Documents> $nsglist | ft name

Name
----
NSG_Subnet1
NSG_Subnet2
NSG_Subnet3
```

Then to activate the flow logs, a simple loop can be used as displayed below:

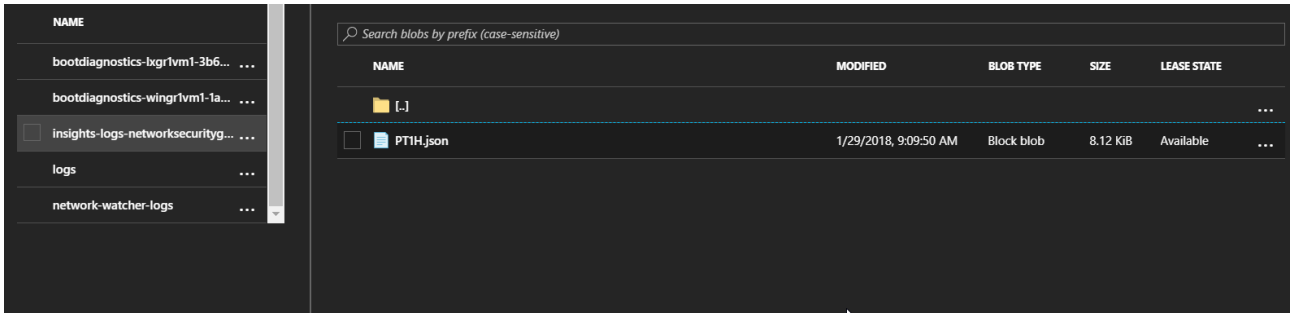
```
PS C:\Users\User1\Documents> foreach($nsg in $nsglist) {Get-
AzureRmNetworkWatcherFlowLogStatus -NetworkWatcher $nw -TargetResourceId $nsg.Id}


TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet1
StorageId        :
Enabled          : False
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }

TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet2
StorageId        :
Enabled          : False
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }

TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet3
StorageId        :
Enabled          : False
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }
```

After this step, another simple loop to activate the logging on the NSG:

```
PS C:\Users\User1\Documents> foreach($nsg in $nsglist) {Set-
AzureRmNetworkWatcherconfigFlowLog -NetworkWatcher $nw -TargetResourceId $nsg.Id -
StorageAccountId $stoalog.id -enableflowlog $true}
```

```
TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet1
StorageId        : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Storage/sto
                   rageAccounts/fswahdiaglogstorage
Enabled          : True
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }

TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet2
StorageId        : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Storage/sto
                   rageAccounts/fswahdiaglogstorage
Enabled          : True
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }

TargetResourceId : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Network/net
                   workSecurityGroups/NSG_Subnet3
StorageId        : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Storage/sto
                   rageAccounts/fswahdiaglogstorage
Enabled          : True
RetentionPolicy  : {
                     "Days": 0,
                     "Enabled": false
                   }
```

After some time we can get the log in the associated storage account:



The log takes the form of a JSON file. Appropriate tools should be selected to exploit this kind of graph. That is however not the subject for this article.

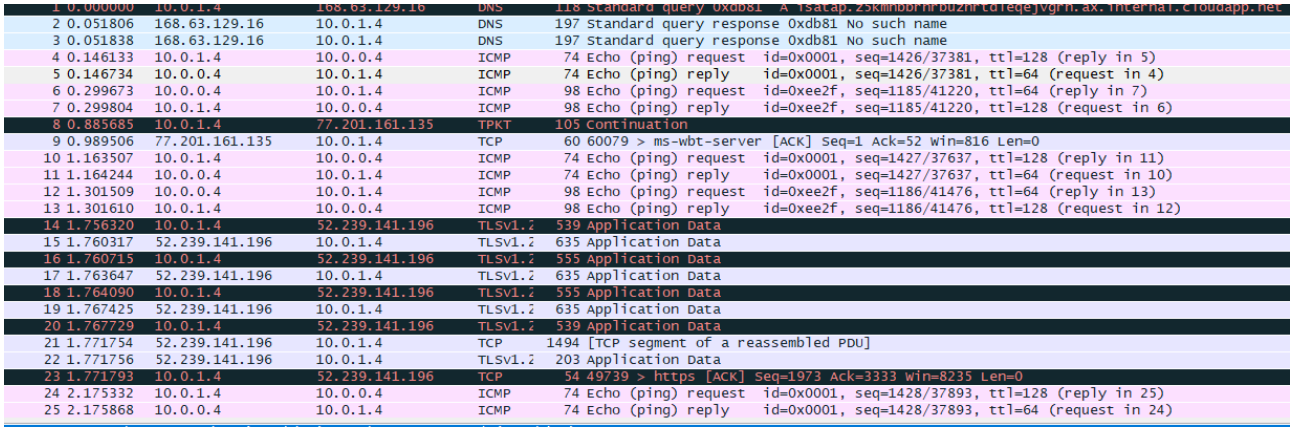
4.4 Capture Network traffic and analyze the result

As discussed earlier, Network Watcher gives the capabilities to perform Network Capture. While it is possible to perform the capture through the portal, the PowerShell cmdlet used is New-AzureRMNetworkWatcherPacketCapture. We do need a storage account. We use the one created in the template We also need the id of a virtual machine on which to perform the capture.

```
PS C:\Users\User1\Documents\>New-AzureRmNetworkWatcherPacketCapture -NetworkWatcher
$nw -PacketCaptureName 20180126testcap -TargetVirtualMachineId
/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
NetWorkWatcherTest/providers/Microsoft.Compute/virtualMachines/LXGR1VM1 -
StorageAccountId $stoalog.id -TimeLimitInSeconds 120


Name                   : 20180126testcap
Id                     : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/rg-networkwatcher/providers/Microsoft.Network/

networkWatchers/NetworkWatcher/packetCaptures/20180126testcap
Etag                   : W/"79c60ca9-c97b-410f-94c8-4ed064a8dccf"
ProvisioningState      : Succeeded
Target                 : /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/providers/Microsoft.Comp
                         ute/virtualMachines/LXGR1VM1
BytesToCapturePerPacket : 0
TotalBytesPerSession   : 1073741824
TimeLimitInSeconds     : 120
StorageLocation        : {
                           "StorageId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-NetWorkWatcherTest/provider
                         s/Microsoft.Storage/storageAccounts/fswahdiaglogstorage",
```

```
                            "StoragePath":
"https://fswahdiaglogstorage.blob.core.windows.net/network-watcher-
logs/subscriptions/f1f020d0-0
                           fa6-4d01-b816-5ec60497e851/resourcegroups/rg-
networkwatchertest/providers/microsoft.compute/virtualmachines/lxgr1
                           vm1/2018/01/26/packetcapture_17_02_07_983.cap"
                           }
Filters                  : []
```

After the capture is complete it is available in the storage account and can be displayed in Wireshark. Here is an extract of a simple capture with ICMP traffic going throught the 2 VMs of our template:



1	0.000000	10.0.1.4	168.63.129.16	DNS	118 Standard query 0xdb81 A 1satap.25kmnbbrhrbu2nrtdreqejvgrh.ax.internal.cloudapp.net
2	0.051806	168.63.129.16	10.0.1.4	DNS	197 Standard query response 0xdb81 No such name
3	0.051838	168.63.129.16	10.0.1.4	DNS	197 Standard query response 0xdb81 No such name
4	0.146133	10.0.1.4	10.0.0.4	ICMP	74 Echo (ping) request id=0x0001, seq=1426/37381, ttl=128 (reply in 5)
5	0.146734	10.0.0.4	10.0.1.4	ICMP	74 Echo (ping) reply id=0x0001, seq=1426/37381, ttl=64 (request in 4)
6	0.299673	10.0.0.4	10.0.1.4	ICMP	98 Echo (ping) request id=0xee2f, seq=1185/41220, ttl=64 (reply in 7)
7	0.299804	10.0.1.4	10.0.0.4	ICMP	98 Echo (ping) reply id=0xee2f, seq=1185/41220, ttl=128 (request in 6)
8	0.885685	10.0.1.4	77.201.161.135	TPKT	105 Continuation
9	0.989506	77.201.161.135	10.0.1.4	TCP	60 60079 > ms-wbt-server [ACK] Seq=1 Ack=52 Win=816 Len=0
10	1.163507	10.0.1.4	10.0.0.4	ICMP	74 Echo (ping) request id=0x0001, seq=1427/37637, ttl=128 (reply in 11)
11	1.164244	10.0.0.4	10.0.1.4	ICMP	74 Echo (ping) reply id=0x0001, seq=1427/37637, ttl=64 (request in 10)
12	1.301509	10.0.0.4	10.0.1.4	ICMP	98 Echo (ping) request id=0xee2f, seq=1186/41476, ttl=64 (reply in 13)
13	1.301610	10.0.1.4	10.0.0.4	ICMP	98 Echo (ping) reply id=0xee2f, seq=1186/41476, ttl=128 (request in 12)
14	1.756320	10.0.1.4	52.239.141.196	TLSv1.2	539 Application Data
15	1.760317	52.239.141.196	10.0.1.4	TLSv1.2	635 Application Data
16	1.760715	10.0.1.4	52.239.141.196	TLSv1.2	555 Application Data
17	1.763647	52.239.141.196	10.0.1.4	TLSv1.2	635 Application Data
18	1.764090	10.0.1.4	52.239.141.196	TLSv1.2	555 Application Data
19	1.767425	52.239.141.196	10.0.1.4	TLSv1.2	635 Application Data
20	1.767729	10.0.1.4	52.239.141.196	TLSv1.2	539 Application Data
21	1.771754	52.239.141.196	10.0.1.4	TCP	1494 [TCP segment of a reassembled PDU]
22	1.771756	52.239.141.196	10.0.1.4	TLSv1.2	203 Application Data
23	1.771793	10.0.1.4	52.239.141.196	TCP	54 49739 > https [ACK] Seq=1973 Ack=3333 Win=8235 Len=0
24	2.175332	10.0.1.4	10.0.0.4	ICMP	74 Echo (ping) request id=0x0001, seq=1428/37893, ttl=128 (reply in 25)
25	2.175868	10.0.0.4	10.0.1.4	ICMP	74 Echo (ping) reply id=0x0001, seq=1428/37893, ttl=64 (request in 24)

5 Conclusion

We explored in this article the Network Watcher service and some of the associated features. Regarding the RBAC capabilities, Terraform now gives us the capabilities to create the role definition and assignment. Also, another interesting feature described in the documentation is the use of a function to launch a Network capture on an alert predefined.
Many interesting topics, to be studied in other blog post.
For those interested, all the code examples are available on GitHub here.

Another Tech blog

My thoughts and experiences on Cloud related tek