

Teknews.cloud

Getting started with Terraform Data Source - Deploy on existing Azure infrastructure

David Frappart

21/01/2018

Contributor

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

Version Control

Version	Date	State
1.0	2018/03/01	Final

Table of Contents

1 Introduction	3
2 The scenario	3
2.1 The foundation architecture	3
2.2 Example of deployment on top of the foundation	5
3 Terraform configuration files	6
3.1 The foundation layers	6
3.1.1 Resource Group Module	6
3.1.2 VNet Module	8
3.1.3 Subnet Module	9
3.1.4 Network Security Group Module	11
3.1.5 Network Security Group Rule Module	13
3.1.6 Storage account module	15
3.1.7 Network Watcher module	18
3.2 Available Data sources in Terraform that can be useful	20
3.2.1 Data source Resource Group	21
3.2.2 Data source VNet	21
3.2.3 Data source subnet	22
3.2.4 Data source Storage Account	22
3.3 Applications of Data source in a VM Deployment template	22
4 Conclusion	51

1 Introduction

The power of the Cloud platforms is the ease with which one can deploy a full infrastructure with Infrastructure as Code tools. Defining an application environment, with the IaaS features, the appropriate managed services and the security requirements is quite easy.

On the other hand, in an extended Datacenter scenario, with either VPN or DirectConnect / ExpressRoute link, the hybrid aspect makes it a little less easy to provision fully the environment. In fact, the Network interconnection will usually be built once, usually due to the fact that the On Premise side is not ready to deliver the required assets for the On premise / Public Cloud interactions.

It can also be a choice to build a Cloud foundation, composed of the basics layers of the infrastructure such as the virtual network tenant and its security feature and the interconnections with the On premise that we discuss earlier.

In those scenarios, Applications will come later, either migrated or built to / on the Cloud platform. So the full deployment with IaC is not applicable here.

In this article, we will take a likewise scenario and make use of Terraform (Again !!!) capabilities to find existing resources in an Azure subscription and deploy new resources on top of it.

So let's get going !

2 The scenario

As described in the introduction, we will take a scenario with a foundation infrastructure in Azure. After that we will get the required reference to deploy VMs on the previously deployed infrastructure.

2.1 The foundation architecture

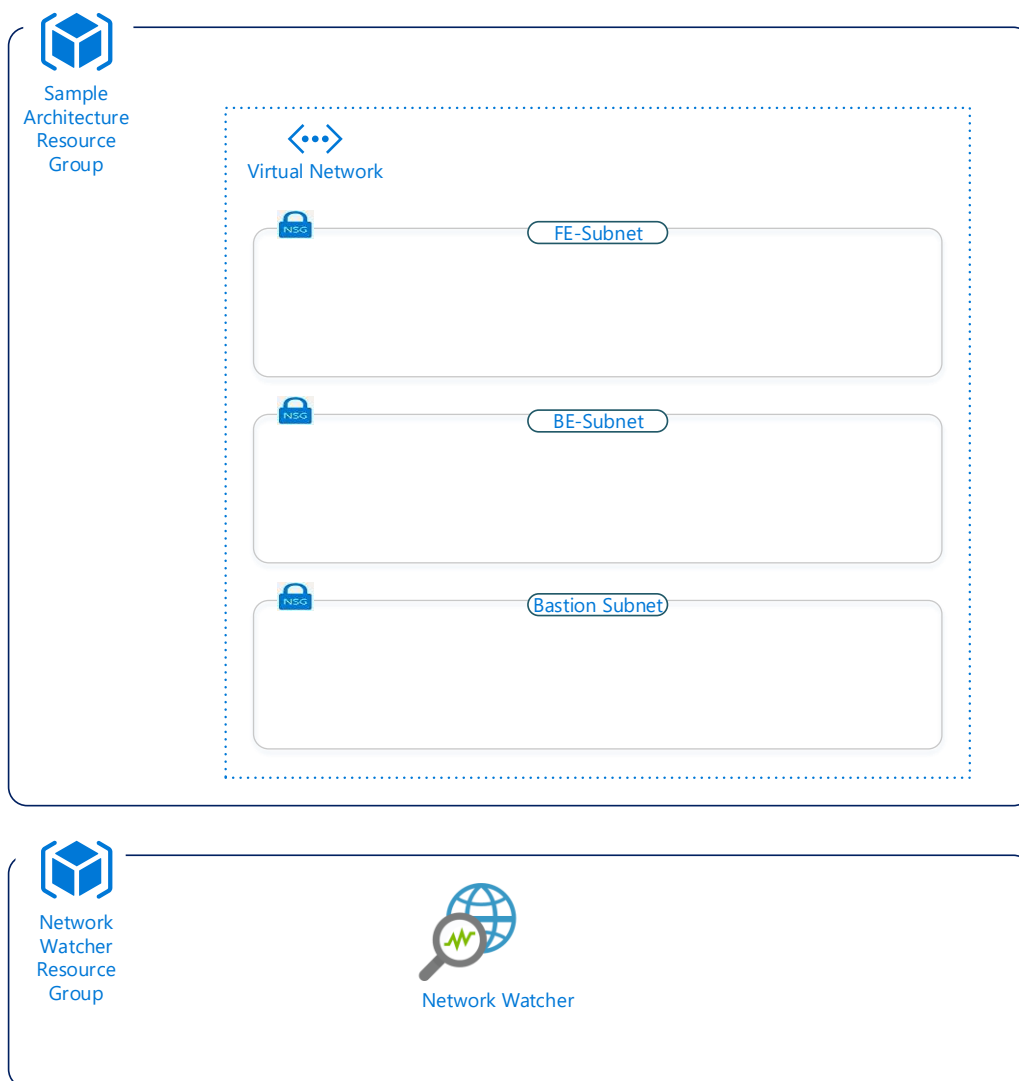
As a target architecture, we will deploy a single VNet, composed of a front-end, a back-end and a bastion subnet. Access through SSH from internet will be allowed only on the Bastion subnet, while only TCP 80 and 443 will be allowed on Front-End.

Inside the VNet, as by Azure Design, all traffic is allowed between all subnet and Outbound to Internet is also allowed by design.

Obviously, everything is regrouped in one resource Group.

In addition, just in case, we also add a network watcher instance in a dedicated resource group.

The architecture looks like this, similar to the architecture taken for the Network watcher article, but without any VMs to start with, with additionally a storage account for boot diagnostic logs, and logs that can be available with the network watcher:

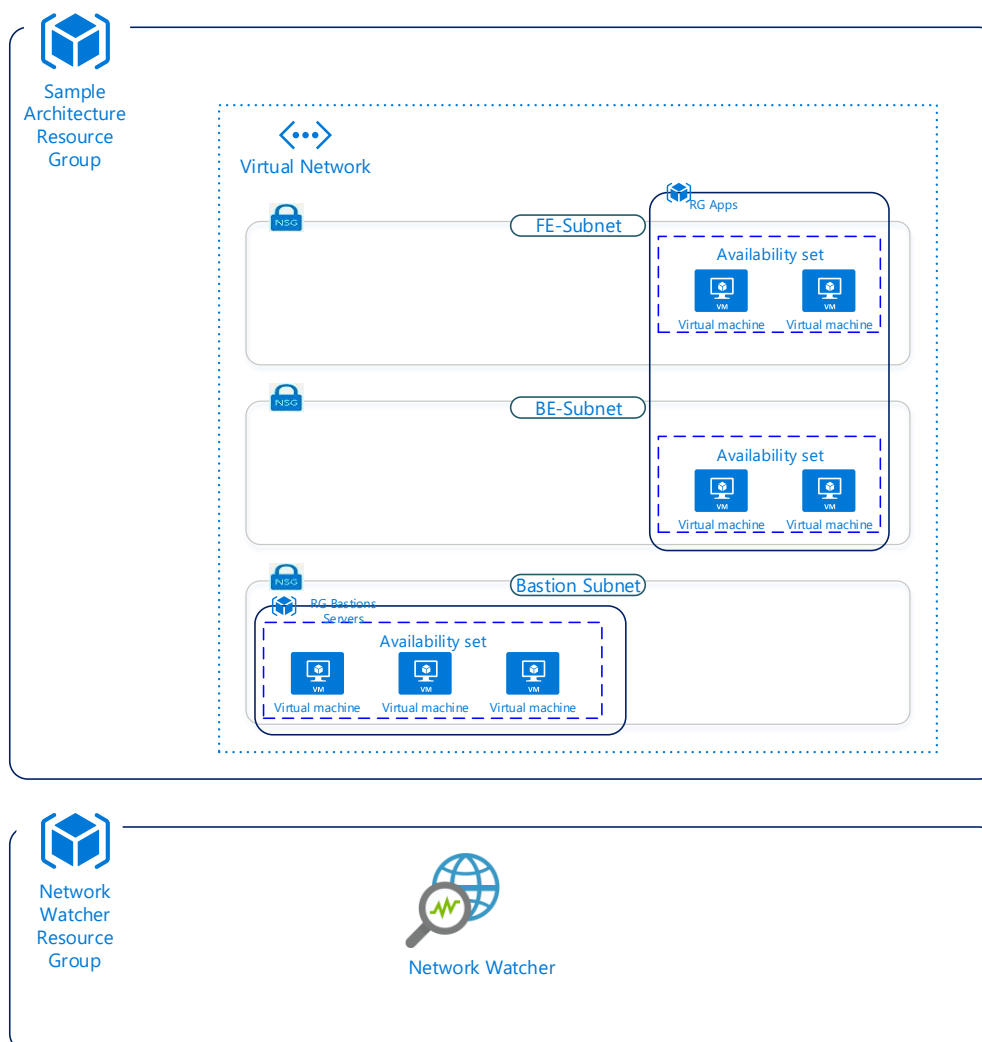


With the Network flows, as described earlier:

Source address	destination address	Protocole	SRC Port	Dest Port	Action	Comment
Internet	FE-Subnet	TCP	*	80,443	Allow	
Internet	Bastion Subnet	TCP	*	22,3389	Allow	Allow SSH and RDP
VNet	VNet	*	*	*	Allow	Exist by design
VNet	Internet	*	*	*	Allow	Exist by design

2.2 Example of deployment on top of the foundation

On a second step (and 3rd, fourth...) we want to be able to provision VMs on top of the existing Network, so we would get something like this:



Each successive deployment is composed of a resource group grouping all associated resources. On this schema, we show only VMs deployed on top of the VNet. Additionally, storage account with Azure file are deployed for each Resource groups

3 Terraform configuration files

Now that the scenario is defined, let's have a look at Terraform configuration files

3.1 The foundation layers

- For the foundation layers, we need the following resources
 - One resource Group with
 - One VNet
 - 3 Subnets
 - 3 security groups, applied to subnets
 - A storage account for Logs (VMs boot diagnostics and Network Watcher logs)
 - Default Security rules as described in the flow matrix
- A second resource group with a Network Watcher instance in it

We will take modules for each objects. Details are exposed in following part

3.1.1 Resource Group Module

The Resource Group Module is displayed below:

```
#####
#This module allows the creation of a RG
#####

#Variable declaration for Module

variable "RGName" {
  type    = "string"
  default = "DefaultRG"
}

variable "RGLocation" {
  type    = "string"
  default = "Westeurope"
}
```

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

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

#Creating a Resource Group
resource "azurerm_resource_group" "Terra-RG" {

  name      = "${var.RGName}"
  location  = "${var.RGLocation}"

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

}

#Output for the RG module

output "Name" {

  value = "${azurerm_resource_group.Terra-RG.name}"
}

output "Location" {

  value = "${azurerm_resource_group.Terra-RG.location}"
}

output "Id" {

  value = "${azurerm_resource_group.Terra-RG.id}"
}
```


3.1.2 VNet Module

The VNet module is displayed below:

```
#####
#This module allows the creation of a vNet
#####

#Variable declaration for Module

variable "vNetName" {
  type    = "string"
  default = "DefaultvNet"
}

variable "RGName" {
  type    = "string"
  default = "DefaultRSG"
}

variable "vNetLocation" {
  type    = "string"
  default = "Westeurope"
}

variable "vNetAddressSpace" {
  type    = "list"
  default = ["10.0.0.0/20"]
}

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

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

#Creating a vNet
```

```
resource "azurerm_virtual_network" "Terra-vNet" {

  name                = "${var.vNetName}"
  resource_group_name = "${var.RGName}"
  address_space       = "${var.vNetAddressSpace}"
  location             = "${var.vNetLocation}"

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

#Output for the vNET module

output "Name" {

  value = "${azurerm_virtual_network.Terra-vNet.name}"
}

output "Id" {

  value = "${azurerm_virtual_network.Terra-vNet.id}"
}

output "AddressSpace" {

  value = "${azurerm_virtual_network.Terra-vNet.address_space}"
}
```

3.1.3 Subnet Module

The subnet module is displayed below:

```
#####
#This module allows the creation of a Subnet
#####

#Variable declaration for Module

variable "SubnetName" {
  type    = "string"
```

```

    default = "DefaultSubnet"
  }

  variable "RGName" {
    type      = "string"
    default = "DefaultRSG"
  }

  variable "vNetName" {
    type      = "string"
  }

  variable "Subnetaddressprefix" {
    type      = "string"
  }

  variable "NSGid" {
    type      = "string"
  }

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

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

  #Creation fo the subnet

  resource "azurerm_subnet" "TerraSubnet" {

    name                        = "${var.SubnetName}"
    resource_group_name        = "${var.RGName}"
    virtual_network_name       = "${var.vNetName}"
  }

```

```

    address_prefix      = "${var.Subnetaddressprefix}"
    network_security_group_id = "${var.NSGid}"
  }

#Output

output "Name" {
    value = "${azurerm_subnet.TerraSubnet.name}"
}

output "Id" {
    value = "${azurerm_subnet.TerraSubnet.id}"
}

output "AddressPrefix" {
    value = "${azurerm_subnet.TerraSubnet.address_prefix}"
}

```

3.1.4 Network Security Group Module

The Network Security Group Module is displayed below:

```

#####
#This module allows the creation of a Netsork Security Group
#####

#Variable declaration for Module

variable "NSGName" {
    type    = "string"
    default = "DefaultNSG"
}

```

```
variable "RGName" {
  type      = "string"
  default   = "DefaultRSG"
}

variable "NSGLocation" {
  type      = "string"
  default   = "Westeurope"
}

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

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

#Creation fo the NSG
resource "azurerm_network_security_group" "Terra-NSG" {

  name          = "${var.NSGName}"
  location      = "${var.NSGLocation}"
  resource_group_name = "${var.RGName}"

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

#Output for the NSG module

output "Name" {

  value = "${azurerm_network_security_group.Terra-NSG.name}"
}

output "Id" {

  value = "${azurerm_network_security_group.Terra-NSG.id}"
}
```

```
}
```

3.1.5 Network Security Group Rule Module

The Network Security Group Rules module is displayed below:

```
#####
#This module allows the creation of a Netsork Security Group Rule
#####

#Variable declaration for Module

# The NSG rule requires a RG location in which the NSG for which the rule is
created is located
variable "RGName" {
  type    = "string"
  default = "DefaultRSG"
}

#The NSG rule requires a reference to a NSG
variable "NSGReference" {
  type    = "string"
}

#The NSG Rule Name, a string value allowing to identify the rule after deployment
variable "NSGRuleName" {
  type    = "string"
  default = "DefaultNSGRule"
}

#The NSG rule priority is an integer value defining the priority in which the rule
is applied in the NSG
variable "NSGRulePriority" {
  type    = "string"
}

#The NSG rule direction define if the rule is for ingress or egress traffic. Allowed
value are inbound and outbound
variable "NSGRuleDirection" {
```

```

    type    = "string"
  }

  #The NSG Rule Access value, a string value defining if the rule allow or block the
  #specified traffic. Accepted value are Allow or Block
  variable "NSGRuleAccess" {
    type    = "string"
    default = "Allow"
  }

  #The NSG rule protocol define which type of traffic to allow/block. It accept the
  #string tcp, udp, icmp or *
  variable "NSGRuleProtocol" {
    type    = "string"
  }

  #The NSG rule source port range define the port(s) from which the traffic origing is
  #allowed/blocked
  variable "NSGRuleSourcePortRange" {
    type    = "string"
  }

  #The NSG rule destination port range define the port(s) on which the traffic
  #destination is allowed/blocked
  variable "NSGRuleDestinationPortRange" {
    type    = "string"
  }

  #The NSG rule address prefix defines the source address(es) from whichthe traffic
  #origin is allowed/blocked
  variable "NSGRuleSourceAddressPrefix" {
    type    = "string"
  }

  #The NSG rule address preifx defines the source address(es) from whichthe traffic
  #origin is allowed/blocked
  variable "NSGRuleDestinationAddressPrefix" {
    type    = "string"
  }

```

```

}
# creation of the rule

resource "azurerm_network_security_rule" "Terra-NSGRule" {

  name                = "${var.NSGRuleName}"
  priority             = "${var.NSGRulePriority}"
  direction            = "${var.NSGRuleDirection}"
  access               = "${var.NSGRuleAccess}"
  protocol             = "${var.NSGRuleProtocol}"
  source_port_range    = "${var.NSGRuleSourcePortRange}"
  destination_port_range = "${var.NSGRuleDestinationPortRange}"
  source_address_prefix = "${var.NSGRuleSourceAddressPrefix}"
  destination_address_prefix = "${var.NSGRuleDestinationAddressPrefix}"
  resource_group_name   = "${var.RGName}"
  network_security_group_name = "${var.NSGReference}"
}

# Module output

output "Name" {

  value = "${azurerm_network_security_rule.Terra-NSGRule.name}"
}

output "Id" {

  value = "${azurerm_network_security_rule.Terra-NSGRule.id}"
}

```

3.1.6 Storage account module

The storage account module is displayed below:

```

#####
#This module allows the creation of a storage account
#####

#module variables

```



```
#The ST Name

variable "StorageAccountName" {
  type    = "string"
}

#The RG Name

variable "RGName" {
  type    = "string"
}

#The Storage Account Location

variable "StorageAccountLocation" {
  type    = "string"
}

#The Storage Account Tier

variable "StorageAccountTier" {
  type    = "string"
  default = "Standard"
}

#The Storage Account Replication Type, accept LRS, GRS, RAGRS and ZRS.

variable "StorageReplicationType" {
  type    = "string"
  default = "LRS"
}

#Variables defining Tags

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

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

#Storage account creation

resource "azurerm_storage_account" "Terra-STOA" {

  name =
"stoa${lower(var.StorageAccountName)}${lower(var.StorageAccountTier)}${lower(var.StorageReplicationType)}"
  resource_group_name = "${var.RGName}"
  location             = "${var.StorageAccountLocation}"
  account_tier         = "${var.StorageAccountTier}"
  account_replication_type = "${var.StorageReplicationType}"
  account_kind         = "Storage"

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

}

#Output for the module

output "Name" {

  value = "${azurerm_storage_account.Terra-STOA.name}"
}

output "Id" {

  value = "${azurerm_storage_account.Terra-STOA.id}"
}

output "PrimaryBlobEP" {

  value = "${azurerm_storage_account.Terra-STOA.primary_blob_endpoint}"
}
```

```

}

output "PrimaryQueueEP" {

  value = "${azurerm_storage_account.Terra-STOA.primary_queue_endpoint}"
}

output "PrimaryTableEP" {

  value = "${azurerm_storage_account.Terra-STOA.primary_table_endpoint}"
}

output "PrimaryFileEP" {

  value = "${azurerm_storage_account.Terra-STOA.primary_file_endpoint}"
}

output "PrimaryAccessKey" {

  value = "${azurerm_storage_account.Terra-STOA.primary_access_key}"
}

output "SecondaryAccessKey" {

  value = "${azurerm_storage_account.Terra-STOA.secondary_access_key}"
}

output "ConnectionURI" {

  value = "${azurerm_storage_account.Terra-STOA.primary_blob_connection_string}"
}

```

3.1.7 Network Watcher module

The Network Watcher module is displayed below:

```

#####
#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"
}

# Network Watcher 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}"
}
```

3.2 Available Data sources in Terraform that can be useful

Now that we have the foundations in place, we will have a look at the available data source for Azure Resource Manager provider.

Data sources in Terraform are objects that allow to reference existing objects in the targeted environment. The format to reference a data source is:

```
data "data_source_object_in_terraform" "data_source_name" {

  param1 = "some_other_value"
  param2 = "some_value"

}
```

Terraform provides attributes that exported for each data source and those output are then used to reference the objet in the other terraform configuration. Interpolation capabilities will be used to reference the resource group under the following format:

```
"${data.azurerm_resource_group.<Resource_Group_Name>.<Resource_Group_Attribute>}"
```

We review the useful data source in the following sections.

3.2.1 Data source Resource Group

In our scenario, we aim to deploy each new set of resources to a new Resource Group. However, all the foundation resources such as the VNet and the subnets are located in the primary Resource Group. We will need to reference foundations objects for the deployment and those object are located partially with the Resource Group in which they are attached to.

The Resource Group is referenced as follow:

```
data "azurerm_resource_group" "SourceRGName" {

  name = "${var.SourceRGName}"

}
```

We can see a reference to a variable for the Resource Group name. Since the Resource Group was defined in a previous deployment, we do have the name and we can use it to reference the Resource Group in the configuration file or to get others outputs.

Following the model from the previous chapter, we will reference the Resource Group from the Infrastructure as follow:

```
"${data.azurerm_resource_group.SourceRGName.<Attribute>}"
```

In Terraform documentation, it is however specified that only the location and the associated tag are exported. We will see a little further away that it is possible to get others outputs.

3.2.2 Data source VNet

The VNet data source makes use of the Resource Group for reference:

```
data "azurerm_virtual_network" "SourceVNetName" {

  #It is possible to retake the same input value as for the "parent" template
  #name = "${var.EnvironmentTag}_VNet"
  #OR since we know the name of the deployed VNet, we could just add it in the
  variable file
  name = "${var.SourcevNetName}"

}
```

```
resource_group_name    = "${data.azurerm_resource_group.SourceRGName.name}"
}
```

As discussed earlier, we can see the reference to the Resource Group in the format planned.

3.2.3 Data source subnet

Below is the data source for the subnet:

```
data "azurerm_subnet" "FE_Subnet" {

    #Again, it is possible to quite elegantly use a vnet data source and the output
    #list of the associated subnet
    #name
    #=
    "${element(data.azurerm_virtual_network.SourcevNetName.subnets,0)}"
    #Or, again, since we do know the name of the subnet, just use the name from a
    #list in the variable file
    name                = "${element(var.SourceSubnetNameList,0)}"
    virtual_network_name = "${data.azurerm_virtual_network.SourceVNetName.name}"
    resource_group_name  = "${data.azurerm_resource_group.SourceRGName.name}"
}
```

As for the Resource Group and the VNet, we do know the names of the subnets from the first deployment. So we use the list variable to reference the subnet data source. This resource is required for the VM deployment to configure the NIC association with the appropriate subnet

3.2.4 Data source Storage Account

The data source for the storage account is below

```
data "azurerm_storage_account" "SourceSTOADiagLog" {
  name                = "stoadiaglogstandardlrs"
  resource_group_name = "${data.azurerm_resource_group.SourceRGName.name}"
}
```

We use this resource for the boot log generated from the VMs. It makes sense to centralize the logs for the VMs on a single Storage Account.

3.3 Applications of Data source in a VM Deployment template

Now that we know which Data source we may benefits from, we can go back to our scenario. We will deploy into 2 differents configurations the described VMs from the chapter 2

We use the following variables:

```
#####
# Variables for Template
#####

# Variable to define the Azure Region

variable "AzureRegion" {

    type      = "string"
    default   = "westeurope"
}

# Variable to define the Tag

variable "EnvironmentTag" {

    type      = "string"
    default   = "StagedDeploy"
}

variable "EnvironmentUsageTag" {

    type      = "string"
    default   = "PoC"
}

# Variable to define the Resource Group Name

variable "RGName" {

    type      = "string"
    default   = "RG-StagedDeploy"
}

# Variable to define the Resource Group Name

variable "RGNWName" {

    type      = "string"
    default   = "RG-NetworkWatcher"
```



```
}

#Variable defining the vnet ip range

variable "vNetIPRange" {

    type = "list"
    default = ["10.0.0.0/20"]
}

variable "SubnetAddressRange" {
#Note: Subnet must be in range included in the vNET Range

    default = {
        "0" = "10.0.0.0/24"
        "1" = "10.0.1.0/24"
        "2" = "10.0.2.0/24"

    }
}

variable "SubnetName" {

    default = {
        "0" = "FE_Subnet"
        "1" = "BE_Subnet"
        "2" = "Bastion_Subnet"

    }
}

#variable defining VM size
variable "VMSize" {

    type = "map"
    default = {
        "0" = "Standard_F1S"
        "1" = "Standard_F2s"
        "2" = "Standard_F4S"
        "3" = "Standard_F8S"

    }
}
```

```
}

# variable defining storage account tier

variable "storageaccounttier" {

    default = {
        "0" = "standard"
        "1" = "premium"
    }
}

# variable defining storage replication type

variable "storagereplicationtype" {

    default = {
        "0" = "LRS"
        "1" = "GRS"
        "2" = "RAGRS"
        "3" = "ZRS"
    }
}

# variable defining storage account tier for managed disk

variable "Manageddiskstorageetier" {

    default = {
        "0" = "standard_lrs"
        "1" = "premium_lrs"
    }
}

# variable defining VM image

# variable defining VM image
```

```
variable "PublisherName" {  
  
    default = {  
        "0" = "microsoftwindowsserver"  
        "1" = "MicrosoftVisualStudio"  
        "2" = "canonical"  
        "3" = "credativ"  
        "4" = "Openlogic"  
        "5" = "RedHat"  
    }  
}
```

```
variable "Offer" {  
  
    default = {  
        "0" = "WindowsServer"  
        "1" = "Windows"  
        "2" = "ubuntuserver"  
        "3" = "debian"  
        "4" = "CentOS"  
        "5" = "RHEL"  
    }  
}
```

```
variable "sku" {  
  
    default = {  
        "0" = "2016-Datacenter"  
        "1" = "Windows-10-N-x64"  
        "2" = "16.04.0-LTS"  
        "3" = "9"  
        "4" = "7.0"  
        "5" = "7.3"  
    }  
}
```

To check that the information we get from the Data Source match the Resources we deployed, we specify output in the first deployment:

```
#####
# This file defines which value are sent to output
#####

#####
# Resource group info Output

output "ResourceGroupName" {

    value = "${module.ResourceGroup.Name}"
}

output "ResourceGroupId" {

    value = "${module.ResourceGroup.Id}"
}

#####
# vNet info Output

output "vNetName" {

    value = "${module.SampleArchi_vNet.Name}"
}

output "vNetId" {

    value = "${module.SampleArchi_vNet.Id}"
}

output "vNetAddressSpace" {

    value = "${module.SampleArchi_vNet.AddressSpace}"
}

#####
# Log&Diag Storage account Info

output "DiagStorageAccountName" {

    value = "${module.DiagStorageAccount.Name}"
}
```

```

}

output "DiagStorageAccountID" {

    value = "${module.DiagStorageAccount.Id}"
}

/*
output "DiagStorageAccountPrimaryBlobEP" {

    value = "${module.DiagStorageAccount.PrimaryBlobEP}"
}

output "DiagStorageAccountPrimaryQueueEP" {

    value = "${module.DiagStorageAccount.PrimaryQueueEP}"
}

output "DiagStorageAccountPrimaryTableEP" {

    value = "${module.DiagStorageAccount.PrimaryTableEP}"
}

output "DiagStorageAccountPrimaryFileEP" {

    value = "${module.DiagStorageAccount.PrimaryFileEP}"
}

output "DiagStorageAccountPrimaryAccessKey" {

    value = "${module.DiagStorageAccount.PrimaryAccessKey}"
}

output "DiagStorageAccountSecondaryAccessKey" {

    value = "${module.DiagStorageAccount.SecondaryAccessKey}"
}
*/

#####
# File Storage account Info

output "FilesExchangeStorageAccountName" {

```

```

    value = "${module.FilesExchangeStorageAccount.Name}"
  }

output "FilesExchangeStorageAccountID" {

    value = "${module.FilesExchangeStorageAccount.Id}"
  }

/*
output "FilesExchangeStorageAccountPrimaryBlobEP" {

    value = "${module.FilesExchangeStorageAccount.PrimaryBlobEP}"
  }

output "FilesExchangeStorageAccountPrimaryQueueEP" {

    value = "${module.FilesExchangeStorageAccount.PrimaryQueueEP}"
  }

output "FilesExchangeStorageAccountPrimaryTableEP" {

    value = "${module.FilesExchangeStorageAccount.PrimaryTableEP}"
  }

output "FilesExchangeStorageAccountPrimaryFileEP" {

    value = "${module.FilesExchangeStorageAccount.PrimaryFileEP}"
  }

output "FilesExchangeStorageAccountPrimaryAccessKey" {

    value = "${module.FilesExchangeStorageAccount.PrimaryAccessKey}"
  }

output "FilesExchangeStorageAccountSecondaryAccessKey" {

    value = "${module.FilesExchangeStorageAccount.SecondaryAccessKey}"
  }
*/

#####
# Subnet info Output

```

```
#####

#####
#FE_Subnet

output "FE_SubnetName" {

    value = "${module.FE_Subnet.Name}"
}

output "FE_SubnetId" {

    value = "${module.FE_Subnet.Id}"
}

output "FE_SubnetAddressPrefix" {

    value = "${module.FE_Subnet.AddressPrefix}"
}

#####
#BE_Subnet

output "BE_SubnetName" {

    value = "${module.BE_Subnet.Name}"
}

output "BE_SubnetId" {

    value = "${module.BE_Subnet.Id}"
}

output "BE_SubnetAddressPrefix" {

    value = "${module.BE_Subnet.AddressPrefix}"
}

#####
#Bastion_Subnet

output "Bastion_SubnetName" {
```

```

    value = "${module.Bastion_Subnet.Name}"
  }

  output "Bastion_SubnetId" {

    value = "${module.Bastion_Subnet.Id}"
  }

  output "Bastion_SubnetAddressPrefix" {

    value = "${module.Bastion_Subnet.AddressPrefix}"
  }

#####
# Network Watcher Output Info

  output "NetworkWatcherID" {

    value = "${module.NetworkWatcher.Id}"
  }

  output "NetworkWatcherName" {

    value = "${module.NetworkWatcher.Name}"
  }

```

The result of the Terraform output is as follow:

```

BE_SubnetAddressPrefix = 10.0.1.0/24
BE_SubnetId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/StagedDeploy_VNet/subnets/
BE_Subnet
BE_SubnetName = BE_Subnet
Bastion_SubnetAddressPrefix = 10.0.2.0/24
Bastion_SubnetId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/StagedDeploy_VNet/subnets/
Bastion_Subnet
Bastion_SubnetName = Bastion_Subnet
DiagStorageAccountID = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Storage/storageAccounts/stoadiaglogstandardlrs

```



```

DiagStorageAccountName = stoadiaglogstandardlrs
FE_SubnetAddressPrefix = 10.0.0.0/24
FE_SubnetId = /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/StagedDeploy_VNet/subnets/
FE_Subnet
FE_SubnetName = FE_Subnet
FilesExchangeStorageAccountID = /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Storage/storageAccounts/stoafilesstandardlrs
FilesExchangeStorageAccountName = stoafilesstandardlrs
NetworkWatcherID = /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-
NetworkWatcher/providers/Microsoft.Network/networkWatchers/NetworkWatcher
NetworkWatcherName = NetworkWatcher
ResourceGroupId = /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy
ResourceGroupName = RG-StagedDeploy
vNetAddressSpace = [
    10.0.0.0/20
]
vNetId = /subscriptions/xxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/StagedDeploy_VNet
vNetName = StagedDeploy_VNet

```

To verify that we do use the appropriate references, we configure the variables file with references to the previous deployment:

```

# Variable to define the Resource Group Name

```

```

variable "SourceRGName" {
    type    = "string"
    default = "RG-StagedDeploy"
}

```

```

#Variable defining the vnet ip range

```

```

variable "SourcevNetName" {
    type = "string"
    default = "StagedDeploy_VNet"
}

```

```
variable "NewRGName" {

    type = "string"
    default = "RG-ExampleApps"
}

variable "SourceSubnetNameList" {

    default = ["FE_Subnet","BE_Subnet","Bastion_Subnet"]

}
```

```
# Variable to define the Resource Group Name

variable "SourceRGName" {

    type    = "string"
    default = "RG-StagedDeploy"
}

#Variable defining the vnet ip range

variable "SourcevNetName" {

    type = "string"
    default = "StagedDeploy_VNet"
}

variable "NewRGName" {

    type = "string"
    default = "RG-ExampleBastion"
}

variable "SourceSubnetNameList" {

    default = ["FE_Subnet","BE_Subnet","Bastion_Subnet"]

}
```

```
}
```

After deployment we get the following outputs, in which we can see that the resources are matching. The InfraResourceId is obtained with the Data Source Resource Group, by specifying the id in the output file, while it is not indicated as a possible attribute of the data source:

```
output "InfraResourceGroupName" {
    value = "${data.azurerm_resource_group.SourceRGName.name}"
}

output "InfraResourceId" {
    value = "${data.azurerm_resource_group.SourceRGName.id}"
}

output "InfraResourceGroupTags" {
    value = "${data.azurerm_resource_group.SourceRGName.tags}"
}

output "InfraResourceGroupLocation" {
    value = "${data.azurerm_resource_group.SourceRGName.location}"
}
```

```
Apply complete! Resources: 20 added, 0 changed, 0 destroyed.
```

```
Outputs:
```

```
InfraResourceId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy
InfraResourceGroupLocation = westeurope
InfraResourceGroupName = RG-StagedDeploy
InfraResourceGroupTags = {
    environment = StagedDeploy
    usage = PoC
}
RGVNetId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy
RGVNetName = RG-StagedDeploy
```

```
RG_ExampleBastionName = RG-ExampleBastion
RG_ExampleBastionId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleBastion
SubnetBastionId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network/virtualNetw
orks/StagedDeploy_VNet/subnets/Bastion_Subnet
VNetId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/Stag
edDeploy_VNet
VNetName = StagedDeploy_VNet
VNetSubnet1 = FE_Subnet
VNetSubnet2 = BE_Subnet
VNetSubnets = [
    FE_Subnet,
    BE_Subnet,
    Bastion_Subnet
]
```

Apply complete! Resources: 24 added, 0 changed, 0 destroyed.

Outputs:

```
InfraResourceId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy
InfraResourceGroupLocation = westeurope
InfraResourceGroupName = RG-StagedDeploy
InfraResourceGroupTags = {
    environment = StagedDeploy
    usage = PoC
}
ResourceId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps
ResourceGroupName = RG-ExampleApps
SubnetBEId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/
StagedDeploy_VNet/subnets/BE_Subnet
SubnetFEId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/
StagedDeploy_VNet/subnets/FE_Subnet
VNetDNS = []
VNetId = /subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx/resourceGroups/RG-
StagedDeploy/providers/Microsoft.Network/virtualNetworks/Stag
edDeploy_VNet
VNetName = StagedDeploy_VNet
```

```
VNetSubnet1 = FE_Subnet
VNetSubnet2 = BE_Subnet
VNetSubnets = [
    FE_Subnet,
    BE_Subnet,
    Bastion_Subnet
]
```

We can list the resources created in PowerShell:

```
PS C:\Users\User1> Get-AzureRmResource | ? {$_.ResourceGroupName -eq "RG-
StagedDeploy"} | ft name,ResourceType
```

Name	ResourceType
NSG_Bastion_Subnet	Microsoft.Network/networkSecurityGroups
NSG_BE_Subnet	Microsoft.Network/networkSecurityGroups
NSG_FE_Subnet	Microsoft.Network/networkSecurityGroups
StagedDeploy_VNet	Microsoft.Network/virtualNetworks
stoadiaglogstandardlrs	Microsoft.Storage/storageAccounts
stoafilesstandardlrs	Microsoft.Storage/storageAccounts

```
PS C:\Users\User1\Documents\> Get-AzureRmResource | ? {$_.ResourceGroupName -eq
"RG-exampleapps"} | ft name,ResourceType
```

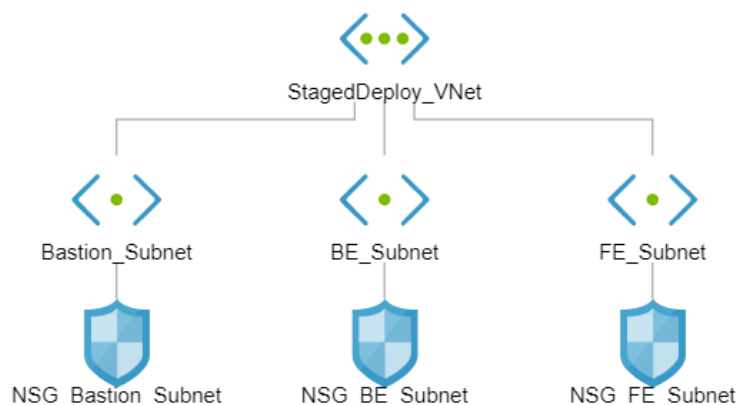
Name	ResourceType
AS_ExampleBE	Microsoft.Compute/availabilitySets
AS_ExampleFE	Microsoft.Compute/availabilitySets
DataDisk_ExampleBE1	Microsoft.Compute/disks
DataDisk_ExampleBE2	Microsoft.Compute/disks
DataDisk_ExampleFE1	Microsoft.Compute/disks
DataDisk_ExampleFE2	Microsoft.Compute/disks
ExampleBE1-OSDisk	Microsoft.Compute/disks
ExampleBE2-OSDisk	Microsoft.Compute/disks
ExampleFE1-OSDisk	Microsoft.Compute/disks
ExampleFE2-OSDisk	Microsoft.Compute/disks
ExampleBE1	Microsoft.Compute/virtualMachines
ExampleBE1/NetworkWatcherAgentForBE1	Microsoft.Compute/virtualMachines/extensions
ExampleBE2	Microsoft.Compute/virtualMachines
ExampleBE2/NetworkWatcherAgentForBE2	Microsoft.Compute/virtualMachines/extensions

```
ExampleFE1           Microsoft.Compute/virtualMachines
ExampleFE1/NetworkWatcherAgentForFE1 Microsoft.Compute/virtualMachines/extensions
ExampleFE2           Microsoft.Compute/virtualMachines
ExampleFE2/NetworkWatcherAgentForFE2 Microsoft.Compute/virtualMachines/extensions
NIC_ExampleBE1       Microsoft.Network/networkInterfaces
NIC_ExampleBE2       Microsoft.Network/networkInterfaces
NIC_ExampleFE1       Microsoft.Network/networkInterfaces
NIC_ExampleFE2       Microsoft.Network/networkInterfaces
fepip1               Microsoft.Network/publicIPAddresses
fepip2               Microsoft.Network/publicIPAddresses
stoaappsstandardlrs Microsoft.Storage/storageAccounts
```

```
PS C:\Users\User1\Documents> Get-AzureRmResource | ? {$_.ResourceGroupName -eq "RG-
exampleapps"
} | ft name,ResourceType
```

Name	ResourceType
AS_ExampleBE	Microsoft.Compute/availabilitySets
AS_ExampleFE	Microsoft.Compute/availabilitySets
DataDisk_ExampleBE1	Microsoft.Compute/disks
DataDisk_ExampleBE2	Microsoft.Compute/disks
DataDisk_ExampleFE1	Microsoft.Compute/disks
DataDisk_ExampleFE2	Microsoft.Compute/disks
ExampleBE1-OSDisk	Microsoft.Compute/disks
ExampleBE2-OSDisk	Microsoft.Compute/disks
ExampleFE1-OSDisk	Microsoft.Compute/disks
ExampleFE2-OSDisk	Microsoft.Compute/disks
ExampleBE1	Microsoft.Compute/virtualMachines
ExampleBE1/NetworkWatcherAgentForBE1	Microsoft.Compute/virtualMachines/extensions
ExampleBE2	Microsoft.Compute/virtualMachines
ExampleBE2/NetworkWatcherAgentForBE2	Microsoft.Compute/virtualMachines/extensions
ExampleFE1	Microsoft.Compute/virtualMachines
ExampleFE1/NetworkWatcherAgentForFE1	Microsoft.Compute/virtualMachines/extensions
ExampleFE2	Microsoft.Compute/virtualMachines
ExampleFE2/NetworkWatcherAgentForFE2	Microsoft.Compute/virtualMachines/extensions
NIC_ExampleBE1	Microsoft.Network/networkInterfaces
NIC_ExampleBE2	Microsoft.Network/networkInterfaces
NIC_ExampleFE1	Microsoft.Network/networkInterfaces
NIC_ExampleFE2	Microsoft.Network/networkInterfaces
fepip1	Microsoft.Network/publicIPAddresses
fepip2	Microsoft.Network/publicIPAddresses
stoaappsstandardlrs	Microsoft.Storage/storageAccounts

As a last part, since we deployed a Network watcher instance, let's have a view of the topology from the portal:



As we can see, while we did proceed to the deployment of 2 additional environments based on VMs, we can only display the Network resources associated to the resource group containing the VNet. To get a view of the topology of the other deployment, we have to use PowerShell:

```
PS C:\Users\User1> Get-AzureRmNetworkWatcherTopology -TargetResourceGroupName rg-
stageddeploy -ResourceGroupName RG-NetworkWatcher -NetworkWatcherName
networkwatcher
```

```

Id                : 032d26e3-c9c0-4039-b827-9398b9f541b9
CreatedDateTime   : 05/03/2018 13:46:21
LastModified      : 05/03/2018 13:33:11
Resources         : [
    {
      "Name": "NSG_Bastion_Subnet",
      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/networkSecurityGroups/NSG_Bastion_Subnet",
      "Location": "westeurope",
      "Associations": [
        {
          "AssociationType": "Associated",
          "Name": "Bastion_Subnet",
  
```

```

        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/Bastion_Subnet"
      }
    ]
  },
  {
    "Name": "NSG_BE_Subnet",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/networkSecurityGroups/NSG_BE_Subnet",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "BE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
      }
    ]
  },
  {
    "Name": "NSG_FE_Subnet",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/networkSecurityGroups/NSG_FE_Subnet",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "FE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
      }
    ]
  },
  {
    "Name": "StagedDeploy_VNet",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/virtualNetworks/StagedDeploy_VNet",

```



```

    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "FE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
      },
      {
        "AssociationType": "Contains",
        "Name": "BE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
      },
      {
        "AssociationType": "Contains",
        "Name": "Bastion_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/Bastion_Subnet"
      }
    ]
  },
  {
    "Name": "FE_Subnet",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "NSG_FE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/networkSecurityGroups/NSG_FE_Subnet"
      }
    ]
  },
  {
    "Name": "BE_Subnet",

```

```

        "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet",
        "Location": "westeurope",
        "Associations": [
            {
                "AssociationType": "Associated",
                "Name": "NSG_BE_Subnet",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/networkSecurityGroups/NSG_BE_Subnet"
            }
        ]
    },
    {
        "Name": "Bastion_Subnet",
        "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Microsoft.Network
/virtualNetworks/StagedDeploy_VNet/subnets/Bastion_Subnet",
        "Location": "westeurope",
        "Associations": [
            {
                "AssociationType": "Associated",
                "Name": "NSG_Bastion_Subnet",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/networkSecurityGroups/NSG_Bastion_Subnet"
            }
        ]
    }
]

```

```

PS C:\Users\User1> Get-AzureRmNetworkWatcherTopology -TargetResourceGroupName rg-
exampleapps -ResourceGroupName RG-NetworkWatcher -NetworkWatcherName networkwatcher

```

```

Id                : 80fb5cc3-3a5e-466a-9ab1-bbe214ec7b31
CreatedDateTime    : 05/03/2018 13:46:30
LastModified       : 05/03/2018 13:36:06
Resources          : [
                        {
                            "Name": "ExampleBE1",

```

```

    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
    virtualMachines/ExampleBE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleBE1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleBE1"
      }
    ]
  },
  {
    "Name": "ExampleBE2",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
    virtualMachines/ExampleBE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleBE2",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleBE2"
      }
    ]
  },
  {
    "Name": "ExampleFE1",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
    virtualMachines/ExampleFE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleFE1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE1"
      }
    ]
  }
}

```

```

    ]
  },
  {
    "Name": "ExampleFE2",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
virtualMachines/ExampleFE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleFE2",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE2"
      }
    ]
  },
  {
    "Name": "NIC_ExampleBE1",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleBE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "ExampleBE1",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleBE1"
      },
      {
        "AssociationType": "Associated",
        "Name": "BE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
      }
    ]
  },
  {
    "Name": "NIC_ExampleBE2",

```

```

        "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
        networkInterfaces/NIC_ExampleBE2",
        "Location": "westeurope",
        "Associations": [
            {
                "AssociationType": "Associated",
                "Name": "ExampleBE2",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleBE2"
            },
            {
                "AssociationType": "Associated",
                "Name": "BE_Subnet",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
            }
        ]
    },
    {
        "Name": "NIC_ExampleFE1",
        "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
        networkInterfaces/NIC_ExampleFE1",
        "Location": "westeurope",
        "Associations": [
            {
                "AssociationType": "Associated",
                "Name": "ExampleFE1",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleFE1"
            },
            {
                "AssociationType": "Associated",
                "Name": "FE_Subnet",
                "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
            },
            {
                "AssociationType": "Associated",

```

```

        "Name": "fepip1",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/publicIPAddresses/fepip1"
      }
    ]
  },
  {
    "Name": "NIC_ExampleFE2",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleFE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "ExampleFE2",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleFE2"
      },
      {
        "AssociationType": "Associated",
        "Name": "FE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
      },
      {
        "AssociationType": "Associated",
        "Name": "fepip2",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/publicIPAddresses/fepip2"
      }
    ]
  },
  {
    "Name": "fepip1",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
publicIPAddresses/fepip1",
    "Location": "westeurope",
    "Associations": [

```

```

        {
          "AssociationType": "Associated",
          "Name": "NIC_ExampleFE1",
          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE1"
        }
      ]
    },
    {
      "Name": "fepip2",
      "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
publicIPAddresses/fepip2",
      "Location": "westeurope",
      "Associations": [
        {
          "AssociationType": "Associated",
          "Name": "NIC_ExampleFE2",
          "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE2"
        }
      ]
    }
  ]
}
]

```

```

PS C:\Users\User1> Get-AzureRmNetworkWatcherTopology -TargetResourceGroupName rg-
exampleapps -ResourceGroupName RG-NetworkWatcher -NetworkWatcherName networkwatcher

```

```

Id                : 80fb5cc3-3a5e-466a-9ab1-bbe214ec7b31
CreatedDateTime    : 05/03/2018 13:46:30
LastModified       : 05/03/2018 13:36:06
Resources          : [
  {
    "Name": "ExampleBE1",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
virtualMachines/ExampleBE1",
    "Location": "westeurope",
    "Associations": [
      {

```

```

        "AssociationType": "Contains",
        "Name": "NIC_ExampleBE1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleBE1"
      }
    ]
  },
  {
    "Name": "ExampleBE2",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
virtualMachines/ExampleBE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleBE2",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleBE2"
      }
    ]
  },
  {
    "Name": "ExampleFE1",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/
virtualMachines/ExampleFE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleFE1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE1"
      }
    ]
  },
  {
    "Name": "ExampleFE2",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Compute/

```



```

    virtualMachines/ExampleFE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Contains",
        "Name": "NIC_ExampleFE2",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE2"
      }
    ]
  },
  {
    "Name": "NIC_ExampleBE1",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleBE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "ExampleBE1",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleBE1"
      },
      {
        "AssociationType": "Associated",
        "Name": "BE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
      }
    ]
  },
  {
    "Name": "NIC_ExampleBE2",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleBE2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",

```

```

        "Name": "ExampleBE2",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleBE2"
      },
      {
        "AssociationType": "Associated",
        "Name": "BE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/BE_Subnet"
      }
    ]
  },
  {
    "Name": "NIC_ExampleFE1",
    "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleFE1",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "ExampleFE1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleFE1"
      },
      {
        "AssociationType": "Associated",
        "Name": "FE_Subnet",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
      },
      {
        "AssociationType": "Associated",
        "Name": "fepip1",
        "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/publicIPAddresses/fepip1"
      }
    ]
  },
},

```

```

    {
      "Name": "NIC_ExampleFE2",
      "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
networkInterfaces/NIC_ExampleFE2",
      "Location": "westeurope",
      "Associations": [
        {
          "AssociationType": "Associated",
          "Name": "ExampleFE2",
          "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Compute/virtualMachines/ExampleFE2"
        },
        {
          "AssociationType": "Associated",
          "Name": "FE_Subnet",
          "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-StagedDeploy/providers/Micro
soft.Network/virtualNetworks/StagedDeploy_VNet/subnets/FE_Subnet"
        },
        {
          "AssociationType": "Associated",
          "Name": "fepip2",
          "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/publicIPAddresses/fepip2"
        }
      ]
    },
    {
      "Name": "fepip1",
      "Id": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
publicIPAddresses/fepip1",
      "Location": "westeurope",
      "Associations": [
        {
          "AssociationType": "Associated",
          "Name": "NIC_ExampleFE1",
          "ResourceId": "/subscriptions/xxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE1"
        }
      ]
    }
  ]
}

```

```

    ]
  },
  {
    "Name": "fepip2",
    "Id": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Microsoft.Network/
publicIPAddresses/fepip2",
    "Location": "westeurope",
    "Associations": [
      {
        "AssociationType": "Associated",
        "Name": "NIC_ExampleFE2",
        "ResourceId": "/subscriptions/xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxxx/resourceGroups/RG-ExampleApps/providers/Micros
oft.Network/networkInterfaces/NIC_ExampleFE2"
      }
    ]
  }
]

```

4 Conclusion

In this article, we just scratched the surface of Terraform Data source. It gave us some hints on how to operate in a Cloud environment with Terraform. We will look in other articles how to include Terraform in a DevOpsWorld with stuff such as Workspace, Remote state and the custom data source. The code for the template is available on github [here](#).

