



# Design Azure Networking for Advanced Security

**Hands-on lab**

Azure offers a number of mechanisms for mitigating risks to virtual machines, networks, and other resources. Site-to-Site and VNET-to-VNET virtual private networks allow data to be transmitted securely, between an on-premise site and a virtual network or between virtual networks. Network Security Groups (NSG) allow designers to configure network access controls at the subnet, VM, or NIC level to provide an additional layer of defense. VPNs and NSGs can complement each other, for example, to provide a means of allowing management access over a secure VPN to reduce the number of ports that have to expose on the Internet.

In this lab, you will learn about the VPN and NSG resources that are available in Azure. As well, you learn how to configure and deploy these resources by using ARM templates.

Produced by HynesITe, Inc  
Version 1.0  
10/2/2015



This document supports a preliminary release of a software product that may be changed substantially prior to final commercial release. This document is provided for informational purposes only and Microsoft makes no warranties, either express or implied, in this document. Information in this document, including URL and other Internet Web site references, is subject to change without notice. The entire risk of the use or the results from the use of this document remains with the user. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Copyright 2014 © Microsoft Corporation. All rights reserved.

Microsoft Active Directory, Azure Active Directory, Azure, Hyper-V, Windows, and Windows Server 2012 are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

# Design Azure Networking for Advanced Security

## Contents

Design Azure Networking for Advanced Security .....	3
Design Azure Networking for Advanced Security .....	5
Before You Begin .....	6
Azure Subscriptions .....	6
Create a Free Trial Account .....	6
Configure an Azure Pass .....	7
Hosted Workstation .....	7
Use of Own System .....	8
GitHub repository for Lab Files .....	8
Required Software .....	8
Optional Software .....	8
Access the Lab Environment .....	10
Introduction and Scenario .....	11
Prepare the Azure Infrastructure .....	12
Configure a public IP address on the Edge server .....	12
Run the Lab03Start.ps1 script .....	12
Deploy Lab 03 infrastructure from GitHub (alternate setup instructions) .....	14
Analyze ARM Template Used To Deploy Network Related Resources .....	16
Open azuredeploy.json template file in Visual Studio Code .....	16
Examine Azure Networking resources deployment in ARM template .....	17
Virtual Network resource .....	17
Public IP addresses resource .....	18
Network Gateway resources .....	20
Complete Site-To-Site VPN Configuration .....	23
Configure local VPN device for site-to-site VPN connection .....	23
Configure Network Security Groups by using ARM Templates .....	29
Example: Defense in depth using NSGs .....	30
More Information .....	32
Verify connectivity to virtual machines .....	32

Deploy Network Security Groups by using an ARM Template.....	35
Lab Scenario Goals .....	35
Troubleshooting template deployment (if necessary) .....	37
Verify NSG rules .....	38
Remove resource group used for lab.....	42
Remove Azure resource group.....	42

## Design Azure Networking for Advanced Security

Azure offers a number of mechanisms for mitigating risks to virtual machines, networks, and other resources. Site-to-Site and VNET-to-VNET virtual private networks allow data to be transmitted securely, between an on-premise site and a virtual network or between virtual networks. Network Security Groups (NSG) allow designers to configure network access controls at the subnet, VM, or NIC level to provide an additional layer of defense. VPNs and NSGs can complement each other, for example, to provide a means of allowing management access over a secure VPN to reduce the number of ports that have to be exposed on the Internet.

In this lab, you will learn about the VPN and NSG resources that are available in Azure. As well, you learn how to configure and deploy these resources by using ARM templates.

## Before You Begin

In this lab, you will examine and analyze a number of quick start ARM templates that are available on GitHub. You will create a GitHub account, if you don't already have one, to host a GitHub repo for a quick start template that you will download and then modify and deploy. When you have completed the lab exercises, you will clean up the Azure resources that you created in the lab.

To as great extent as possible, the lab instructions assume the use the Azure Preview Portal, which is located at <https://portal.azure.com>. Some tasks are only available through the Azure Portal. There will be some need to switch back and forth between the two portals. Most of what you will be doing could also be done using the full portal. However, for the sake of consistency and clarity, lab instructions have only been written using the Preview Portal whenever possible. The full portal is located at <https://manage.windowsazure.com>.

For more information on the preview portal, please see <http://channel9.msdn.com/Blogs/Windows-Azure/Azure-Preview-portal> for a brief demonstration or <http://azure.microsoft.com/en-us/documentation/preview-portal/> to read the current documentation for the preview portal.

## Azure Subscriptions

This IT Camp lab requires a valid Azure subscription. While you may use an existing subscription such as a subscription associated MSDN account or existing corporate account, it is strongly recommended to use an Azure Free Trial account or an Azure Pass. By using a Free Trial or an Azure Pass, you will avoid any charges against your MSDN or corporate subscription that would result from doing the exercises in this camp.

Your instructor may be able to provide you with a code that will allow you to redeem an Azure Pass. Or, you may use a CLEAN and UNUSED Azure Trial account - details on how to set up both a Free Trial account and configure an Azure Pass subscriptions are provided below.

### Create a Free Trial Account

To create a new Azure trial account perform the following steps.

1. Navigate to [www.live.com](http://www.live.com) and click **Sign up now**.
2. Follow the on-screen instructions to create a new Microsoft Account.
3. Navigate to [www.azure.com](http://www.azure.com) and click **Free Trial**.
4. Follow the on-screen instructions to activate a new Windows Azure Trial.
5. Navigate to [Manage.windowsazure.com](http://Manage.windowsazure.com) and sign in.
6. In Microsoft Azure portal, in the upper left, click your user name, and then click **View my bill**.
7. Click your current trial subscription, and then click **Edit subscription details**.
8. Type a name you will recognize in SUBSCRIPTION NAME, such as ITCamps, and then click the **Done** icon.

### Configure an Azure Pass

Your instructor may be able to provide you with a pre-provisioned Microsoft Account that already has an Azure Pass subscription associated with it. Alternatively, your instructor may be able to provide you with an Azure promotional code.

To activate the promotional code and create a new Azure Pass account perform the following steps.

1. If you are not using the lab virtual machine to activate your Azure Pass promotional code, ensure you open an InPrivate browser session before performing these steps.
  - ❖ It is critically important that you do not accidentally associate the promotional code with any account that has previously been associated with or linked to an Azure subscription. Use an InPrivate browser session to ensure that no credentials are unintentionally forwarded during the process to activate and redeem the promotional code. If you fail to activate the code because you logged in with the wrong account, you will render the code useless and will not be able to use it again.
2. Navigate to [www.live.com](http://www.live.com) and click **Sign up now**.
3. Follow the on-screen instructions to create a new Microsoft Account.
  - ★ Please ensure, you create an outlook.com, live.com or Hotmail.com account. Do not use accounts that have country code suffixes, such as .dk, ca, uk, etc. in their names.
4. Navigate to <http://www.microsoftazurepass.com> and follow the onscreen instructions to redeem the promotional code.
  - ★ Once you have submitted the promotional code, it will take a few minutes for the account to become activated. Only one promo code can be redeemed per the life of the Microsoft ID.
5. Follow the on-screen instructions to activate a new Windows Azure Trial.
6. Navigate to [Manage.windowsazure.com](http://Manage.windowsazure.com) and sign in.
7. In Microsoft Azure portal, in the upper left, click your user name, and then click **View my bill**.
8. Click your current trial subscription, and then click **Edit subscription details**.
9. Type a name you will recognize in SUBSCRIPTION NAME, such as ITCamps, and then click the **Done** icon.

### Hosted Workstation

Labs in this camp are written to be completed on a pre-configured workstation. Additional labs require an on-premises environment consisting of multiple servers. A hosted virtual machine environment is provided for this purpose. Your instructor will provide a link to this environment.

If you are using the hosted workstation environment, use Administrator as the username and Passw0rd! as the password.

## Use of Own System

You may complete lab instructions using your own workstation (either Windows 10 or Windows 8.1), providing you download the files used for the lab from GitHub and have the following software installed.

- ⚠ Please note that to perform all the lab steps, you will need a public IP address and VPN device to configure a site-to-site VPN. These are provided for you in the hosted lab environment. If you are using your own lab environment, for example, using virtual machines, please keep in mind that your VPN device (e.g, Windows Server 2012 R2 with RRAS installed), must not be behind a NAT device.

You can do most, but not all of the lab steps, if you cannot meet the requirement for a public IP address and VPN device that is not behind a NAT device.

## GitHub repository for Lab Files

If you are not using the hosted virtual machine and are using your own workstation, any custom files the lab instruction call out can be found in a GitHub repository. The repository is located here:

<https://github.com/AZITCAMP/Labfiles>.

## Required Software

1. Microsoft Azure PowerShell - <http://go.microsoft.com/?linkid=9811175&clcid=0x409> (also installs the Web Platform Installer)
2. Visual Studio Code - <https://code.visualstudio.com/>
3. GitHub Desktop for Windows - <https://desktop.github.com/>
4. Windows Credential Store for Git (if VSCode won't authenticate with GitHub) - <http://gitcredentialstore.codeplex.com/>
5. Iometer - <http://sourceforge.net/projects/iometer/>

## Optional Software

Any additional software that you require will be called out in the lab. The following software may be useful when working with Azure in general.

1. Remote Server Administration Tools - <http://support.microsoft.com/kb/2693643> (Windows 8.1) or <http://www.microsoft.com/en-ca/download/details.aspx?id=45520> (Windows 10)
2. AzCopy - <http://aka.ms/downloadazcopy>
3. Azure Storage Explorer - <http://azurestorageexplorer.codeplex.com/downloads/get/891668>
4. Microsoft Azure Cross-platform Command Line Tools (installed using the Web Platform Installer)
5. Visual Studio Community 2015 with Microsoft Azure SDK - 2.7.1 (installed using the Web Platform Installer)
6. Msysgit - <http://msysgit.github.io>
7. PuTTY and PuTTYgen - [www.putty.org](http://www.putty.org)



8. Microsoft Online Services Sign-In Assistant for IT Professionals RTW -  
<http://go.microsoft.com/fwlink/?LinkID=286152>
9. Azure Active Directory Module for Windows PowerShell (64-bit version) -  
<http://go.microsoft.com/fwlink/p/?linkid=236297>

Please note that these lab exercises to require a minimum version of 0.9.8 of the Microsoft Azure module for PowerShell. To determine the module version installed on your system, open a Windows PowerShell prompt, type the following commands, and then press ENTER.

```
↪ import-module Azure  
↪ get-module Azure).version
```

```
PS C:\> import-module azure  
PS C:\> (get-module Azure).Version  
Major Minor Build Revision  
----
```

Major	Minor	Build	Revision
0	9	8	-1

## Access the Lab Environment

For this lab you will be accessing a hosted environment that contains all the VMs and resources you require. The hosted environment is located here: <https://labondemand.com/Launch/E461682F>.

You should be able to connect with any recent web browser, including Microsoft Edge. Once you have connected to the lab environment, take a few minutes to familiarize yourself with Launchpad.

For this course there are four VMs that you will work in. If you look at the Machines tab on the right side of the lab environment you will find a listing of all the VMs. To switch to another VM, just click on the appropriate name in the Machines list. Below you will find a listing of the VMs for this course.

Virtual Machine	Role
AZRCamp-Admin	Windows 10, A member of the Contoso.com domain. Used for Azure management.
AZRCamp-Edge	A Stand-alone Windows Server 2012 R2 Server. Routing and Remote Access has been installed and it is acting as the default gateway for all outbound traffic
AZRCamp-DC	Windows Server 2012 R2 domain controller and DNS server.
AZRCamp-Sync	Directory Sync for use in other Labs.

The password for all logons in these VMs is "Passw0rd!".

- ★ You can type this in to the VM manually, or use the **Commands→Paste→Paste Password** sequence from the Launchpad.

## Introduction and Scenario

Contoso, Inc. has asked you to continue to investigate the advantages of deploying IaaS using Azure Resource Manager. A significant is the security of applications that reside in Azure. Contoso would like to see its best practices reflected in the configuration of the Azure networking to help ensure the integrity and confidentiality of its data. In particular, Contoso is interested in learning how Azure Resource Manager templates may be used to quickly deploy an infrastructure that uses VPN to provide connectivity to manage resources and at the same time uses Network Security Groups to control access.

In this lab, you will analyze the JSON objects that make it possible to deploy network configurations. You will then complete a partially finished template to deploy Network Security Groups.

## Prepare the Azure Infrastructure

In this exercise, you will use the Lab03Start.ps1 script to log on to your Azure subscription and deploy the Azure infrastructure required for these set of lab exercises. The script creates a new resource group called RG-AZITCAMP-LAB03. The script then uses an Azure Resource Manager (ARM) template to deploy 3 subnets and two virtual machines to the resource group.

The script requires your initials and a password as inputs. The unique storage account and public DNS parameters are generated by the script and passed to the ARM template at run time, along with the password. The script takes about 10 - 20 minutes to complete the deployment. When the script completes, it displays the two dynamic public IP addresses assigned to the two virtual machines.

### Configure a public IP address on the Edge server

To configure a site-to-site VPN, you require a public IP address that is connected to your VPN device. It is important to note that, while your VPN device may be behind a firewall device or load balancer of some kind, it cannot be behind a NAT (Network Address Translation) device.

In our lab environment, the RRAS server (AZRCAMP-EDGE) is behind a NAT device to preserve public IP addresses. In this task you will change the Launchpad configuration to directly connect the external interface of AZRCAMP-EDGE to the Internet and the change the configuration of the external adapter get a public IP address via DHCP.

- ✎ Perform the following tasks on AZRCAMP-EDGE logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

- ✦ **Note:** If the virtual machine is not already logged on, on the menu bar, click **Commands**, and then click **Ctrl+Alt+Delete** to display the log on screen.

10. Switch to the **AZRCamp-Edge** in Launchpad.
11. In Launchpad, on the **Machines** tab, change the Internet connection to **Internet-Direct**.
12. On the desktop, double-click **get-public-ip.cmd**.
  - ✦ The script uses the netsh command to change the address type to DHCP. When a public IP address is acquired for the external network interface, it is displayed in the command prompt output. You will need to know this IP address for subsequent steps.
13. Record the public IP address displayed in the command prompt.
  - ✦ You will need to know this address in later steps.

### Run the Lab03Start.ps1 script

In this task, you will run the Lab03Start.ps1 script to deploy the Azure VM and network infrastructure for Lab 03. As an alternative to running this script, you can also deploy the infrastructure from a GitHub repository. Please see the alternate instructions below, if you wish to use this alternate method.

✎ Perform the following tasks on **AZRCAMP-ADMIN**:

1. Open File Explorer and navigate to **C:\LabFiles\AZRITPROCamp\Lab03 – Design Azure Networking for Advanced Security**.

✎ You may also download files used for this lab from the GitHub repository for the course at <https://github.com/AZITCAMP/Labfiles>.

2. Right-click **Lab03Start.ps1**, and click **Edit**.

✎ The Windows PowerShell ISE console opens.

3. In Windows PowerShell ISE, on the upper Ribbon, click **Run Script** (green arrow).

4. When prompted, enter a lower-case string that represents your initials, and press ENTER.

✎ Your initials are used to create unique names for the storage account and public DNS names.

5. In the Sign in to Windows Azure PowerShell dialog box, enter the email address of the account associated with your Azure subscription, and click **Continue**.

6. On the sign in page, enter your password, and click **Sign in**.

✎ The script starts running and then pauses to display the storage account and public DNS names that will be used for the lab.

7. Press ENTER to continue.

8. When prompted for the Admin Password parameter, type **Passw0rd!** and press ENTER.

9. In the command pane, at the prompt for the localGatewayIpAddress parameter, type the IP address of the external adapter on the AZRCAMP-EDGE server that you determined earlier, and press ENTER.

✎ The deployment starts.

✎ After 10 - 20 minutes or more minutes, the infrastructure is deployed. The VPN gateway will take the longest to deploy, as long as 30 minutes. Please be patient. If you log on to <https://portal.azure.com> you can watch the progress of the deployment.

✎ When the script completes, the public IP addresses and DSNs name are displayed as the output.

✎ Please continue with the lab steps. You do not have to wait for the deployment to complete in its entirety to do subsequent steps.

10. Record the public IP addresses and the DNS name.

11. Leave the Windows PowerShell ISE console open for subsequent lab exercises.

## Deploy Lab 03 infrastructure from GitHub (alternate setup instructions)

As an alternative to deploying the Azure infrastructure using a PowerShell script, you can set up the infrastructure using a template deployment directly from GitHub.

❖ **Perform these instructions only if you have NOT performed the instructions above. These instructions are provided as an alternate method for setting up the Azure lab infrastructure.**

✎ Perform the following tasks on **AZRCAMP-ADMIN**:

1. Open Microsoft Edge, and browse to

<https://github.com/AZITCAMP/Labfiles/tree/master/lab03> .

★ The lab03 repository contains all files you need to complete the lab.

2. In the Readme.md section, click **Deploy to Azure**.

3. If prompted, log on to the Azure portal.

★ The Parameters tab for the virtual machine custom deployment appears.

4. In the PUBLICDNSNAME, type a unique name, such as your initials plus a random number between 10,000 and 99,000 (e.g, abc12345).

5. In NEWSTORAGEACCOUNTNAME, type a unique name such as [abc]store#, where [abc] represents your initials and # is a random 3 or 4 digit number.

★ This name must be unique, must all be lower case, and contain only letters and numbers.

6. In ADMINPASSWORD, type **Passw0rd!**

7. In LOCALGATEWAYIPADDRESS, enter the IP address of the external interface of AZRCAMP-EDGE that you determined earlier.

8. Leave the remaining parameters at their default value, and click **OK**.

9. On the custom deployment blade, in the Resource group section, **Or create new**.

10. In the text box, type RG-AZITCAMP-LAB03.

★ You can name the resource group anything you want. However, keep in mind that the RGCleanup.ps1 script relies on the existence of a resource group(s) named \*AZITCAMP\*.

11. Click **Legal terms**.

12. In the Buy blade, click **Buy**.

13. On the Custom deployment tab, click **Create**.

14. Leave the Microsoft Edge browser open for subsequent steps.

## Analyze ARM Template Used To Deploy Network Related Resources

The Azure Resource Manager (ARM) template you use to deploy the infrastructure creates a number network-related resources, including a virtual network, subnets, and virtual network gateway. The virtual network and the subnets can be created either by using the Azure preview portal or by using an ARM template that relies on the Network REST API (<https://msdn.microsoft.com/en-us/library/azure/mt163658.aspx>).

Currently, at the time of this writing, it is not possible to use the Azure preview portal to create virtual network gateways and virtual network gateway connections using the Azure preview portal. This means that if you wish your deployment to include a site-to-site or a point-to-site VPN and you wish to use Resource Manager deployment model to create a virtual network gateway and virtual network connection for the VPN, you must use either Azure PowerShell cmdlets or ARM templates. In either case, the PowerShell cmdlet or the ARM template that you use to create your virtual network gateway and connection will rely on the Azure Network Gateway REST API (<https://msdn.microsoft.com/en-us/library/azure/mt163859.aspx>).

In this lab exercise, you will examine how the network-related resources are deployed by using the ARM template used to create the infrastructure for Lab 03.

### Open azuredeploy.json template file in Visual Studio Code

In this task, you will open a copy of the azuredeploy.json template file in Visual Studio Code.

❖ These instructions assume that you are using the provided lab environment. If you do not have the lab files stored locally, you can acquire them at <https://github.com/AZITCAMP/Labfiles/tree/master/lab03>.

✎ Perform the following tasks on AZRCAMP-ADMIN and AZRCAMP-EDGE logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

✦ Note: If the virtual machine is not already logged on, on the menu bar, click Commands, and then click Ctrl+Alt+Delete to display the log on screen.

1. On AZRCAMP-ADMIN, open File Explorer, and navigate to **C:\LabFiles\AZITPROCamp\Lab03**.
2. Double-click **azuredeploy.json**.  
✦ The azuredeploy.json file opens in Visual Studio Code.
3. Leave Visual Studio Code open for subsequent lab steps.



## Examine Azure Networking resources deployment in ARM template

Aside from the need for a local gateway, which could be another Azure network that has its own virtual network gateway, to create a virtual network gateway, you need to have a virtual network that contains at least two subnets. One of the subnets must be named GatewaySubnet and is used as a routing domain between the remote site (or point) and your IaaS deployment. As such, the GatewaySubnet requires that it be configured with at least a /29 CIDR to provide a minimum number of IP addresses to create the routes.

### Virtual Network resource

To meet the pre-requisite of the Gateway subnet, it is necessary for you to use the Microsoft.Network/virtualNetworks resource in your template.

The following shows a simplified version of the Microsoft.Network/virtualNetworks resource:

- ✦ For the sake of clarity, some of the elements, such as tags, dhcpOptions and dnsServers, have been omitted. For a complete list of the elements used in the Microsoft.Network/virtualNetworks resource, please see: <https://msdn.microsoft.com/en-us/library/azure/mt163661.aspx>.

To create a subnet or subnets, we need to create a virtual network to contain the subnets. The virtual network resource requires, at a minimum, a name and location in order to create it. You may configure the resources so that it depends on other resources being present as a pre-requisite. The following is a brief description of the other resource elements.

- **addressSpace:** The addressSpace element contains an array of IP address ranges denoted by list of IP of addressPrefixes.
- **addressPrefixes:** that denote the range of IP addresses in CIDR notation that can be consumed by subnets in your deployment.
- **subnets:** the subnets element contains an array of subnets. As with virtual networks, subnets have a name and an addressPrefix. If you wish to associate a Network Security Group (NSG), you do it in this section (please see later lab exercises for an explanation of this).

In the azuredeploy.json template that you opened earlier, the Microsoft.Network/virtualNetworks resource starts at about line 117:

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/virtualNetworks",
  "name": "name-of-VNET",
  "location": "location",
  "dependsOn": [
    ],
  "properties": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "subnets": [
      {
        "name": "subnet1",
        "properties": {
          "addressPrefix": "10.0.0.0/24",
        }
      }
    ]
  }
}
```

The relevant variables are declared earlier in the template, as follows:

```
"VNETName": "Lab03_VNET",
"FESubnetPrefix": "10.0.0.0/24",
"FESubnetName": "FESubnet",
"BESubnetPrefix": "10.0.1.0/24",
"BESubnetName": "BESubnet",
"GWSubnetPrefix": "10.0.200.0/28",
"GWSubnetName": "GatewaySubnet",
"vnetAddressRange": "10.0.0.0/16"
```

Given the values provided for the resource, the template causes a virtual network named Lab03\_VNET to be created using a CIDR block of 10.0.0.0/16. Within the VNET, 3 subnets are created, FESubnet, BESubnet, and GatewaySubnet. Each of these subnets is assigned a CIDR block of 10.0.0.0/24, 10.0.1.0/24, and 10.0.200.0/28, respectively. Note that the location value is provided by a function: [resourceGroup().location].

### Public IP addresses resource

A virtual network gateway requires a public IP address that is assigned to it. You assign public IP addresses to resources such as virtual network gateways and virtual machines by using the Microsoft.Network/publicIPAddresses resource, which may be found here:

<https://msdn.microsoft.com/en-us/library/azure/mt163590.aspx>.

In the deployment, we require 3 public IP address: one for each of the two servers and another for the virtual network connection endpoint. In the template, these resources are created immediately after the variables section starting at about line 70.

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "[concat(variables('publicIPAddressName'),'0')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "publicIPAllocationMethod": "Dynamic",
    "dnsSettings": {
      "domainNameLabel": "[parameters('publicDnsName')]"
    }
  }
},
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "[concat(variables('publicIPAddressName'),'1')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "publicIPAllocationMethod": "Dynamic"
  }
},
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/publicIPAddresses",
  "name": "[variables('gatewayPublicIPName')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "publicIPAllocationMethod": "Dynamic"
  }
},
}
```

In the Microsoft.Network/publicIPAddresses, there are a number of settings that are required, such as the name and the publicIPAllocationMethod. For the publicIPAllocation method, only two values are possible: Static or Dynamic.

The relevant variables and parameters are defined earlier in the json file. The public DNS name must be unique and is defined as a parameter to allow the template user the ability to enter a unique name upon deployment.

```
"publicDnsName": {
  "type": "string",
  "metadata": {
    "description": "Unique public DNS prefix for the deployment.
The fqdn will look something like
'<dnsname>.westus.cloudapp.azure.com'. Up to 62 chars, digits or
dashes, lowercase, should start with a letter: must conform to '^[a-
z][a-z0-9-]{1,61}[a-z0-9]$'."
  }
},
```

The public IP address name is created by using two variables: "publicIpAddressName": "PubIP", and "gatewayPublicIpName": "GWIP". The PubIP variable is used in conjunction with the concatenate function to provide the unique names PubIP0 and PubIP1. These resources are assigned to the virtual machines, FE1 and BE1 in the template.

## Network Gateway resources

The Azure Network Gateway resources include Local Network Gateways and Virtual Network Gateways. A complete description of the APIs that comprise these resources is found here:

<https://msdn.microsoft.com/en-us/library/azure/mt163859.aspx>.

As with any resource, these resources may be created by using either PowerShell cmdlets or ARM templates.

Consider the following PowerShell commands that could be used to create the same VPN deployment in the lab environment as the template:

```
$rgname = RG-AZITCAMP-LAB03
$loc = "West US"
$vnet = Get-AzureVirtualNetwork -Name Lab03_VNET `
  -ResourceGroupName "$rgname"

$GWSubnet = Get-AzureVirtualNetworkSubnetConfig -Name `
  'GatewaySubnet' -VirtualNetwork '$vnet'

$gwip = New-AzurePublicIpAddress -Name gwip `
  -ResourceGroupName "$rgname" -Location "$loc" `
  -AllocationMethod Dynamic

$gwipconfig = New-AzureVirtualNetworkGatewayIpConfig -Name `
  vnetGatewayConfig -SubnetId $GWSubnet.Id -PublicIpAddressId $gwip.Id

New-AzureVirtualNetworkGateway -Name VNETGW -ResourceGroupName `
  "$rgname" -Location "$loc" `
  -IpConfigurations $gwipconfig -GatewayType vpn

New-AzureLocalNetworkGateway -Name LocalSite -ResourceGroupName `
```

```
"$rgname" -Location "$loc" -GatewayIpAddress "128.136.x.y" `
-AddressPrefix "192.168.10.0/24"
```

ARM resources in the template require the same settings as the PowerShell cmdlets. Examine the New-AzureVirtualNetworkGateway cmdlet above. To create a new Azure Virtual Network resource, we need to give the virtual network gateway a name, specify a resource group, specify a location, ensure that a Public IP address is available for it use, ensure a private IP address (dynamic or static) is allocated, specify a Gateway type, and so on.

The ARM template provides identical information in a different format. Otherwise, there is no difference between the resource defined by the template and the PowerShell cmdlet, as shown below.

Note that in this resource, we use the "dependsOn" element to ensure that a Public IP address is available.

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/virtualNetworkGateways",
  "name": "[variables('gatewayName')]",
  "location": "[resourceGroup().location]",
  "dependsOn": [
    "[concat('Microsoft.Network/publicIPAddresses/',
variables('gatewayPublicIPName'))]",
    "[concat('Microsoft.Network/virtualNetworks/',
variables('VNetName'))]"
  ],
  "properties": {
    "ipConfigurations": [
      {
        "properties": {
          "privateIPAllocationMethod": "Dynamic",
          "subnet": {
            "id": "[variables('GWsubnet-id')]"
          },
          "publicIPAddress": {
            "id":
"[resourceId('Microsoft.Network/publicIPAddresses',variables('gatewayP
ublicIPName'))]"
          }
        },
        "name": "vnetGatewayConfig"
      }
    ],
    "gatewayType": "Vpn",
    "vpnType": "RouteBased",
    "enableBgp": false
  }
}
```

- ★ Note that, by default, the virtual network gateway is configured using the Default SKU. The VPN default SKU provides up to ~80 Mbps throughput for a site-to-site VPN (~500 Mbps for ExpressRoute) and up to 10 site-to-site-tunnels. The VPN high perform SKU can provide up ~200 Mbps throughput for a site-

to-site VPN (~1000 Mbps for ExpressRoute) and up to 30 site-to-site tunnels. To create a VPN that uses the high performance SKU or to modify an existing VPN that uses the default SKU, you can use the "gatewaySize" element in the virtualNetworkGateways resource to specify the SKU. Please see <https://msdn.microsoft.com/en-us/library/azure/mt130667.aspx> and <https://azure.microsoft.com/en-us/blog/azure-virtual-network-gateway-improvements/> for more information.

Likewise, the PowerShell cmdlet to create the Local Gateway, which refers to the settings for the on premise or remote network, provides the same settings as the resource defined in the template. In the cmdlet, we need to specify a name for the local gateway, the resource group, the location, the IP address of the endpoint connection, and an address range for the local site.

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/localNetworkGateways",
  "name": "[variables('localGatewayName')]",
  "location": "[resourceGroup().location]",
  "properties": {
    "localNetworkAddressSpace": {
      "addressPrefixes": [
        "[variables('localGatewayAddressPrefix')]"
      ]
    },
    "gatewayIpAddress": "[parameters('localGatewayIpAddress')]"
  }
}
```

## Complete Site-To-Site VPN Configuration

Site-to-site VPNs are used to create secure connections between Azure sites on-premises locations or other virtual networks (more typically known as Vnet-to-Vnet). Point-to-site VPNs are used to create a secure connection between an Azure VNet and a local computer, when it is not possible or desirable to use a VPN device to create the secure connection. Site-to-site, Vnet-to-Vnet, and point-to-site connections all require that you configure a virtual network gateway in Azure.

To configure a virtual network gateway in Azure, you need to determine the Gateway SKU, Basic, Standard or High Performance you want to use. As you learned on the previous lab exercise, the VPN default (standard) SKU provides up to ~80 Mbps throughput for a site-to-site VPN (~500 Mbps for ExpressRoute) and up to 10 site-to-site-tunnels. The VPN high perform SKU can provide up ~200 Mbps throughput for a site-to-site VPN (~1000 Mbps for ExpressRoute) and up to 30 site-to-site tunnels. In the lab environment, the virtual network gateway is created using the Standard (Default) SKU.

As part of your virtual network gateway configuration, you need to specify the gateway types, either static routing (aka, policy-based VPN) or dynamic routing (aka route-based VPN). The setting you use in the virtual network gateway configuration is determined by the VPN device used on the other end of the VPN connection, your on-premises VPN device. For example, RRAS in Windows Server 2012 R2 supports only dynamic routing. Furthermore, not all configurations are possible using static (policy-based) routing. For example, point-to-site or multi-site configuration are not possible using static routing.

Once you have made the choices with regard to Gateway SKU and gateway type and have configured the virtual network gateway, you need to configure your VPN device. In the classic mode, after configuring the VPN gateway, you could download a configuration file to assist you the configuration on the local VPN device. However, at the time of this writing, the preview portal does not provide a link to download the configuration file. Therefore, to configure the VPN, you will need to follow device-specific instructions that are available here: <https://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-about-vpn-devices/>.

In the following lab exercise, you will complete the configuration of the VPN connection settings and the local VPN device to establish a secure connection between the on-premises network (192.168.10.0/24) and the Azure Vnet (10.0.0.0/16).

### Configure local VPN device for site-to-site VPN connection

Once the VPN gateway has finished provisioning, you can configure the local on-premise VPN device to connect to the Azure VPN gateway.

In this task, you will configure AZRCAMP-EDGE server to connect to the Azure VPN gateway.

Please note that, although these steps assume you are using the provided lab environment, these steps will work for any Windows Server 2012 R2 server configured with RRAS, as long as it is not behind a NAT device. If you are not using the provided lab environment are using some other VPN device, you will need to follow instructions specific for your device. You can find these instructions at <https://azure.microsoft.com/en-us/documentation/articles/vpn-gateway-about-vpn-devices/>.

To configure a site-to-site VPN, you require a public IP address that is connected to your VPN device. It is important to note that, while your VPN device may be behind a firewall device or load balancer of some kind, it cannot be behind a NAT (Network Address Translation) device.

❖ These instructions assume that the template deployment you initiated earlier has successfully completed deploying. If the template has not finished deploying, please wait until it has done so.

✎ Perform the following tasks on AZRCAMP-ADMIN and AZRCAMP-EDGE logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

✦ Note: If the virtual machine is not already logged on, on the menu bar, click **Commands**, and then click **Ctrl+Alt+Delete** to display the log on screen.

1. On AZRCAMP-ADMIN, in the Windows ISE PowerShell console you left open in the previous exercise, type the following command and press ENTER.

```
Get-AzurePublicIpAddress -Name gwip -ResourceGroupName RG-AZITCAMP-LAB03 | select name, IPAddress
```

✦ The output of the command displays the public IP address of the VPN gateway.

2. Record the IP address for use in later steps.
3. In Windows PowerShell ISE console, click **File** and then click **Open**.
4. In the Open dialog box, navigate to **C:\LabFiles\AZITPROCAMP\Lab03**.
5. Click **AZVPNConnectionConfig.ps1**, and click **Open**.  
✦ The AZVPNConnectionConfig.ps1 script runs the `New-AzureVirtualNetworkGatewayConnection` cmdlet to configure the connection in Azure between the remote site VPN endpoint and the VPN gateway in Azure. In subsequent steps, you will configure the on premises VPN device (RRAS) and initiate the VPN connection.
6. On the ribbon, click **Run Script**.
7. After a few moments, the output of the command returns with something similar to the following:

```
VirtualNetworkGateway1 : Microsoft.Azure.Commands.Network.Models.PSVirtualNetworkGateway
VirtualNetworkGateway2 : Microsoft.Azure.Commands.Network.Models.PSLocalNetworkGateway
LocalNetworkGateway2    : Microsoft.Azure.Commands.Network.Models.PSLocalNetworkGateway
ConnectionType           : IPSec
RoutingWeight            : 10
Overload                  : abc123
VirtualNetworkGateway1Test : /subscriptions/2a487473-acac-4d4c-910a-e6f2c0a75232/resourcegroups/RG-AZITCAMP-LAB03/providers/microsoft.network/virtualnetworkgateways/VNET04
VirtualNetworkGateway2Test : /subscriptions/2a487473-acac-4d4c-910a-e6f2c0a75232/resourcegroups/RG-AZITCAMP-LAB03/providers/microsoft.network/localnetworkgateways/LocalSite04
LocalNetworkGateway2Test : /subscriptions/2a487473-acac-4d4c-910a-e6f2c0a75232/resourcegroups/RG-AZITCAMP-LAB03/providers/microsoft.network/localnetworkgateways/LocalSite04
ResourceGroupname        : RG-AZITCAMP-LAB03
Location                  : []
Tag                       : {}
TagsTable                 : {}
Name                      : s2svpn
ITag                      : /subscriptions/2a487473-acac-4d4c-910a-e6f2c0a75232/resourcegroups/RG-AZITCAMP-LAB03/providers/microsoft.network/connections/s2svpn
ID                        :
```

8. Leave the Windows PowerShell ISE console open for subsequent steps.
9. Switch to the **AZRCamp-Edge** in Launchpad.
10. On the taskbar, right-click **Windows PowerShell**, and click **Window PowerShell ISE**.



11. In Windows PowerShell ISE console, click **File** and then click **Open**.
12. In the Open dialog box, navigate to **C:\Scripts\**.
13. Click **RRASConfig.ps1**, and click **Open**.
14. In the script, locate the two instances of **<SP\_AzureGatewayIpAddress>**.

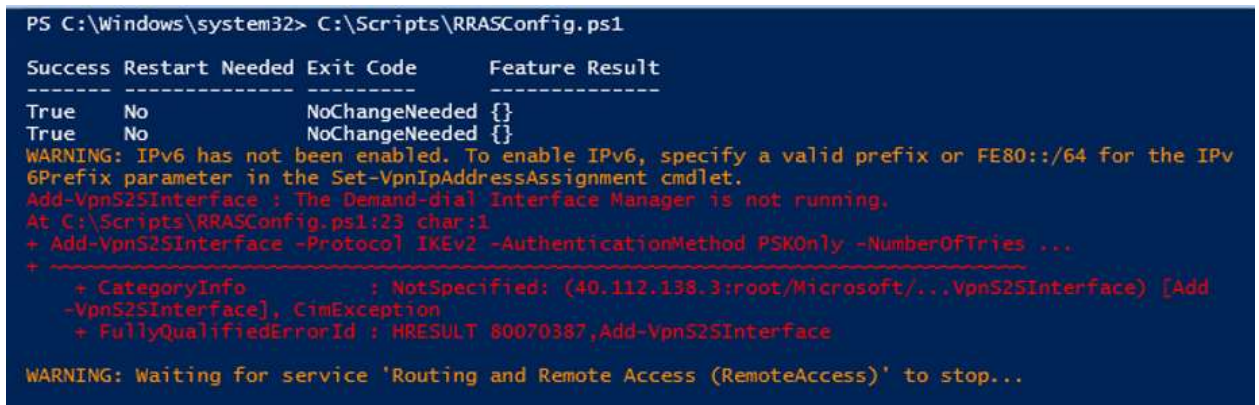


```

22 # Install RRAS role
23 Import-Module ServerManager
24 Install-WindowsFeature RemoteAccess -IncludeManagementTools
25 Add-WindowsFeature -Name Routing -IncludeManagementTools
26
27 # !!! NOTE: You may be required to reboot before continuing on with the script.
28
29 # Install S2S VPN
30 Import-Module RemoteAccess
31 Install-RemoteAccess -VpnType VpnS2S
32
33 # Add S2S VPN interface
34 Add-VpnS2SInterface -Protocol IKEv2 -AuthenticationMethod PSKOnly -NumberOfTries 3 -ResponderAuthenticationMethod PSKOnly -Name <SP_AzureGatewayIpAddress> -DestIpAddress <SP_AzureGatewayIpAddress> -SPv4Sub
35
36 # Restart the RRAS service
37 Restart-Service RemoteAccess
38
39 # Optional: Set to Azure gateway (optional)
40 # Connect-VpnS2SInterface -Name <SP_AzureGatewayIpAddress>
    
```

15. Replace both instances of **<SP\_AzureGatewayIpAddress>** with the IP address you determined in step 1 above.
16. On the ribbon click **Run Script**.
17. Click **OK**.

❖ In the output of the command, you may see an error, such as the following. This is expected. RRAS may not be reporting its status accurately.



```

PS C:\Windows\system32> C:\Scripts\RRASConfig.ps1

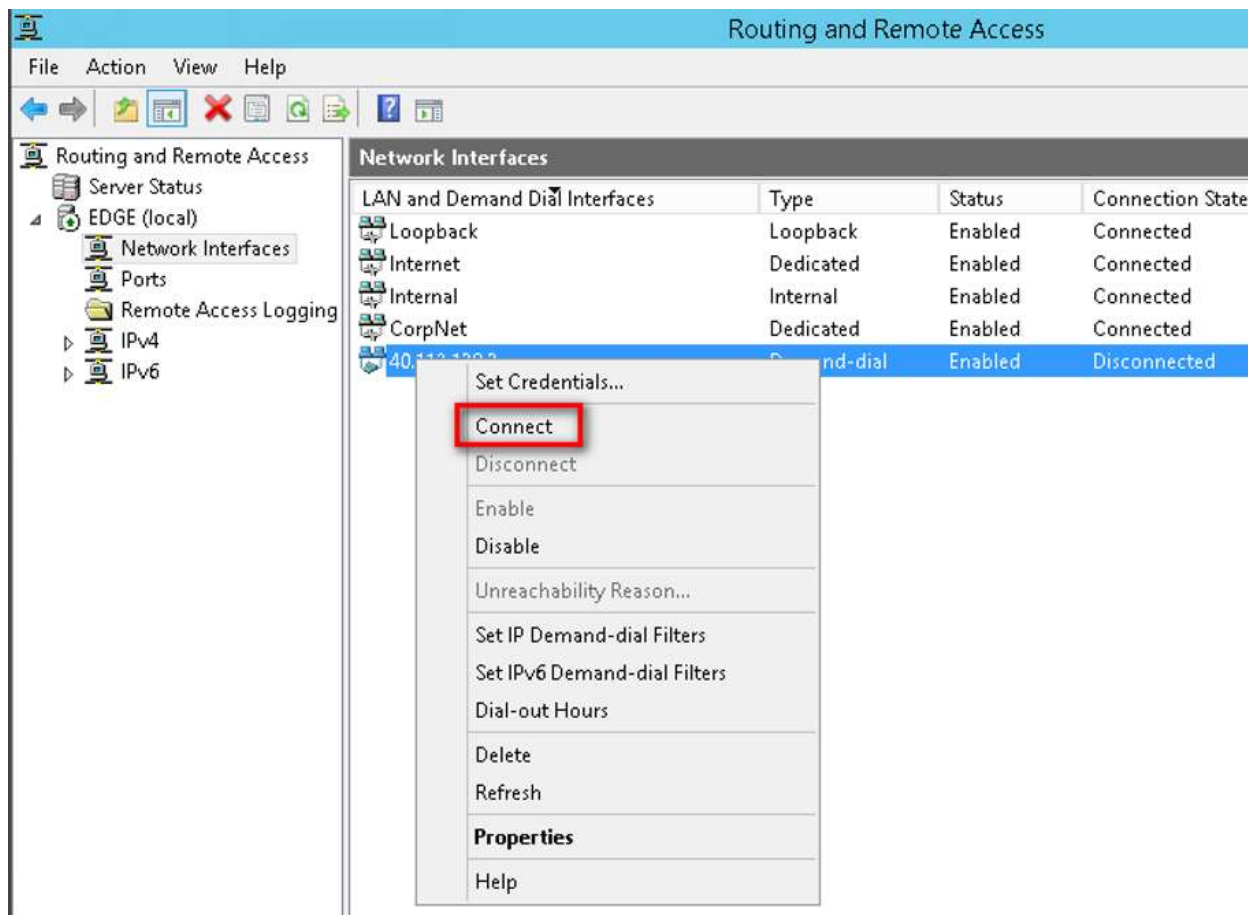
Success Restart Needed Exit Code      Feature Result
-----
True      No           NoChangeNeeded {}
True      No           NoChangeNeeded {}

WARNING: IPv6 has not been enabled. To enable IPv6, specify a valid prefix or FE80::/64 for the IPv6Prefix parameter in the Set-VpnIpAddressAssignment cmdlet.
Add-VpnS2SInterface : The Demand-dial Interface Manager is not running.
At C:\Scripts\RRASConfig.ps1:23 char:1
+ Add-VpnS2SInterface -Protocol IKEv2 -AuthenticationMethod PSKOnly -NumberOfTries ...
+ ~~~~~
+ CategoryInfo          : NotSpecified: (40.112.138.3:root/Microsoft/...VpnS2SInterface) [Add-VpnS2SInterface], CimException
+ FullyQualifiedErrorId : HRESULT 80070387,Add-VpnS2SInterface

WARNING: Waiting for service 'Routing and Remote Access (RemoteAccess)' to stop...
    
```

18. Open **Server Manager**.
19. In Server Manager, click **Tools**, and then click **Routing and Remote Access**.
20. In Routing and Remote Access console, in the tree pane, expand **Edge**, and then click **Network Interfaces**.
21. You should see the network interface that is represented by the IP Address of the VPN Gateway. If you do not see the interface, return to the Windows PowerShell ISE console, run the command **restart-service RemoteAccess**, and then rerun the **RRASConfig.ps1** script.
22. Right-click the connection, and click **Connect**.

- ✦ In a few moments, the connection should be established.



23. Switch to AZRCAMP-ADMIN.

- ✦ In these next steps, you will verify that the VPN site-to-site connection is functioning as intended.

24. In the Windows PowerShell ISE console, in the command pane, type the following command, and press ENTER.

```
(Get-AzureNetworkInterface).IpConfigurations.PrivateIpAddress
```

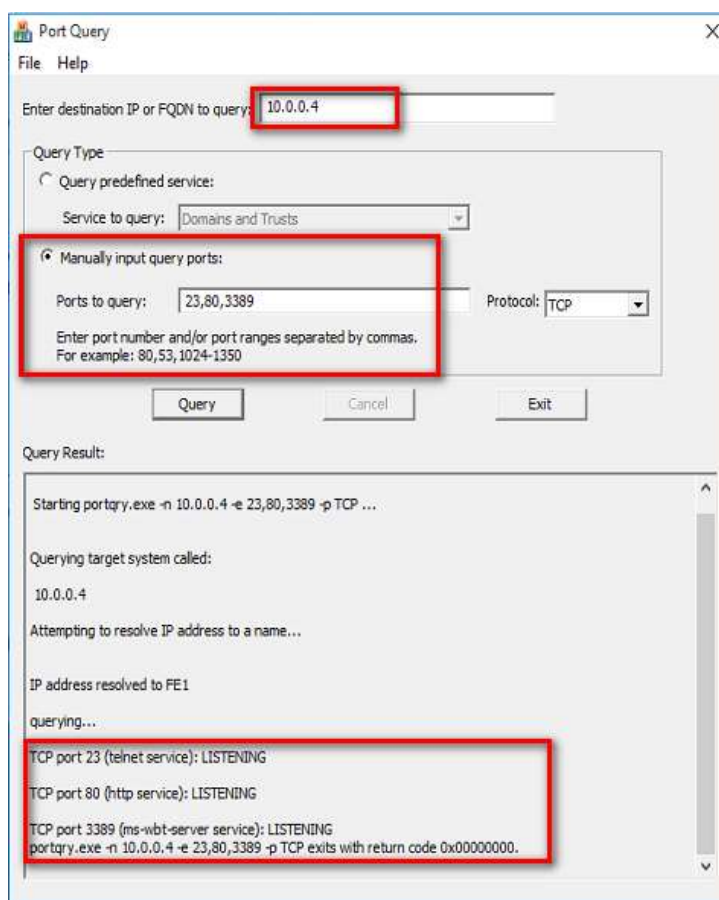
- ✦ The output of the command shows the two private IP address that are assigned to the virtual machines BE1 and FE1 in the lab environment. The IP addresses will likely be 10.0.1.4 and 10.0.0.4, as .4 is the first address in the octet assigned to the first virtual machines that are provisioned in a subnet.

```
PS C:\Users\admin.CONTOSO> (Get-AzureNetworkInterface).IpConfigurations.privateipaddress
10.0.1.4
10.0.0.4
PS C:\Users\admin.CONTOSO> |
```

25. Open **File Explorer**, and navigate to **C:\LabFiles\Utils\PortQryUI**.

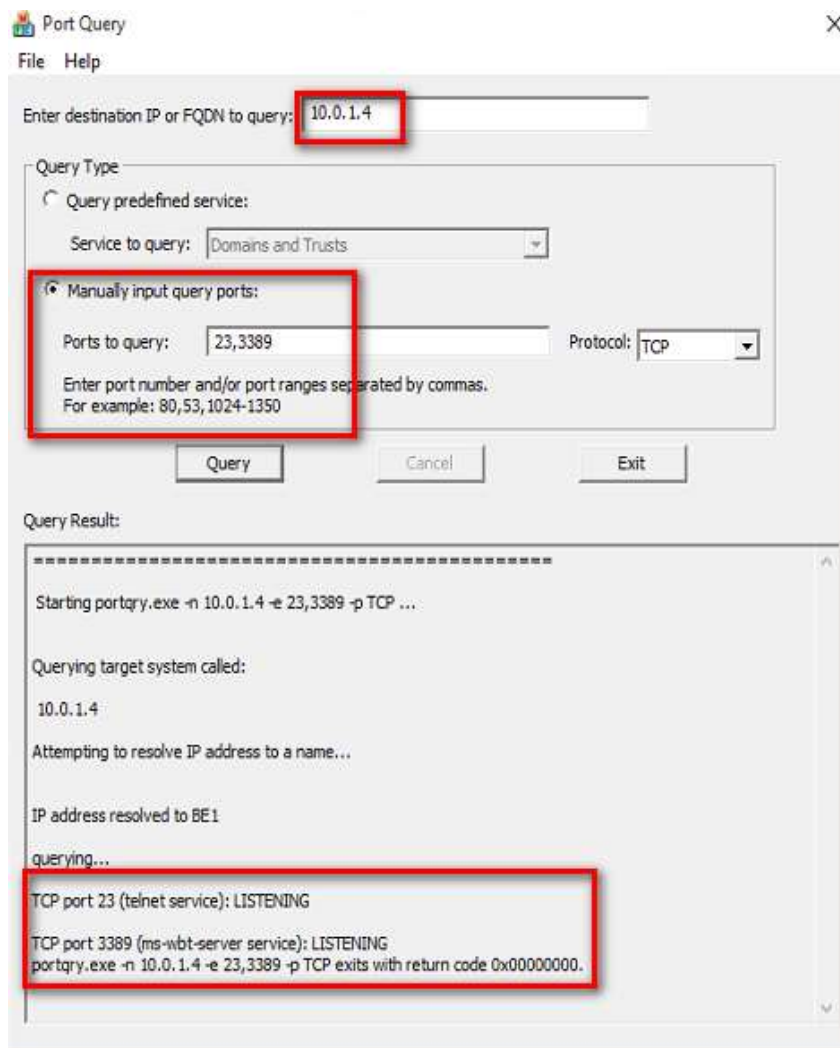
26. Double-click **portqueryui.exe**.
27. In Enter destination IP or FQDN to query, type **10.0.0.4** (or the actual private IP address of FE1, if different).
28. Click **Manually input query ports**.
29. In Ports to Query, type **23, 80, 3389**, and click **Query**.

✦ The Port Query application almost instantly displays the output showing the status of the 3 ports on FE1 as LISTENING. This means that the ports are open. Had any of the ports been blocked, the output would have taken a few moments to be displayed and the status for the blocked ports would be denoted by FILTERED.



30. Repeat steps 38-30 to query BE1 at its private IP address 10.0.1.4 to see if the ports 23 and 3389 are accessible through the VPN connection.

✦ IIS is not installed on BE1.



31. Leave PortQuery open for subsequent lab steps.

## Configure Network Security Groups by using ARM Templates

Network Security Groups (NSG) are a recent feature of Microsoft Azure that allow you to configure firewall policies at the network level to control the flow of inbound and outbound traffic to and from virtual machine instances. An NSG contains access control rules, similar to firewall rules that allow or deny access to virtual machines in your subscription. Unlike network ACLs, which can only be applied to the public endpoints, an NSG can be applied to VMs, NICs and subnets. This means that all instances virtual machines that reside in a subnet can have the same access control rules applied to them and that each virtual machine can have access control rules specifically tailored for them. If a VM has multiple NICs, and an NSG is applied to only one of the NICs, the other NICs are not subject to the access control rules of the NSG.

Only one NSG can be associated with each subnet, VM, or NIC. However, Network Security Groups can contain up one to 200 rules. NSG rules have the following characteristics:

- NSGs contain rules that consist of the following properties:
  - **Name:** unique identifier for the rule
  - **Type:** Inbound/Outbound
  - **Priority:** an integer between 0 and 4096 that determine the order of processing; lower priority number is processed before higher priority numbers.
  - **Source IP address:** CIDR of source IP range
  - **Source Port Range:** Range between 0 and 65500
  - **Destination IP Range:** CIDR of destination IP range
  - **Protocol:** TCP, UDP or \* for all
  - **Access:** Allow/Deny
- Rules are stateful. This means that NSGs keep track of the communication. When you create an inbound rule, it is not necessary to configure a corresponding outbound rule and vice versa to enable bi-direction communication between the local and remote entity. For example, when you configure an inbound rule to provide inbound access to a Web server on TCP port 80, you do not have to create a corresponding outbound rule to allow the Web server to respond to the request on port 80.
- Rules are processed according to priority value, which is an integer between 100 and 4096. The lower the number, the higher the priority (order of processing). When a traffic match is made by a higher priority rule, processing of rules stops for that particular traffic.
- When you implement a NSG on a subnet, VM, or NIC, a set of default rules is also implemented. The default rule can be overridden by rules that have a lower priority value. The default rules are as follows:
  - **Allow VNET inbound:** Allows inbound traffic from all hosts in the VNET. Priority = 65,000.
  - **Allow Azure Load Balancer Inbound:** Allows Azure load balancer to probe health of VM. Priority = 65,001.
  - **Deny all Inbound:** Denies all inbound traffic. Priority = 65,500.

- **Allow all VNET Outbound:** Allows outbound traffic from all hosts in the VNET to all hosts in the VNET. Priority = 65,000.
- **Allow Internet Outbound:** Allows all outbound traffic for the Internet outbound. Priority = 65,001
- **Deny all outbound:** denies all outbound traffic. Priority = 65,500.

Some other important characteristics of NSGs include the following:

- Default tags and special characters are used to identify a category of IP addresses. The default tags are:
  - **VIRTUAL\_NETWORK:** denotes the entire address space of the virtual network.
  - **AZURE\_LOADBALANCER:** denotes Azure's load balancer.
  - **INTERNET:** denotes the Internet
  - \*: Special character to indicate "all".
- NSGs are available only for virtual machines in a regional Vnet: they are not available for Vnets associated with affinity groups.
- Endpoint-based access control lists (ACLs) and NSGs are not compatible for the same virtual machine. You must use either ACLs or NSGs, not both.
- You can have up to 100 NSGs per subscription.

## Example: Defense in depth using NSGs

The flexibility and scope provided by NSGs enable defense-in-depth scenarios, where NSGs at the network level can be combined with firewall rules at the OS level. Because NSGs can control traffic at the network level between subnets, they can be used help realize secure multi-tier application architectures. In these architectures, typically only one tier of the application is exposed to the Internet in a DMZ. The remaining tiers of the application are segmented from the DMZ by firewalls. Generally, following the principle of least privilege, no hosts in any of the tiers would be allowed to initiate communication with hosts on the Internet. Furthermore, only the minimum traffic necessary for the application is allowed to traverse the firewalls that segment the application tiers from each other.

✦ In the case of Azure IaaS deployment, exceptions would be required for the KMS license server and any PaaS services hosted in Azure, such as a SQL database. For KMS, the Azure virtual machines would need to be able to communicate with the Internet on TCP port 1688.

Consider a 3-tier application that comprises a Web front-end, a business logic middle-tier, and a backend data-tier. In this case, the only traffic allowed from the Internet would be Web traffic on TCP ports 80 and 443. The Web servers in the DMZ would not be allowed to initiate traffic to the Internet. The Web servers in the DMZ would be able to communicate across a firewall to the middle tier, perhaps on well-known ports or on custom ports. The servers in the middle tier would be allowed to initiate communication with the DMZ or the Internet, but would be allowed to communicate with the data tier. The servers in the data tier would not be able to initiate communication to host in the internet or the other tiers of the applications.

## NSG ARM template examples for Network Security Groups

To create a Network Security Gateway resource, we need to use the **Microsoft.Network/networkSecurityGroups** resource, which can be found here:

<https://msdn.microsoft.com/en-us/library/azure/mt163615.aspx>.

The following shows a somewhat abbreviated version of the elements that comprise the NSG resource:

```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/networkSecurityGroups",
  "name": "NSG_Name",
  "location": "NSG_Location",
  "properties": {
    "securityRules": [
      {
        "name": "rule-name",
        "properties": {
          "description": "rule-description",
          "protocol": "Tcp,UDP, or *",
          "sourcePortRange": "source-port-range",
          "destinationPortRange": "destination-port-range",
          "sourceAddressPrefix": "CIDR-of-source-IP-range",
          "destinationAddressPrefix": "destination-CIDR-range",
          "access": "Allow or Deny",
          "priority": integer-value-for-rule-priority,
          "direction": "Inbound or Outbound"
        }
      }
    ]
  }
}
```

It is possible to create an NSG without associating it with a subnet. However, we would typically want to associate the NSG with a subnet (or, using classic mode, a VM or NIC). To associate a rule with a subnet, we need to use the **Microsoft.Network/virtualNetworks** resource, which can be found here:

<https://msdn.microsoft.com/en-us/library/azure/mt163650.aspx>.

For clarity, the following shows an abbreviated version of the elements in the virtualNetworks resource that enable the association of an NSG with a subnet:



```
{
  "apiVersion": "2015-05-01-preview",
  "type": "Microsoft.Network/virtualNetworks",
  "name": "name-of-VNET",
  "location": "location",
  "dependsOn": [
  ],
  "properties": {
    "addressSpace": {
      "addressPrefixes": [
        "10.0.0.0/16"
      ]
    },
    "subnets": [
      {
        "name": "subnet1",
        "properties": {
          "addressPrefix": "10.0.0.0/24",
          "networkSecurityGroup": {
            "id": .../networkSecurityGroups/NSG-name"
          }
        }
      }
    ]
  }
}
```

### More Information

For more information, please see the following blog posts and articles:


<http://azure.microsoft.com/blog/2014/11/04/network-security-groups/>

<https://azure.microsoft.com/en-us/documentation/articles/virtual-networks-nsg/>

<http://blog.kloud.com.au/2014/11/07/secure-azure-virtual-network-and-create-dmz-on-azure-vnet-using-network-security-groups-nsg/>

### Verify connectivity to virtual machines

In the previous lab exercise, you verified connectivity. In this lab exercise, you will verify the connectivity to FE1 and BE1 to establish a baseline for determining the effects of associating Network Security Groups with subnets in the subsequent lab exercise.

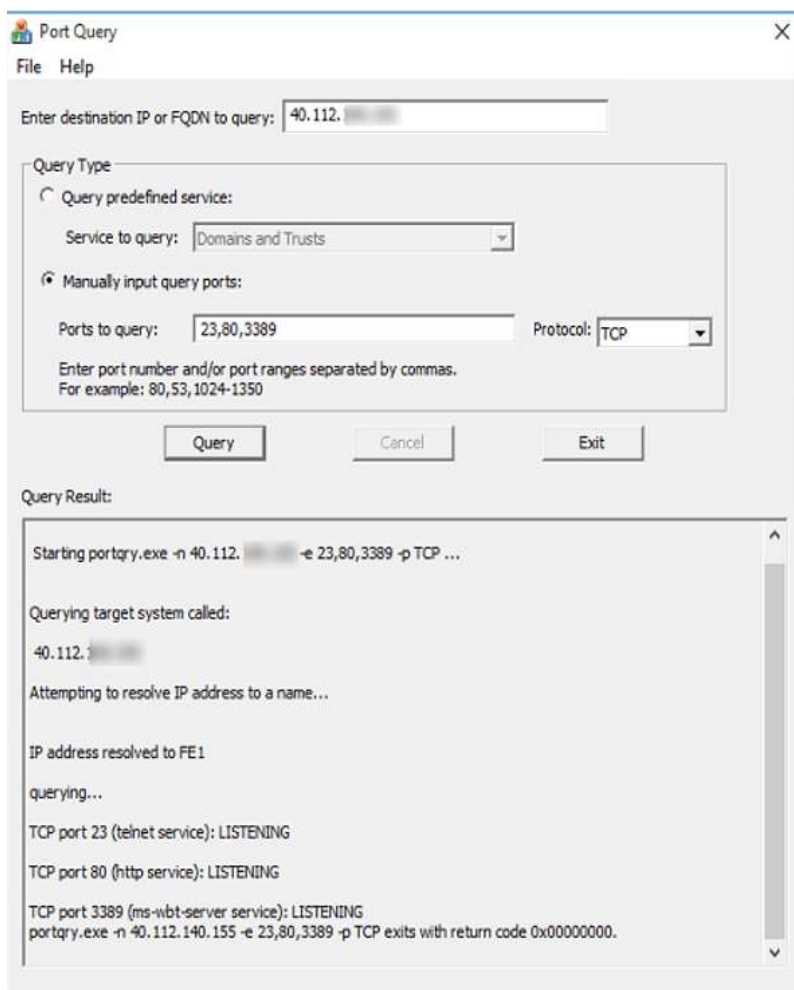
-  Perform the following tasks on AZRCAMP-ADMIN logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

1. In the Windows PowerShell ISE console that you left open in the previous exercise, in the command pane, type the following command on a single line, and press ENTER.

```
↩ Get-AzurePublicIpAddress | where {$_.Name -like "Pub*"} | Select  
Name, IPAddress
```

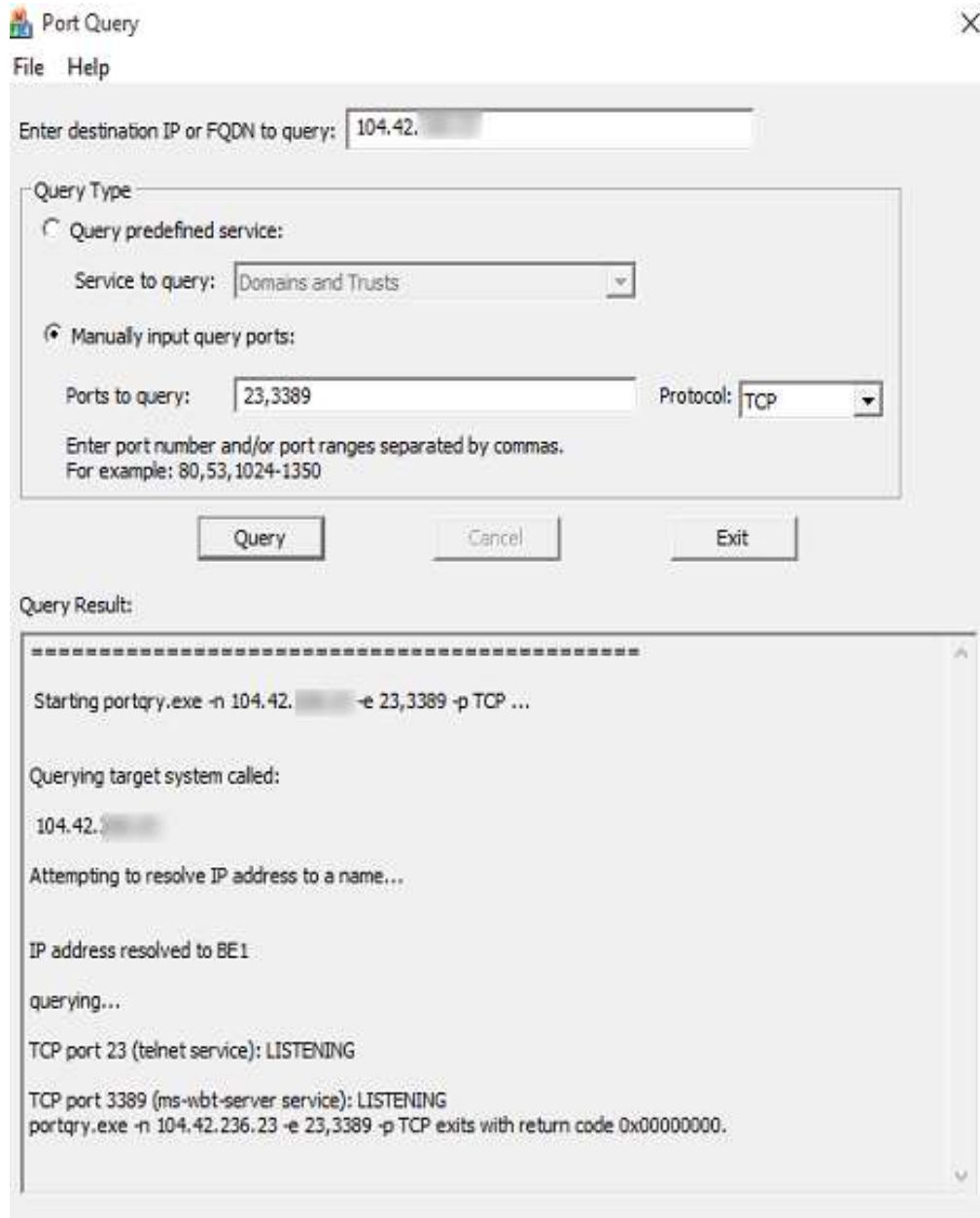


- ✦ The output of the command shows the two public IP address that are assigned to the virtual machines FE1 and BE1 in the lab environment. PupiP0 corresponds to FE1; PubIP1 corresponds to BE1.
- 2. Record both IP addresses that are displayed in the output.
- 3. Open File Explorer, if not already open, and navigate to **C:\LabFiles\Utils\PortQryUI**.
- 4. Double-click **portqueryui.exe**.
- 5. In Enter destination IP or FQDN to query, type **[PubIP0]** (where PubIP0 is the public IP address of FE0).
- 6. Click **Manually input query ports**.
- 7. In Ports to Query, type **23, 80, 3389**, and click **Query**.
- ✦ The Port Query application almost instantly displays the output showing the status of the 3 ports on FE1 as LISTENING. This means that the ports are open. Had any of the ports been blocked, the output would have taken a few moments to be displayed and the status for the blocked ports would be denoted by FILTERED.



8. Repeat steps 3-5 to query BE1 at its public IP address to verify that ports 23 and 3389 on BE1 are accessible from the Internet.

✦ IIS is not installed on BE1.



## Deploy Network Security Groups by using an ARM Template

In this lab exercise, you will modify an incomplete starter template that contains incomplete set of rules for 2 Network Security Groups. This lab exercise does not provide you with some detailed step-by-step instructions for completing the exercise. Rather, the lab exercise asks you to create a template to deploy NSGs that meets specific design goals in the scenario provided below.

### Lab Scenario Goals

You have deployed a two-tiered Azure IaaS application. The front-end tier consists of a subnet (10.0.0.0/24) and single Web server. The back-end tier consists of a subnet (10.0.1.0/24) and virtual machine that run an application on TCP port 23. (In actual fact, this is the port for the Telnet server -- in the lab environment, we use this port to stand in for a database application so we can deploy the virtual machines as quickly as possible.) You have a VPN connection between your on-premise environment and the virtual network (10.0.0.0/16).

You need to ensure the following controls are enforced with your network security groups:

1. Clients on the Internet can access the Web server in the frontend subnet using its public IP address.
2. All other Inbound traffic to both the front-end and the back-end subnets is denied.
3. The virtual machines in the frontend (10.0.0.4/24) and the backend (10.0.1.0/24) subnets must be allowed access to the KMS server, which resides on the Internet at TCP port 1688.
4. All other outbound Internet access from the backend and frontend servers is denied.
5. All outbound access from the backend subnet to the frontend subnet is denied.
6. TCP port 23 must be accessible from the frontend subnet to the backend subnet; all other outbound traffic from the frontend to the backend subnet is denied.
7. RDP access on TCP port 3389 from the on-premise subnet (192.168.10.0/24) must be allowed ingress to the frontend and backend networks through the site-to-site VPN for management purposes.
8. All other traffic from 192.168.10.0/24 to the frontend and backend subnets through the site-to-site VPN must be explicitly denied.
9. Optionally, create an exception to allow traffic on TCP port 80 from the on-premises subnet to FE1 through the site-site VPN.
10. Because NSGs can have unpredictable effects on VPNs, no NSGs should be applied to the gateway subnet (10.0.200.0/28).

 Perform the following tasks on AZRCAMP-ADMIN logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

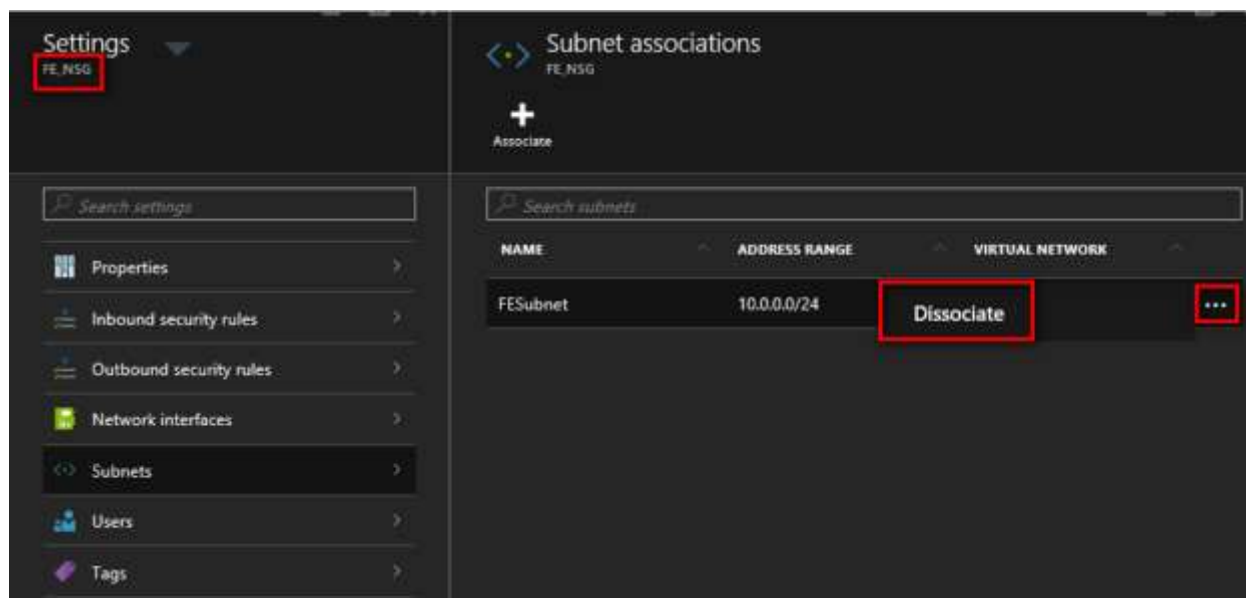
1. Open File Explorer, if not already open.
2. Navigate to **C:\LabFiles\AZITPROCamp\Lab03**.
3. Double-click **start.azuredeploy.nsg.json** to open the file in Visual Studio Code.
4. Start.azuredeploy.nsg.json represents an incomplete deployment of Network Security Group rules to meet the criteria specified above. It can, however, be deployed as is.
  - ✦ Using the criteria above, add the appropriate Network Security Group rules to the start.deploy.nsg.json file.
  - ⚠ You do not have to edit any existing resources, parameters, or variables. You only have to add rules to the backend and frontend subnet NSGs.
  - ✦ If you get stuck or just want to peek quickly at a possible solution for more guidance, please see the azuredeploy.nsg.json file in the **C:\LabFiles\AZITPRO\Camp\Solutions\Lab03** folder.
5. When you have completed the modifications to the start.azuredeploy.nsg.json, save it as **azuredeploy.nsg.json**.
  - ✦ Before saving the file, make sure that the file does not display any red squiggles that indicate a problem with the file format.
6. In the azuredeploy.nsg.json file, press CTRL+A, and then press CTRL+C to copy the entire contents of the file to the clipboard.
7. Leave the azuredeploy.nsg.json open for subsequent steps.
8. Open Edge browser, browse to <https://portal.azure.com>, and log on to your Azure subscription.
9. Click **New**, and then click **Template deployment**.
10. If you do not see Template deployment in the New blade, search for it in the Marketplace.
11. On the Custom deployment blade, click **Edit template**.
12. In the Edit template blade, select all the lines of JSON code and delete them.
13. Press CTRL+V to past the contents of the clipboard (the azuredeploy.nsg.json) file to template area.
  - ✦ When you have pasted the file, note the small green square in the upper right. This indicates that the file has the correct format.
  - ⚠ If you do not see the green square, it is likely you have missed a comma, inserted a comma in a wrong place, have missing braces, etc. You will see a red square indicating the approximate location of the issue. Please resolve the problem before proceeding.
14. Click **Save**.

15. In the Custom deployment blade, click **Edit parameters** to review the parameter values, and then click **OK**.
16. Click **Select a resource group**, and click **RG-AZITCAMP-LAB03**.
17. Click **Legal terms**, and then click **Buy**.
18. In the Custom deployment blade, click **Create**.

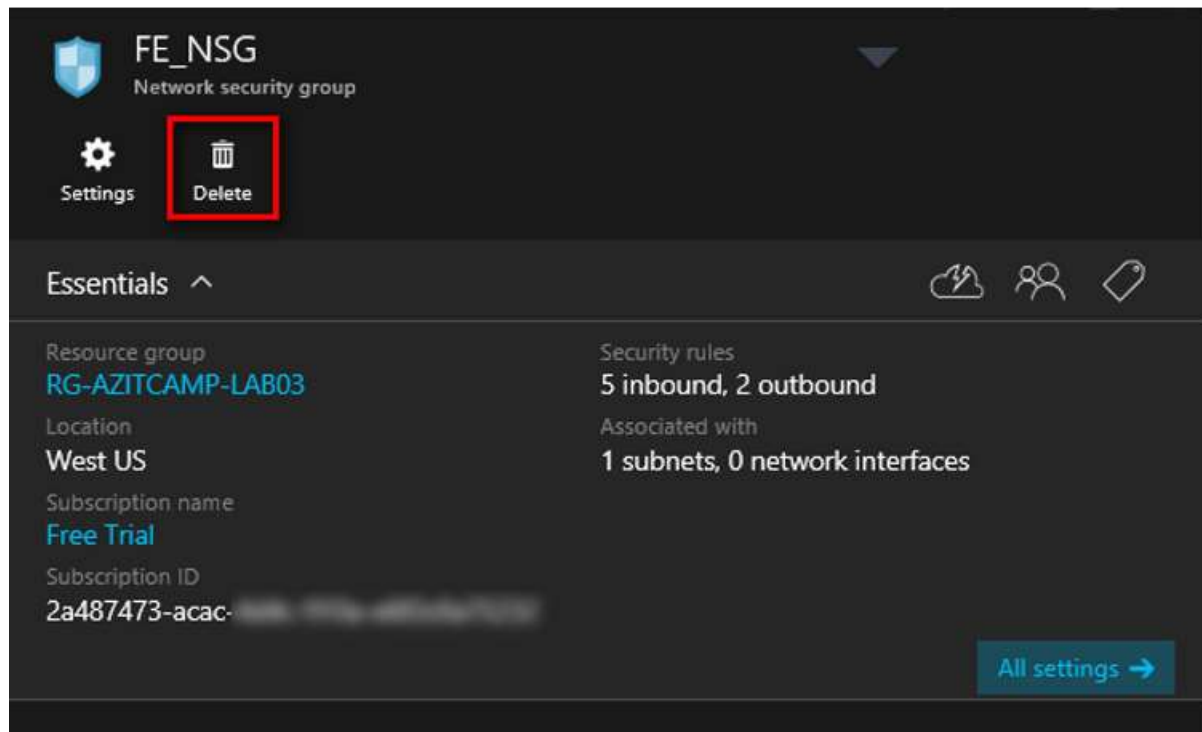
## Troubleshooting template deployment (if necessary)

If your template fails to deploy properly, please examine the error message that accompanies the failure notification. Common reasons for the failure may be related to copying and pasting rules to create new ones from the copies. Please ensure that you use unique names and unique priority values for the rules you create.

Even though templates are idempotent, you may wish to remove your NSG resources and start the deployment over again. However, if your NSG is associated with a subnet, you will not be able to delete it. You must first disassociate the NSG from the subnet. You can do so using a PowerShell cmdlet or you can do so using the preview portal as shown in the screenshot below.



Once you have disassociated the NSG from the subnet, you can delete the NSG resource, as shown below.

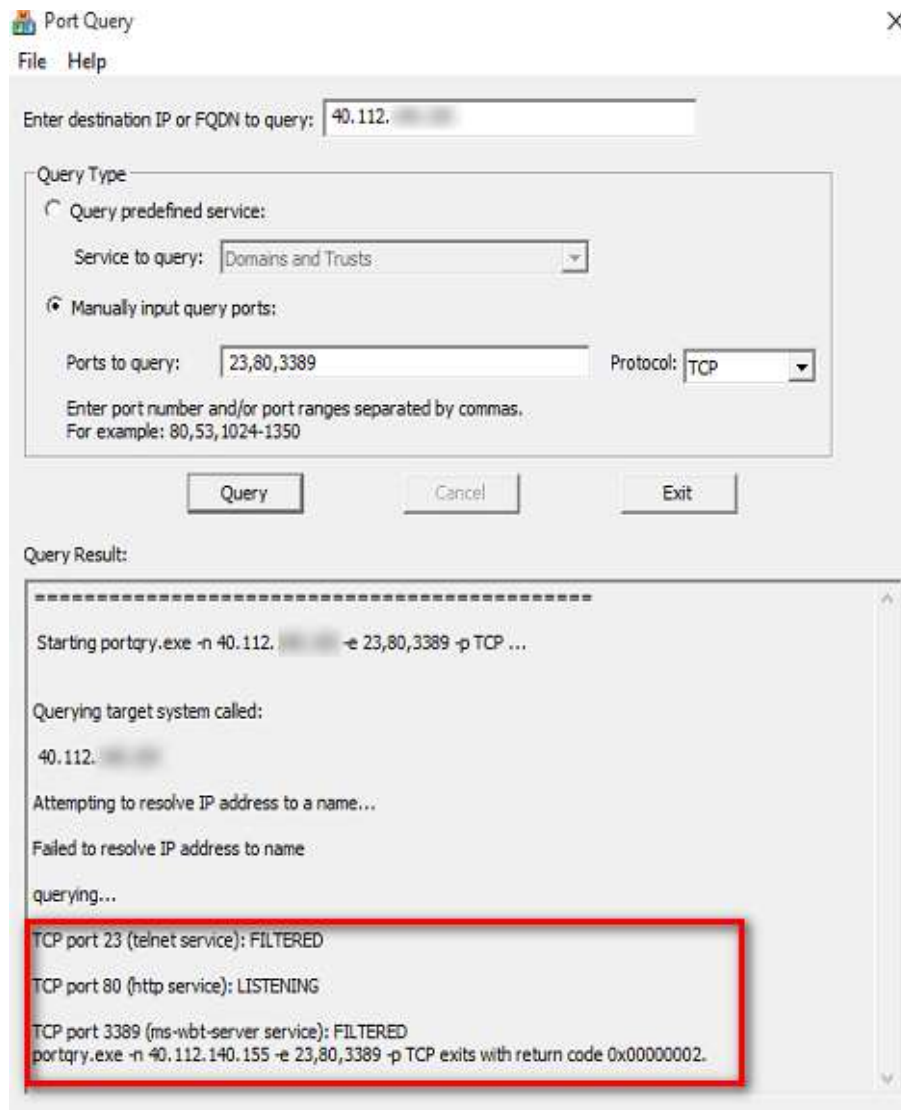


## Verify NSG rules

In the following task, you will verify that your NSG rules are properly implemented.

 Perform the following tasks on **AZRCAMP-ADMIN**:

1. Using the PortQuery application that you left open in a previous step, verify that the only port that is accessible from the Internet for either FE1 or BE1 is port 80 on FE1.
2. You can also use Microsoft Edge to verify that IIS is available at the public IP address for FE1.

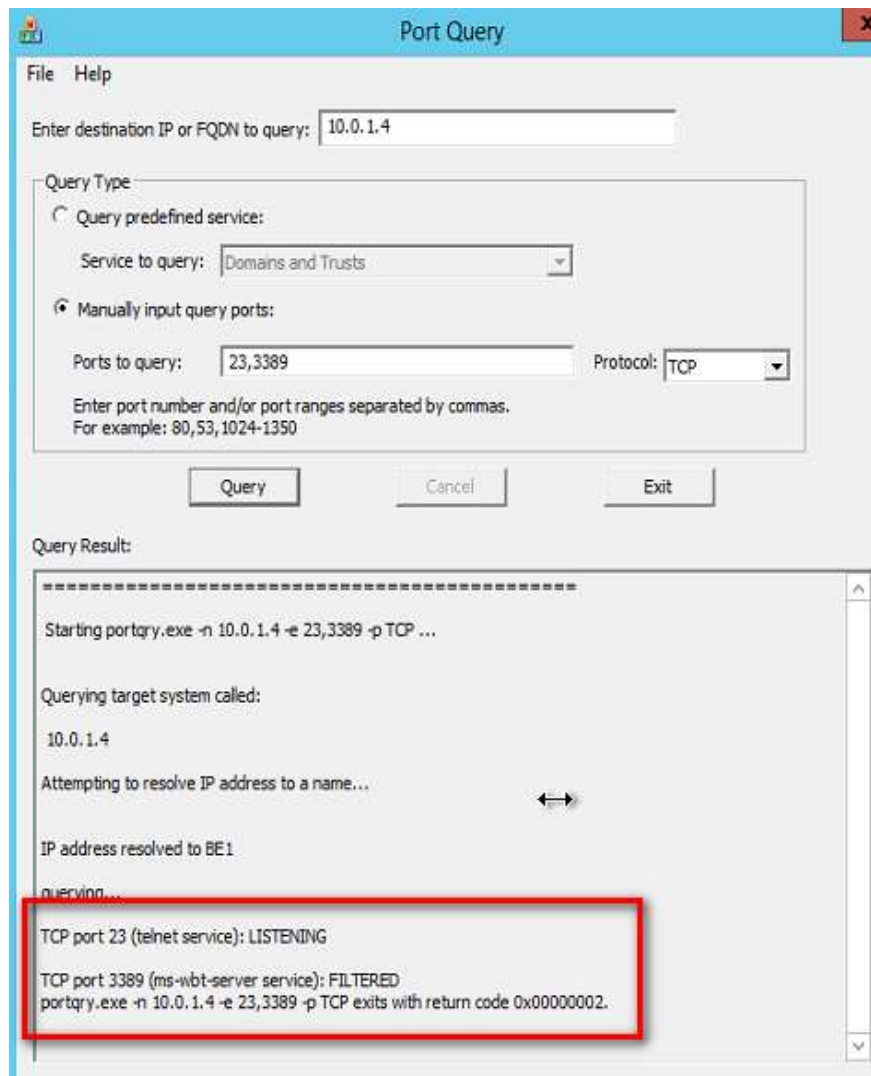


3. Using PortQuery, verify that only TCP port 3389 is available for FE1 and BE1 through the site-to-site VPN.
  - ✦ If you also created the option rule to allow TCP port 80 from the on premise subnet to the frontend subnet, TCP port 80 will also be available.
4. On AZRCAMP-ADMIN, open a PowerShell or a command prompt, if you do not have one of these open already.
5. At the prompt, type **mstsc**, and press ENTER.
6. In the Remote Desktop Connection dialog box, type **10.0.0.4** (or the actual private IP address of FE1, if different), and click **Connect**.
7. In the Windows Security dialog box, click **Use another account**.
8. In User name, enter **ITCampAdmin**; in Password, enter **Passw0rd!**, and click **OK**.

9. In the Remote Desktop Connection dialog box, click **Yes**.
10. When you have logged on to FE1, open **File Explorer**, and navigate **C:\Source**.
11. Right-click **PortQryUI**, and click **Extract All**.
12. On the Select a Destination and Extract Files page, shorten the path to **C:\Source**, and click **Extract**.
13. Open the **PortQryUI** folder, and double-click **portqueryui.exe**.
14. In Enter destination IP or FQDN to query, type **10.0.1.4** (or the actual private IP address of BE1, if different).
15. Click **Manually input query ports**.
16. In Ports to Query, type **23, 3389**, and click **Query**.

✦ The output should show that Port 23 on BE1 is open (LISTENING) and that port 3389 is closed (FILTERED).





17. Repeat steps 6 - 16 to establish a Remote Desktop session with BE1 (10.0.1.4) and to query ports 23 and 3389 on FE1.

✦ All ports on FE1 should be unavailable to BE1.

## Remove resource group used for lab

Because each lab in this series begins with an empty resource and because Azure resources are potentially billable, it is necessary to remove the resource group you created and used in this lab. Also, because Azure trial accounts are limited to 4 compute cores, it is important that you remove the resource group to ensure you do not run out of resources, if you have an Azure trial account.

### Remove Azure resource group

In this task you will run a Windows PowerShell script to remove the resource group you created and used in this lab.

 Perform the following tasks on AZRCAMP-ADMIN logged on as **Contoso\Administrator** using **Passw0rd!** as the password:

1. If not already open, open Windows PowerShell ISE.
2. Click **File**, click **Open**, browse to **C:\LabFiles\AZITPROCamp\Scripts**, select **RGCleanup.ps1**, and click **Edit**.
3. On the menu, click **Run Script**.
4. When prompted, log into your Azure subscription.
5. When prompted to delete **RG-AZITCAMP-LAB03**, click **Yes**.
6. If you used a different resource group for the lab, you can modify the PowerShell script to delete that resource group.