A picture containing indoor, building

Description generated with high confidence

Creating a Virtual Data Center

A Network perspective

Student Guide | September 23, 2018

# Preface

You can configure the Azure workshop resources in three different ways:

1. Using PowerShell scripts
2. Invoking ARM templates via PowerShell
3. Accessing the Azure Portal

Typically, PowerShell scripts or ARM templates are used for mass (scalable, repeatable, and programmable) deployments. Alternatively, the Azure Portal provides a graphical interface that aids learning.

This documentation is divided into two parts. The first part guides you in setting up your environment to deploy resources via PowerShell or ARM templates. The second part guides you through deploying resources via the Azure Portal.

The workshop consists of eight different steps. Each step builds on previous steps. You can also configure different steps via different ways. The first four steps are relatively easy, because each of them focuses on creating one major resource. The last four steps are complex as they create multiple resources. So, if you are going out try different ways of configuring Azure resources during the workshop, the recommendation would be to try initial steps using Azure portal and later steps using either PowerShell scripts or ARM templates.



Network Architecture Diagram of the Virtual Data Center

# Part I

Part I

Executing

PowerShell Scripts or ARM Templates

# Introduction

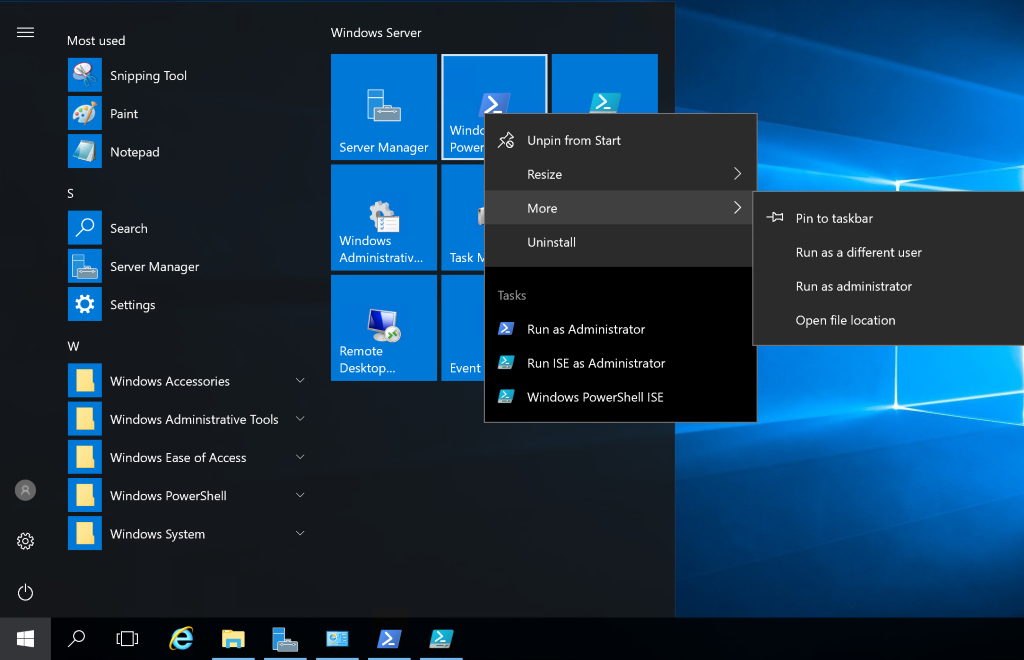
You can find the PowerShell scripts and ARM templates to complete this workshop in the Microsoft Azure flash drive provided to you. ARM templates are .json files that can be executed via PowerShell to deploy resources in Azure.

Go through the following steps to set up your execution environment:

* Set the Execution Policy to *RemoteSigned*
* Install Azure SDK
* Copy the *Scripts* folder to the local VM
* Update the Company ID number in the *init* file
* Load the PowerShell script/ARM template in PowerShell ISE
* Execute the PowerShell scripts or ARM templates

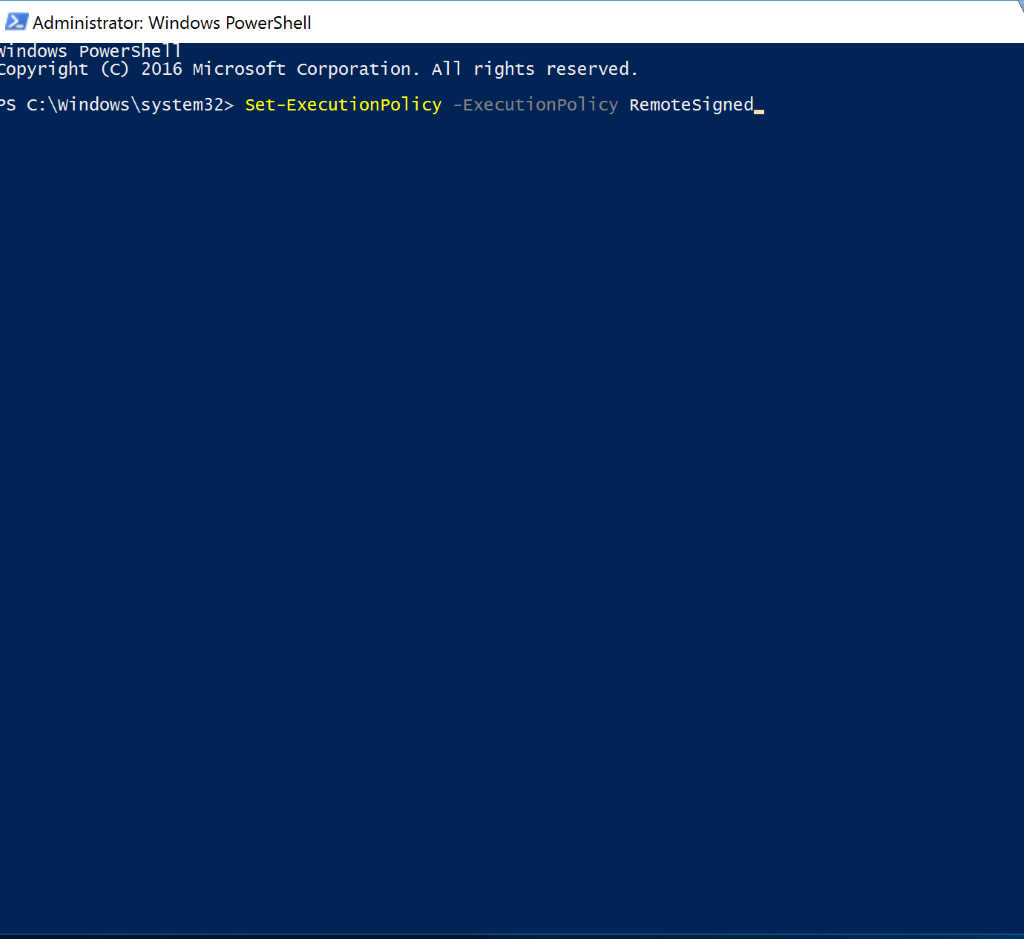
# Set Execution Policy

On your windows desktop, Click on the task bar, right click Windows PowerShell, select *More*, *Run as administrator.*



Once PowerShellopens, run the following command:

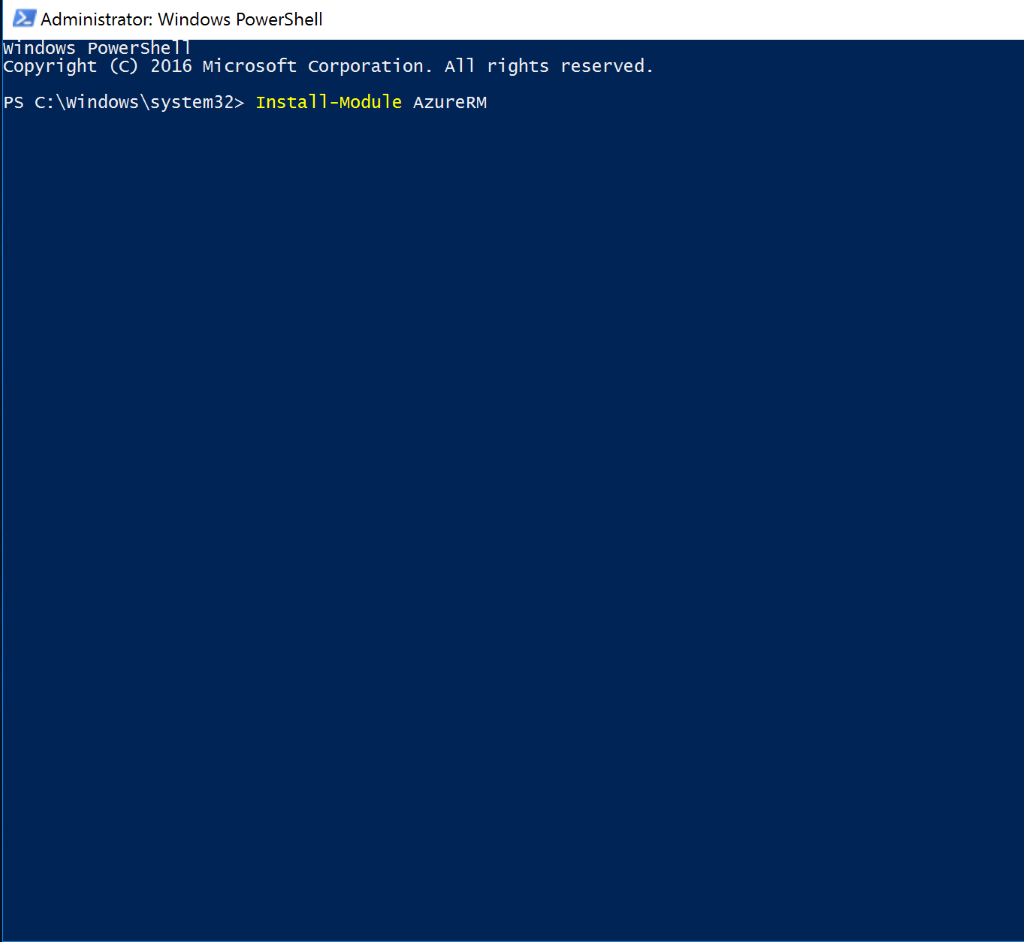
**Set-ExecutionPolicy -ExecutionPolicy RemoteSigned**



# Install Azure SDK

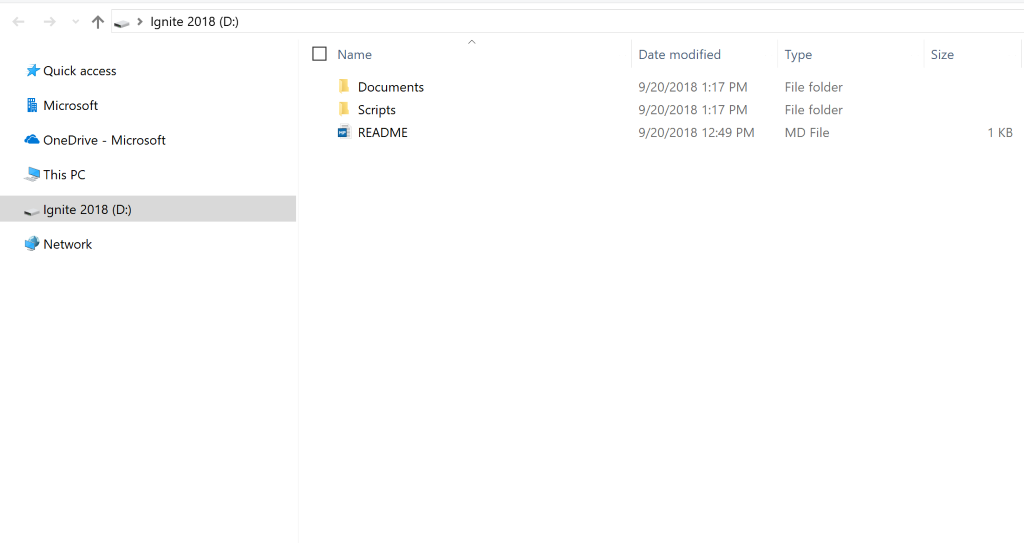
Install the Azure SDK by running the following command:

**Install-Module AzureRM**

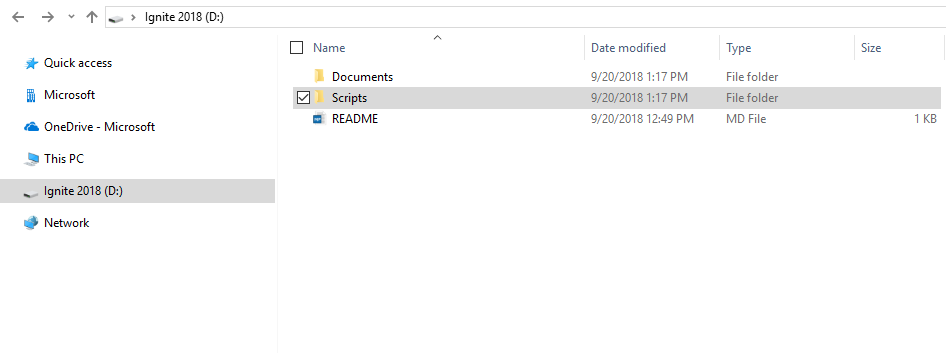


# Copy the Scripts to the VM’s local *Documents* folder

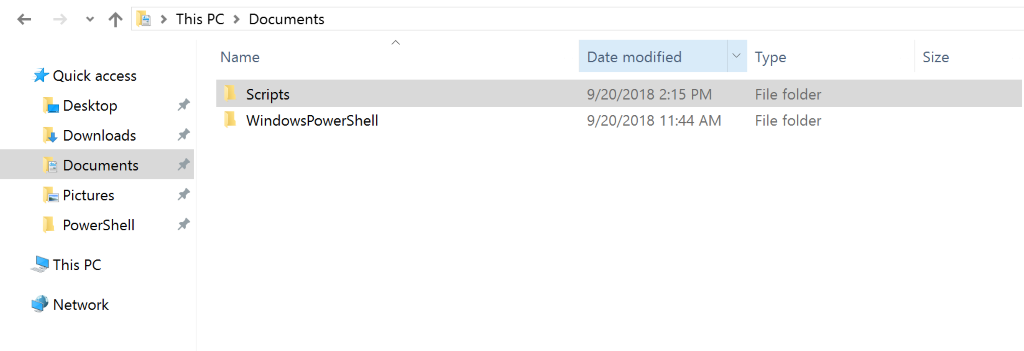
Browse the Ignite 2018 drive on your local device.



Navigate to *Scripts,* select and copy the folder:



Paste the folder into the *Documents* folder of your VM.



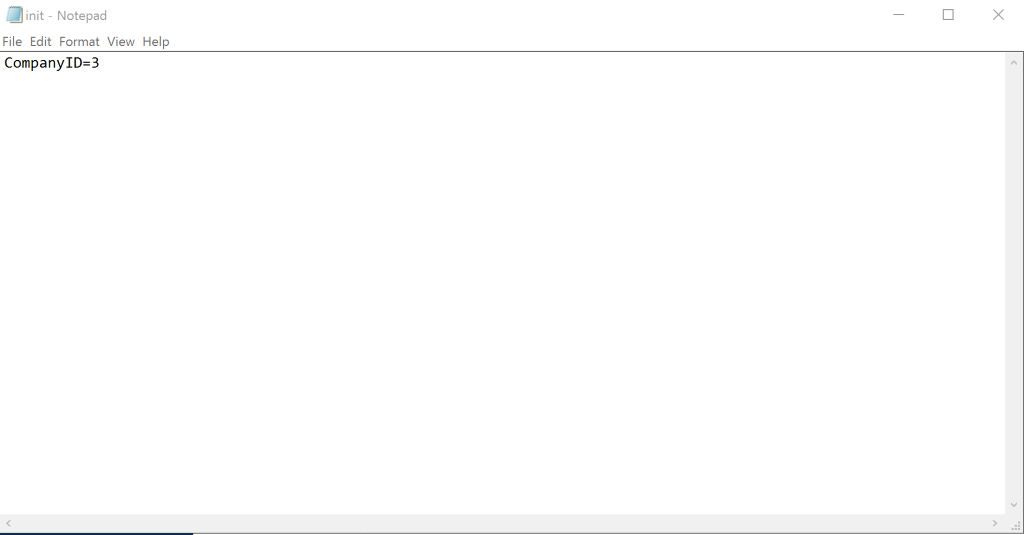
# Update the *Company ID* number in the *init* file

Open the *init* file and update the *Company ID* number to the number assigned to you in your index card. Save and close the file.

If you are deploying using PowerShell scripts, navigate to the *PowerShell* folder within the *Scripts* folder; if you are deploying using ARM Templates, navigate to the *ARMTemplate* folder within the *Scripts* folder. Once you are in the correct folder, open the *init* file, and update the *company ID*.

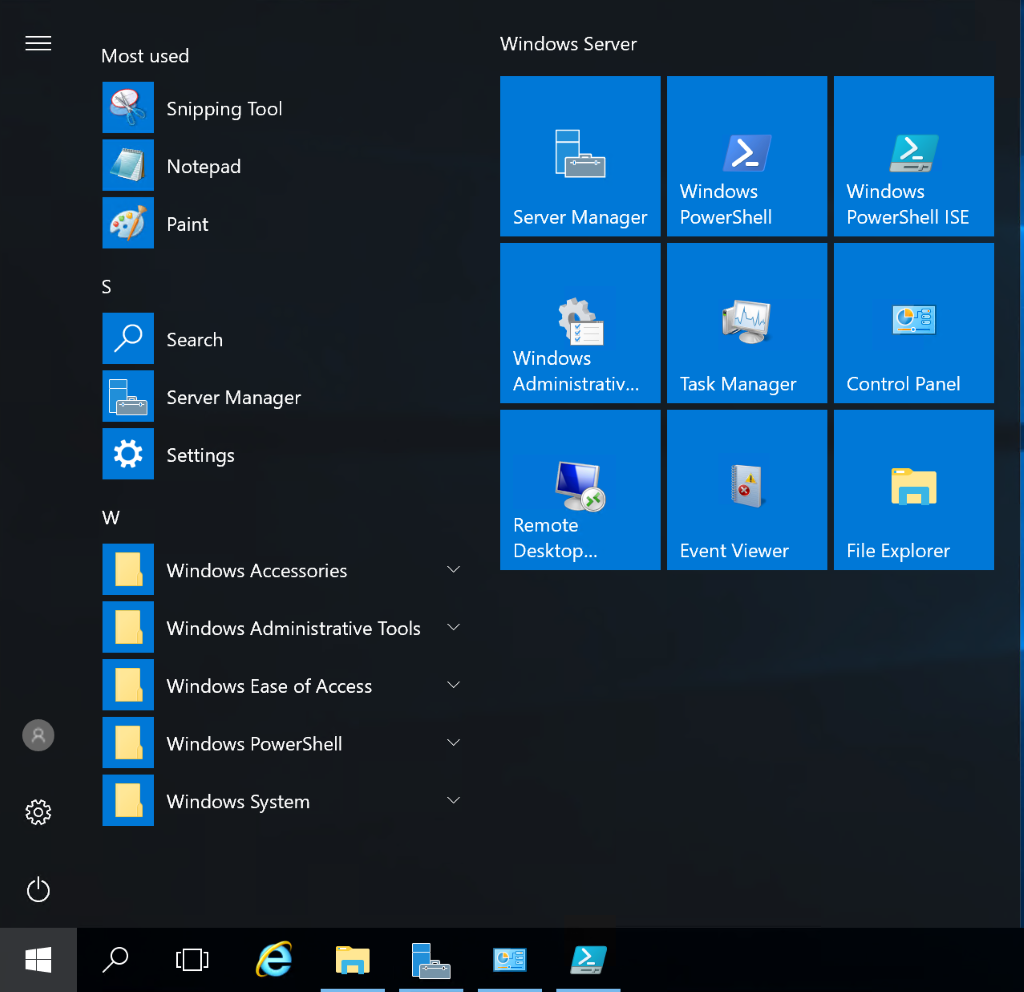
Tip: To avoid confusion you may want to update both the init files in the PowerShell and ARMTemplate folders

Example: Company03’s init file should read as follows:



# Open PowerShell ISE, and load the Scripts or Templates

Click on the task bar, and open *Windows PowerShell ISE*



## Deploying using PowerShell scripts

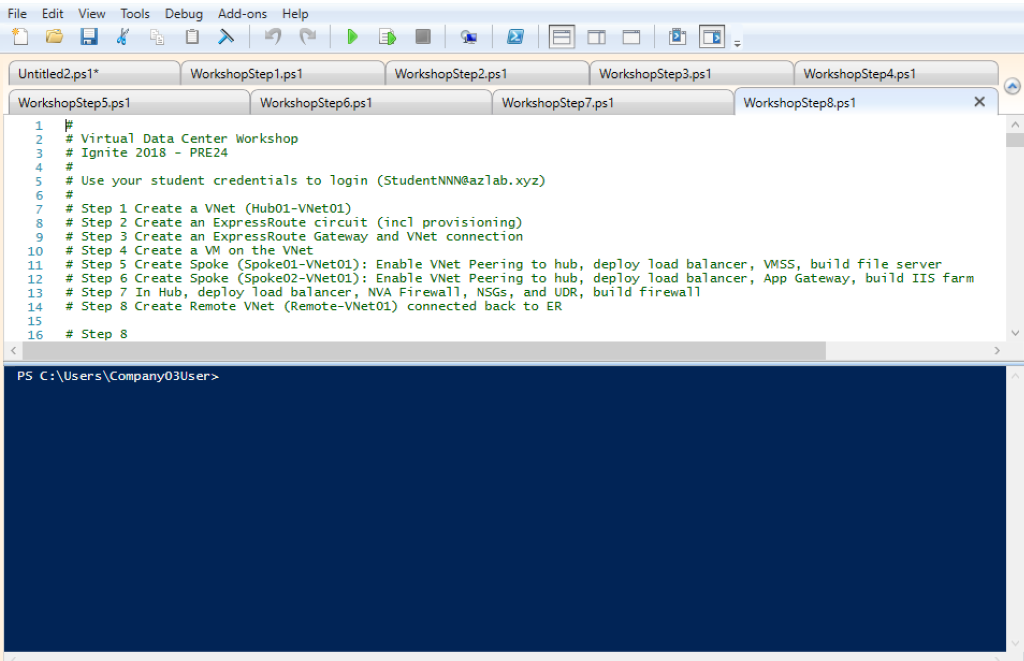
If you are deploying using the PowerShell scripts, browse *Documents > Scripts > PowerShell.*

Navigate to the *file* tab in the upper-left-hand corner and select *Open.* Browse to *Documents > Scripts > PowersShell*, and open *WorkStep1.*

The file will open a new tab in the ISE named *WorkshopStep1.ps1.*



Repeat this step for each *WorkshopStep* file, there are 8 files in total. Once all the files are loaded, your ISE should look as follows:

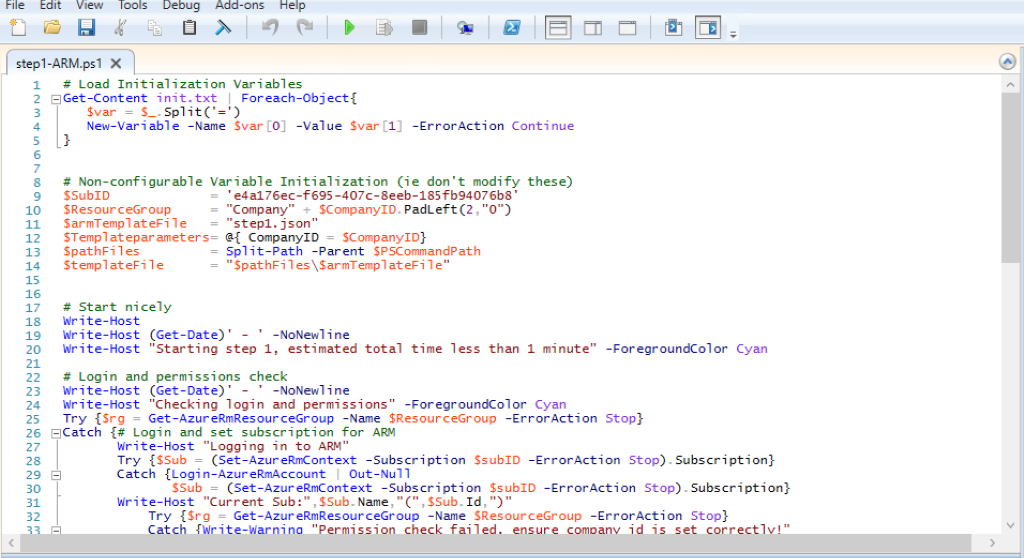


## Deploying using ARM Templates

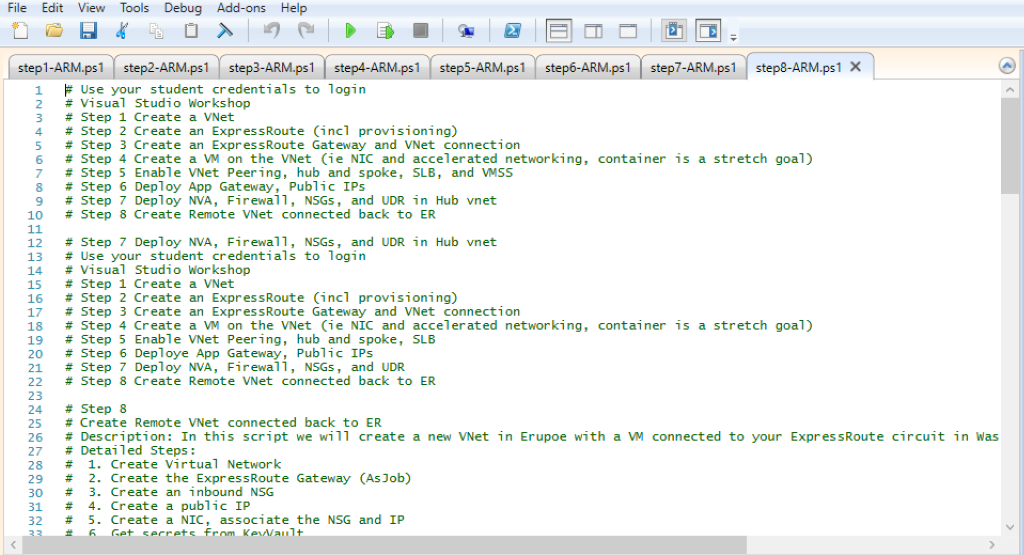
If you are deploying using ARM templates scripts, browse *Documents > Scripts > ARM Templates.*

Navigate to the *file* tab in the upper-left-hand corner and select *Open.* Browse to the *Documents > Scripts > ARMTemplate* and open *step1-ARM.*

The file will open a new tab in the ISE named *step1.json.*



Repeat this step for each ARM template, there are 8 files in total. Once all the files are loaded, your ISE should look as follows:



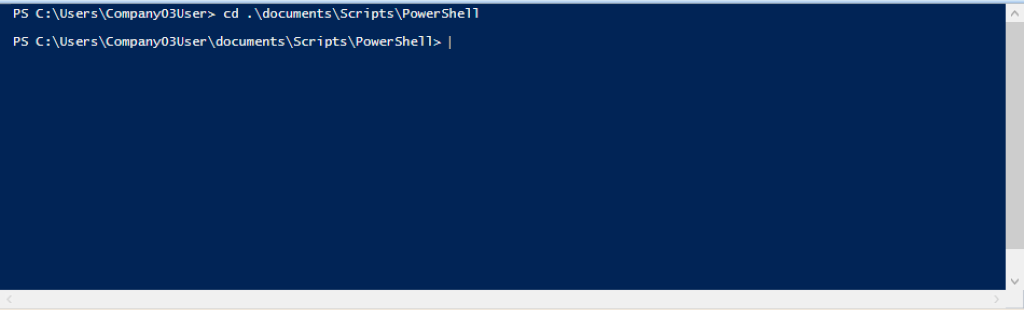
# Executing the Scripts or Templates

## PowerShellScripts

The PowerShellscripts need to be executed in order. Select the PowerShellscript that you want to run.

Within the script file, navigate to *Documents >Scripts > PowerShell*, by executing the following command:

**cd .\documents\Scripts\PowerShell**



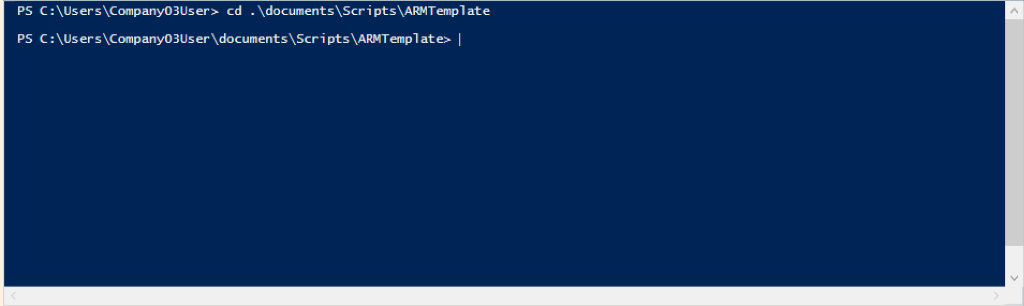
Once you are inside the correct folder, you can run the PowerShellscript. To do so, click C:\Users\rambala\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\2F9D4872.tmpon the top navigation bar.

## Using ARM Templates

The ARM templates need to be executed in order. Select the ARM template that you want to run.

Within the ARM template, navigate to *Documents >Scripts > ARMTemplate*, by running the following command:

**cd .\documents\Scripts\ARMTemplate**



Once you are inside the correct folder, you can run the *ARM*. To do so, click C:\Users\rambala\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\EAF1B3DE.tmpon the top navigation bar.

# Part II

Part II

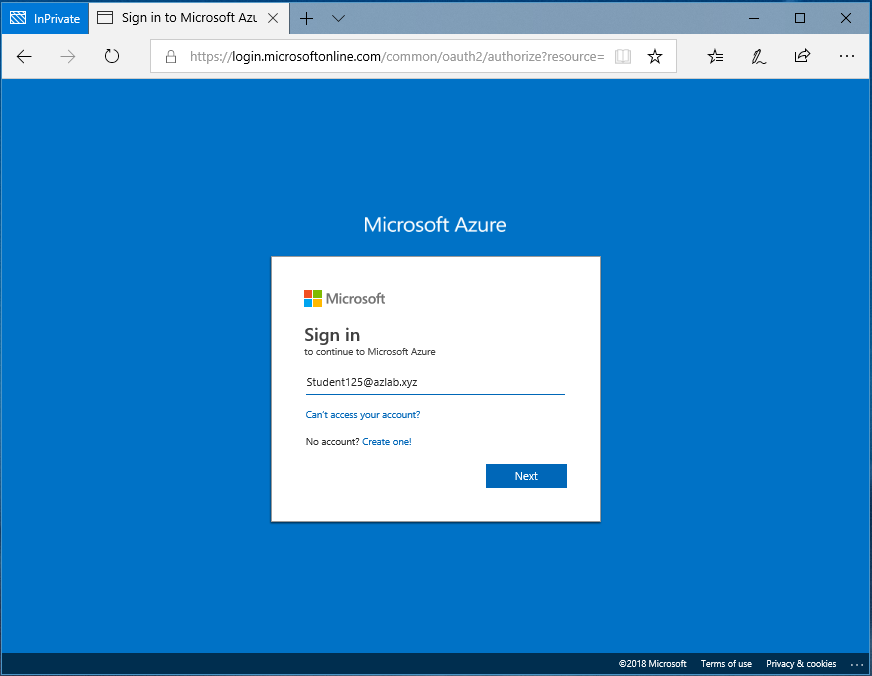
Azure Portal Guide

# Let’s Get Set!

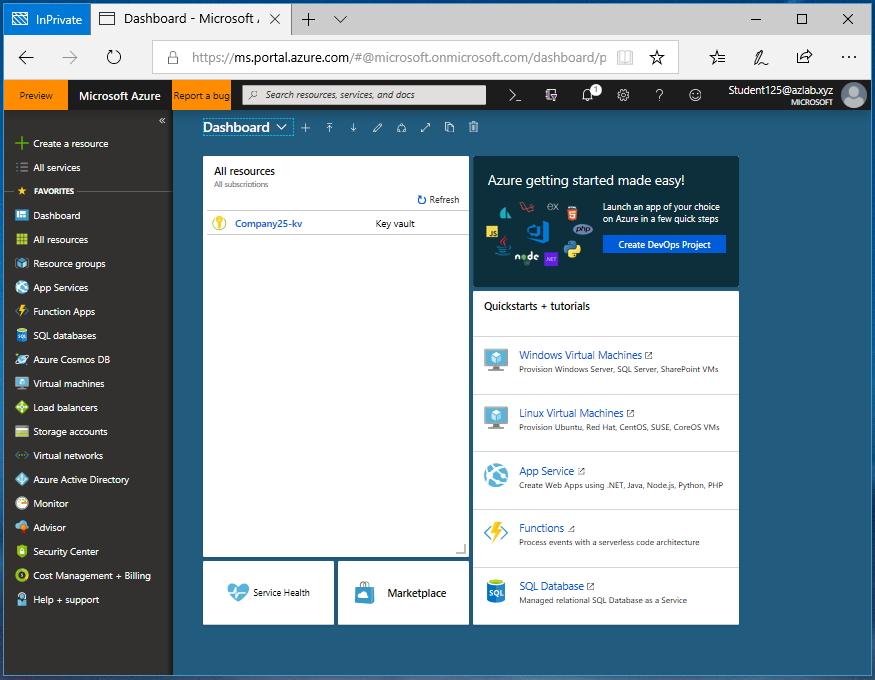
Normally to login to Azure portal, we go to portal.azure.com.

However, for this workshop there is an Azure Ignite2018 preview portal. Using your student username and password in the flash card provided to you, login http://aka.ms/[Ignite2018-AzurePreviewPortal](https://aka.ms/Ignite2018-AzurePreviewPortal) .

Note: Post Ignite2018, please use the regular Azure portal at http://portal.azure.com



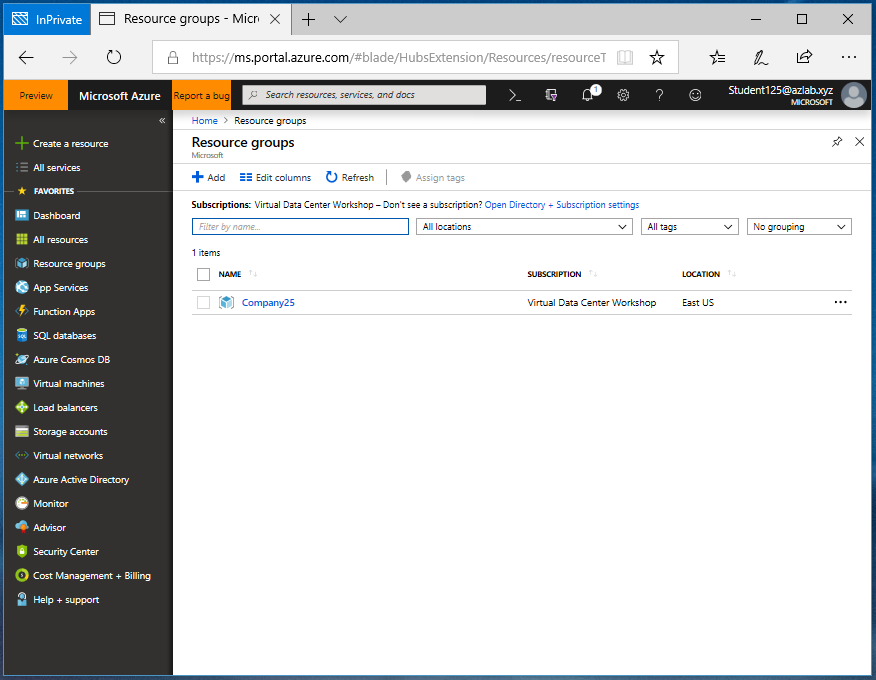
Following a successful login, you will land in the Azure portal homepage, showing the following dashboard:



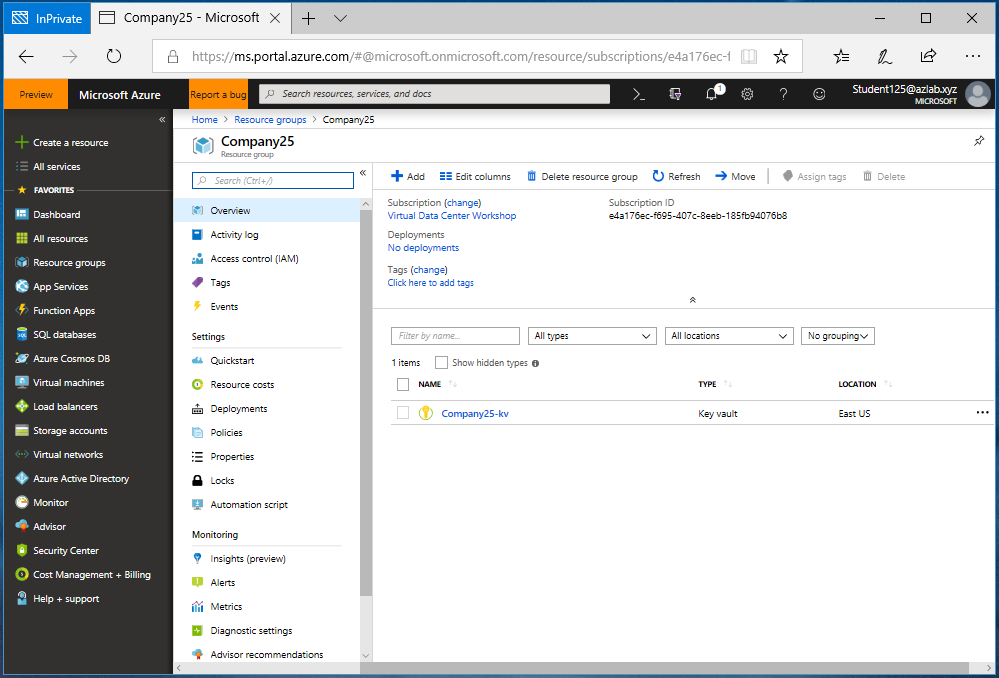
At this point, all you have is a Key vault resource under your company resource group that was pre-configured by the instructor.

## Azure Resource Group Blade:

You can see the resource group, by clicking  on the left navigation pane.



Clicking on a resource group name, will take you to the list of resources created within the resource group.



Each resource in Azure must belong to a resource group. A resource group is simply a logical construct that groups multiple resources together, so they can be managed as a single entity. For example, resources that share a similar lifecycle, such as the resources for an n-tier application may be created or deleted as a group.

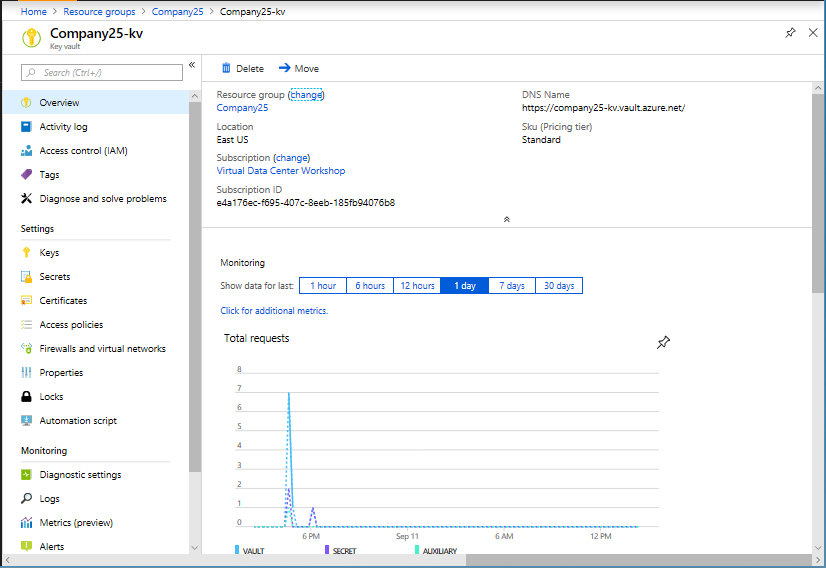
For this workshop, we will be creating all the resources under the previously created CompanyXX resource group. Therefore, you can use the CompanyXX resource group blade, shown above, as the base, and create all the resources from here. As you create new resources, they will be listed along with the key vault resource, in the overview of the *resource group* blade*.*

NOTE: The CompanyXX resource group blade illustrated in the preceding picture is your base, from which you will create the resources for this workshop. In the rest of this document this blade is referred to as ‘base blade’.

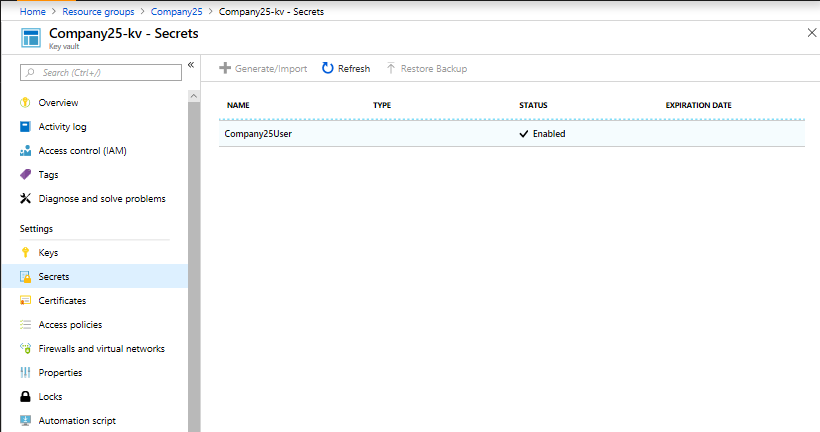
The pre-created key vault resource contains the username and password that you need to access the on-premises Virtual Machine (VM) assigned to you. You need to use this username and password for the admin accounts on the VMs you will be creating in Azure.

## Accessing Azure Key Vault

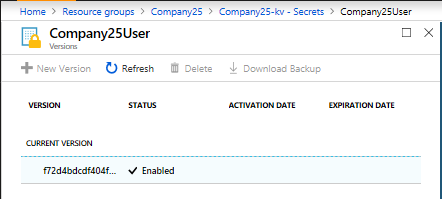
Click on the Key Vault resource (CompanyXX-kv) in your base blade.



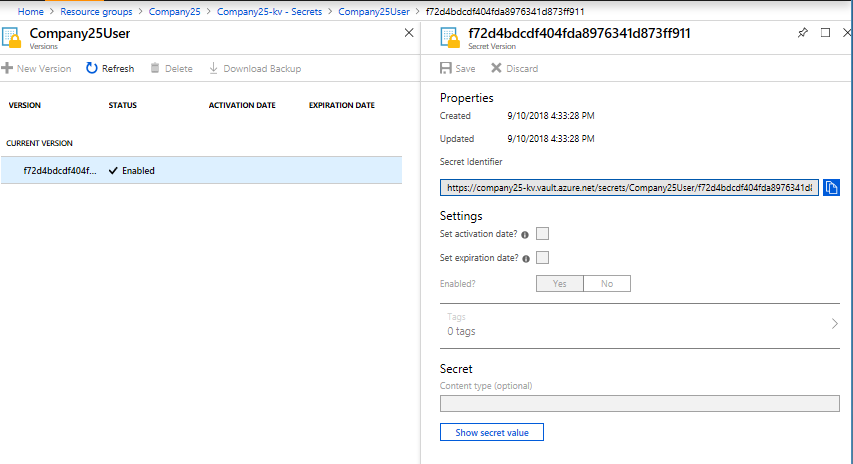
On the *key vault blade* that opens, click on the left navigation pane of the blade.



On the *key vault Secrets* blade that opens, click on your username (ex: Company25User).



Click on the current version of the secret.



On the extended blade, click on *Show secret value* and copy the secret value by clicking the copy icon that follows the value.

Tip: You may want to copy the secret value and the username to a notepad and keep it readily accessible by you (and only by you), for the rest of the training.

Note: Throughout this guide, if value is not specified for a configuration parameter, then use the default value. For example, in Step 1, Virtual network deployment model defaults to “Resource Manager” (in this workshop, we will be deploying all the resources in Resource Manager model)

# Step 1: Create a VNet

In this step, let’s create a VNet that will be used as a regional hub in your resource group, and carve out two additional subnets, within the VNet.

## Create a VNet

We need the following information to create a VNet:

VNet Name: **Hub01-VNet01**

Address space: **10.10.1***XX***.0/25** (ex: 10.10.125.0/25 for company 25)

Subscription (Name): **Virtual Data Center Workshop**

Resource group: **Company***XX* (ex: Company**25**)

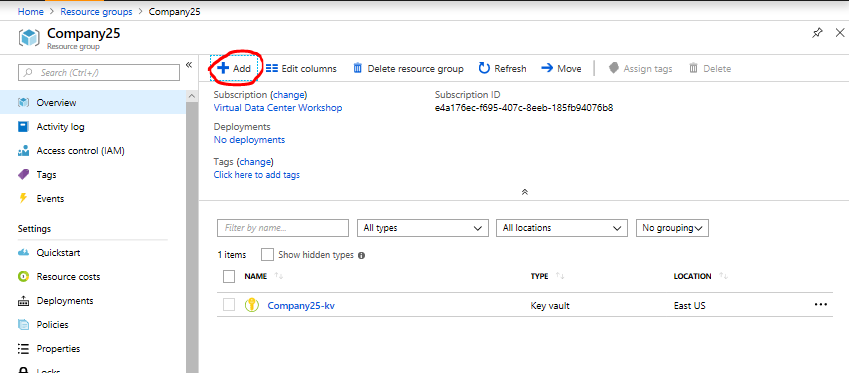
Location (Azure Region): **East US**

Subnet Name: **Tenant**

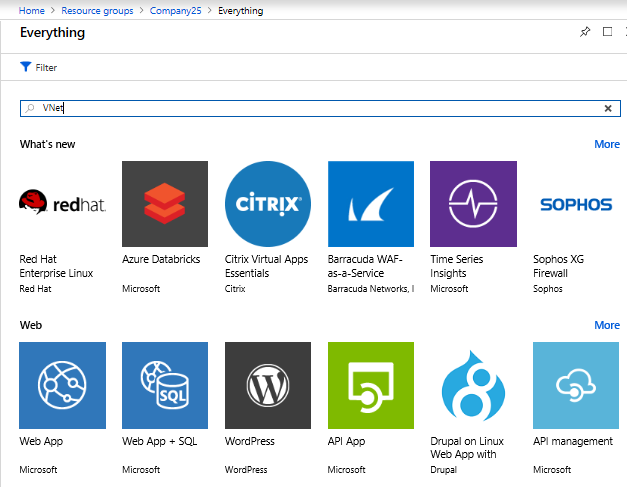
Subnet Address Space: **10.10.1***XX***.0/28** (ex: 10.10.125.0/28)

Now that we have all the information needed to create our first VNet, let’s proceed with the VNet creation process:

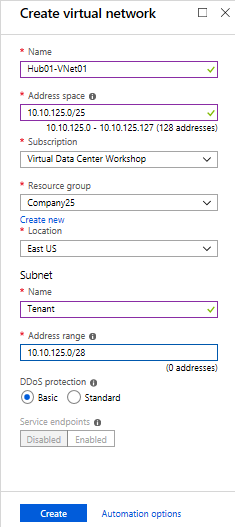
On your base blade, click  button at the top.



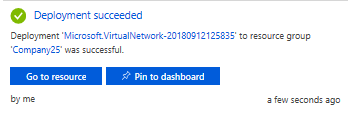
This will open the *Everything* blade. The *Everything* blade lists different Azure resources that you can create. Type “VNet” or “Virtual Network” on the search bar on the top and choose *Virtual Network*.



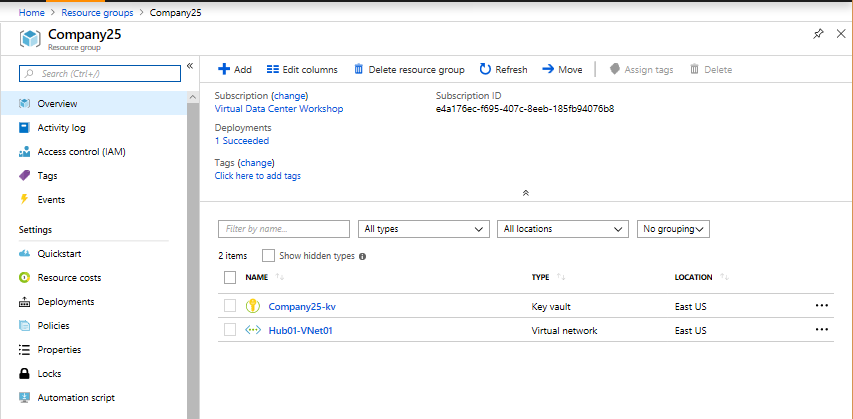
On the *Create virtual network* blade, enter the required information and click 



The VNet creation will take a few seconds. After successful completion of the creation process, you will receive a notification as shown below:

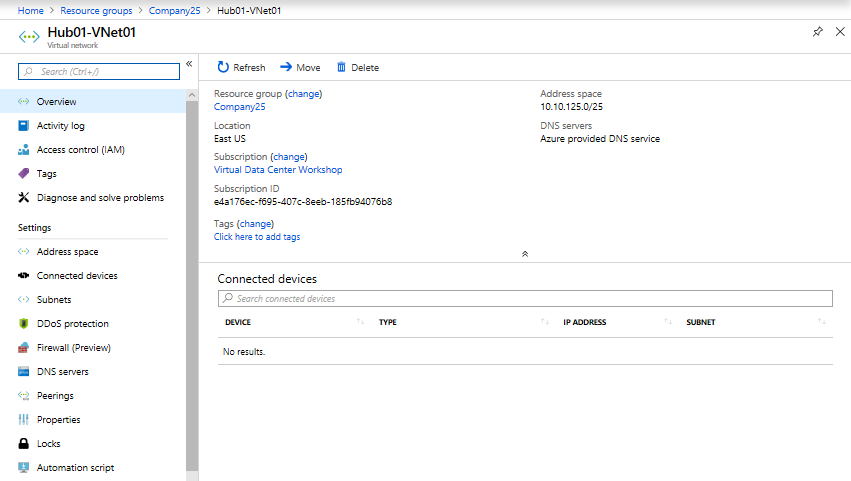


Now your base blade will display Hub01-VNet01, listed alongside the key vault resource.

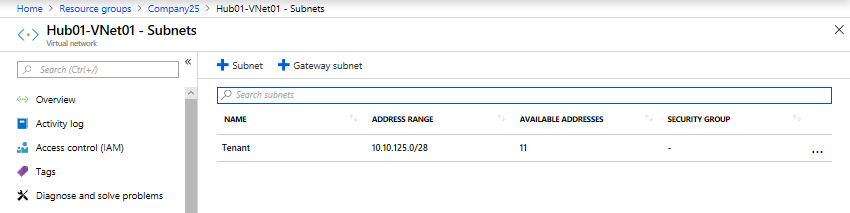


## Create Additional Subnets on the VNet

To create two additional subnets within the VNet, click on the name of the VNet (Hub01-VNet01) in your base blade. This will open the *VNet blade*.



On the VNet blade, click  on the left navigation pane, to open *the Subnets blade.*

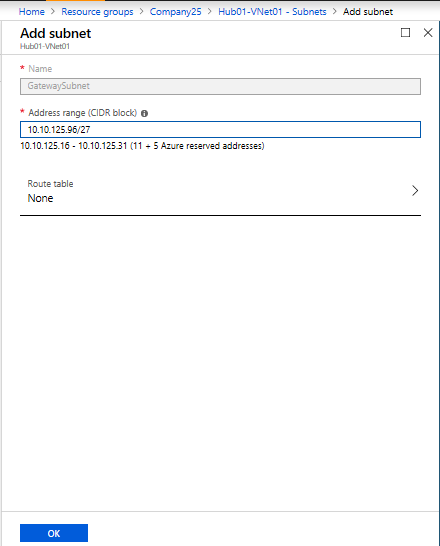


Let’s add a Gateway subnet first, with the following information:

Subnet Name: **GatewaySubnet** (default and you cannot change it).

Address Range: **10.10.1***XX***.96/27**

To add the Gateway subnet, click on the top of the blade.



After entering the address range, click  at the bottom of *the* blade. Once created, the new Gateway Subnet will be listed on the *Subnets blade* of the VNet.

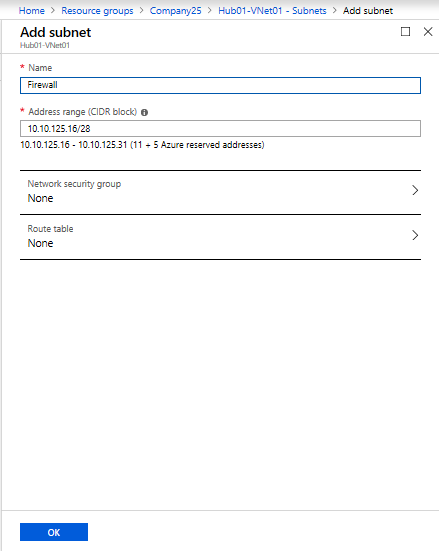


Now, let’s add a third subnet for deploying firewalls. The subnet information is as follows:

Name: Firewall

Address range: **10.10.1***XX***.16 /28** (ex: 10.10.125.16/28)

To add the Subnet, click on the top of the blade.

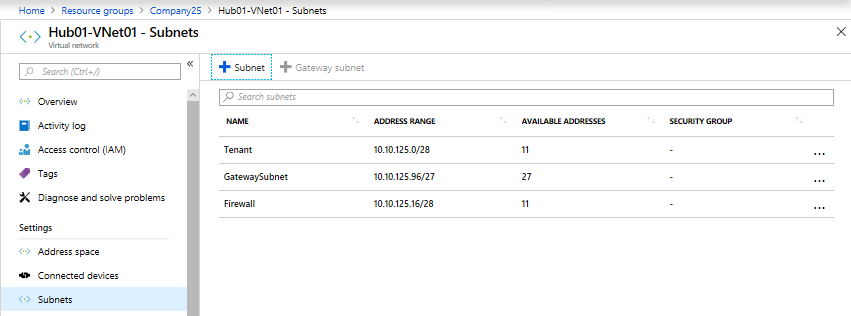


After entering the subnet name and the address range on the *Add subnet blade*, click .

Tip: The creation process running in the background is notified by the moving blue line underneath the bell (notification) icon on the top of the Azure portal.



Following successful creation, the new Subnet will be listed on the *Subnets* blade of the VNet.



With that, we have completed *Step 1* of the workshop.

# Step 2: Create an ExpressRoute Circuit

In this step, let’s create an ExpressRoute circuit in your resource group, using the following information:

Circuit Name: **Company***XX***-er** (ex: Company25-er)

Provider: **Equinix**

Peering location: **Washington DC**

Bandwidth: **50 Mbps**

SKU: **Premium**

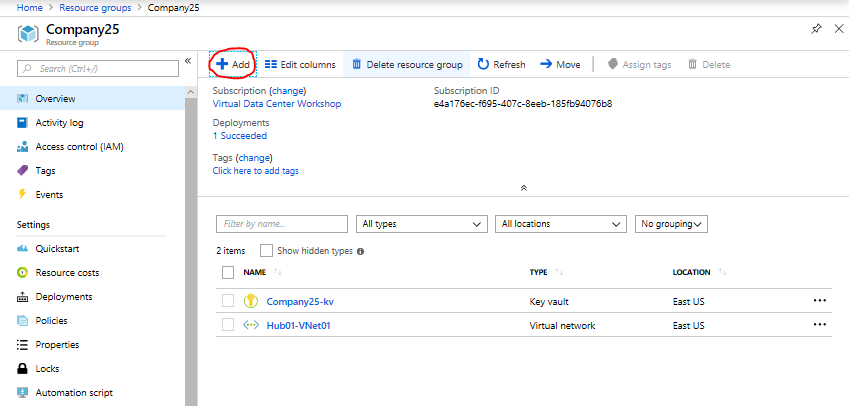
Billing model: **Metered**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX* (ex: Company25)

Location: **East US**

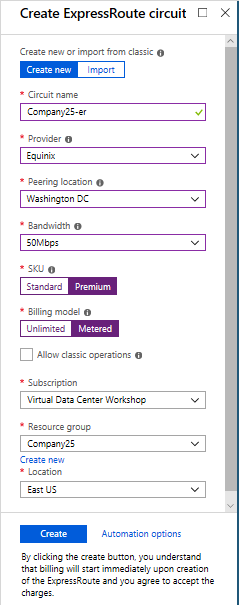
On your base blade, click  at the top.



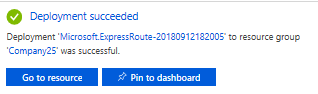
This will open a blade listing different resources that you can create. Type “ExpressRoute” on the search bar on the top and choose *ExpressRoute*. This will open the *ExpressRoute* introduction blade. Click  at the end of the blade.



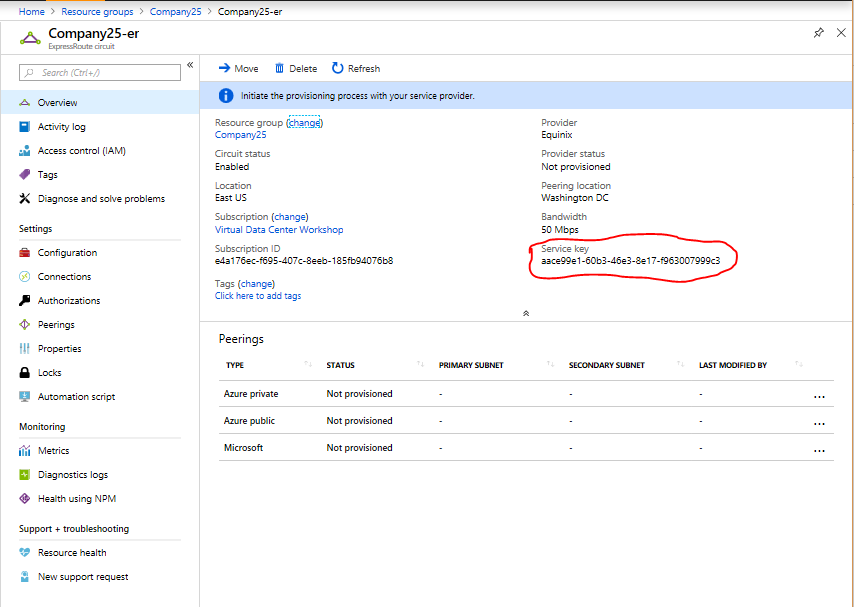
On the *Create ExpressRoute circuit* blade, enter the required information and click  at the bottom



Following the successful creation of the ExpressRoute circuit, you will receive the following message:

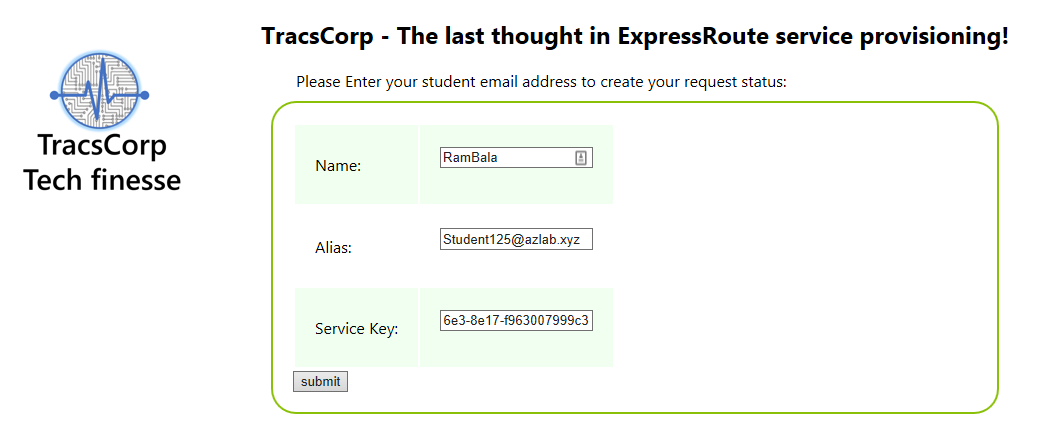


You can click  on the above notification or access the ExpressRoute circuit from your base blade.



Note that the *ExpressRoute circuit* blade indicates Circuit Status as *Enabled*, and Provider status as *Not provisioned*. This implies that the circuit is provisioned on the Microsoft end and the provisioning of the circuit on provider (ex: Equinix) end is pending.

To provision the circuit on the provider’s end, copy the service key (see the example circled above in the ExpressRoute circuit blade) of the ExpressRoute circuit, and submit it along with your name & your alias (username on the flashcard given to you at the beginning of the workshop) at the following website: http://aka.ms/WorkshopSP

With that, we have created all the resource involved in *Step 2.* You can check if your ExpressRoute circuit has been provisioned by the provider, half-way through *Step 3*.

# Step 3: Create an ExpressRoute Gateway and VNet Connection

In this step, let’s create an ExpressRoute gateway in your VNet (from *Step 1*) and connect to the ExpressRoute circuit (from *Step 2*).

## Create an ExpressRoute Gateway

Information we need to create the ExpressRoute gateway:

Name: **Hub01-VNet01-gw**

Gateway type: **ExpressRoute**

SKU: **Standard**

Virtual network: **Hub01-VNet01**

Public IP address: **Create new**

**Hub01-VNet01-gw-ip** (name of the public IP address)

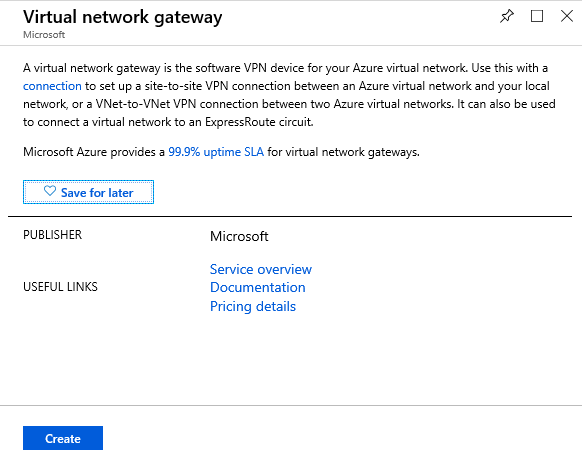
Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX* (ex: Company25)

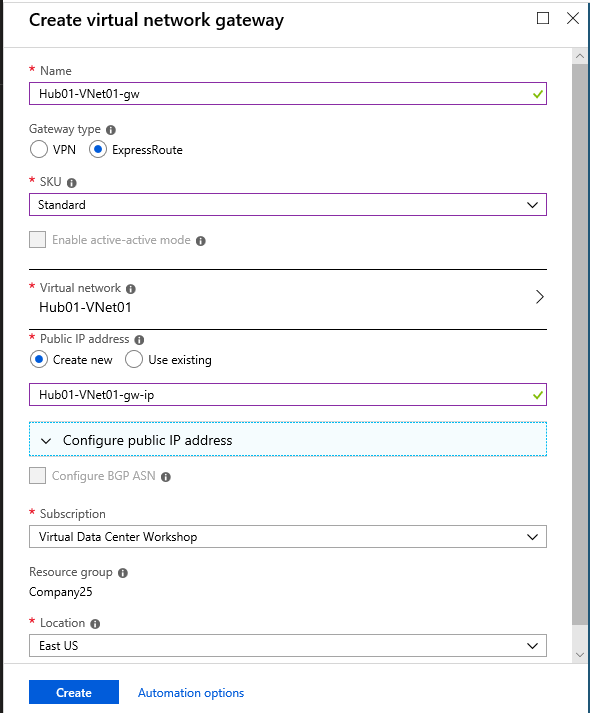
Location: **East US**

As we did in *Step 1* and *2*, on your base blade, click  at the top.

This will open the *Everything* blade, listing different resources that you can create. Start typing “virtual” on the search bar on the top and choose *Virtual network gateway*. This will open the *Virtual network gateway* introduction blade. Click  at the end of the blade.

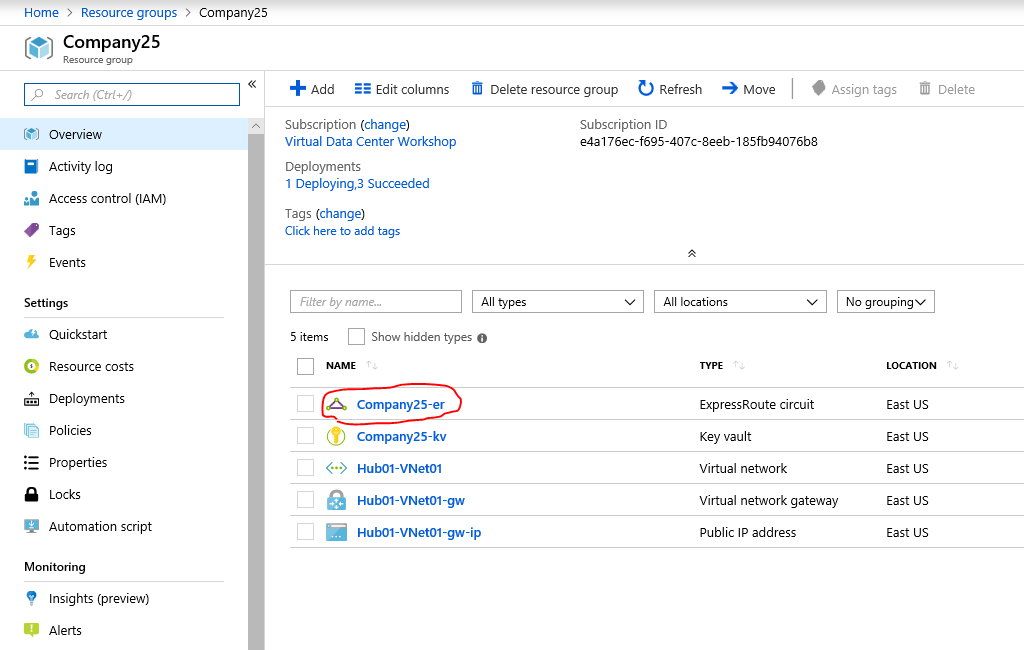


On the *Create virtual network gateway* blade, enter the required information and click  . Creation of the virtual network gateway (VNet GW) will take 10 to 15 mins.

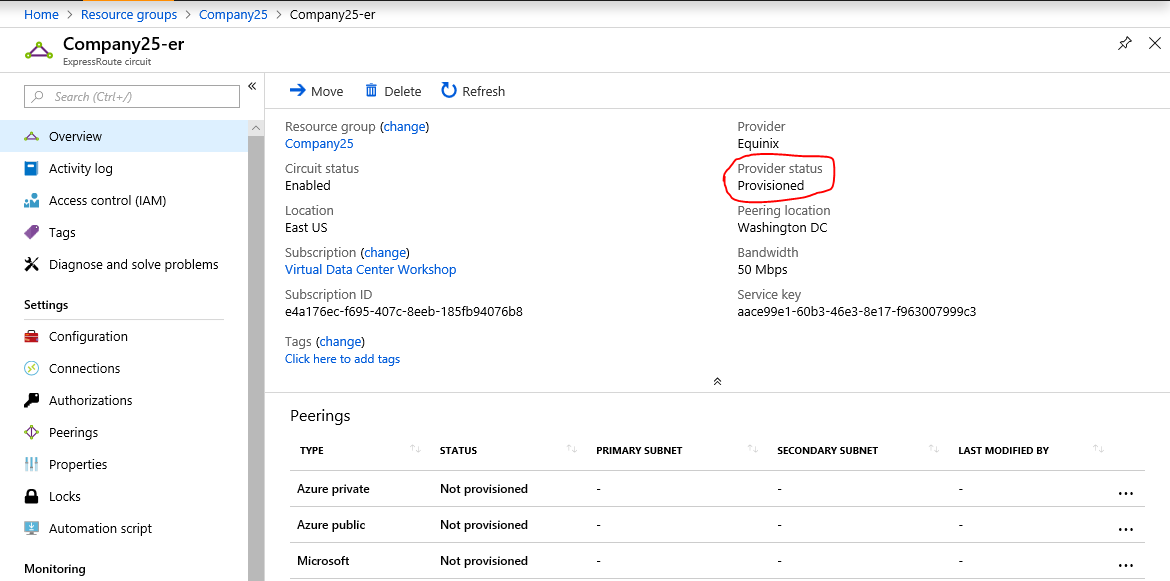


## Is the ExpressRoute circuit, created in *Step 2*, provisioned?

On your base blade, click on the name of the ExpressRoute circuit.



On the *ExpressRoute circuit* blade, check if the Provider status has changed to *Provisioned*.



If the circuit is provisioned, proceed with the rest of *Step 3*. If not provisioned, check with your instructor/proctor. Alternatively, if not provisioned, proceed with *Step 4*. After completing *Step 4* you can return and complete the rest of the *Step 3*.

## Create a Private Peering over ExpressRoute Circuit

Information we need to create the private peering:

Peer ASN: **65021**

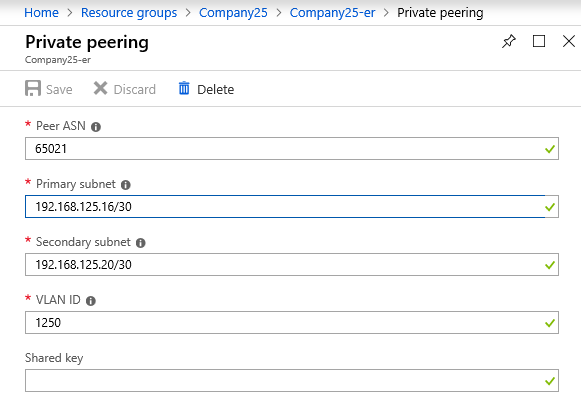
Primary subnet: **192.168.1***XX***.16/30** (xx is the company id)

Secondary subnet: **192.168.1***XX***.20/30** (xx is the company id)

VLAN ID: **1***XX***0** (ex. 1030 for Company03 and 1250 for Company25)

Shared key: (leave it blank)

On the *ExpressRoute circuit* blade, click *Azure private* under *Peerings*. This will open the *Private peering* creation blade. Enter the required information and click  at the top of the blade.



## Create a connection object

Let’s create a connection object to connect the VNet GW earlier in this step and the ExpressRoute circuit. Information we need to create the connection:

Name: **Hub01-VNet01-gw-conn**

Connection type: **ExpressRoute**

Virtual network gateway: **Hub01-VNet01-gw**

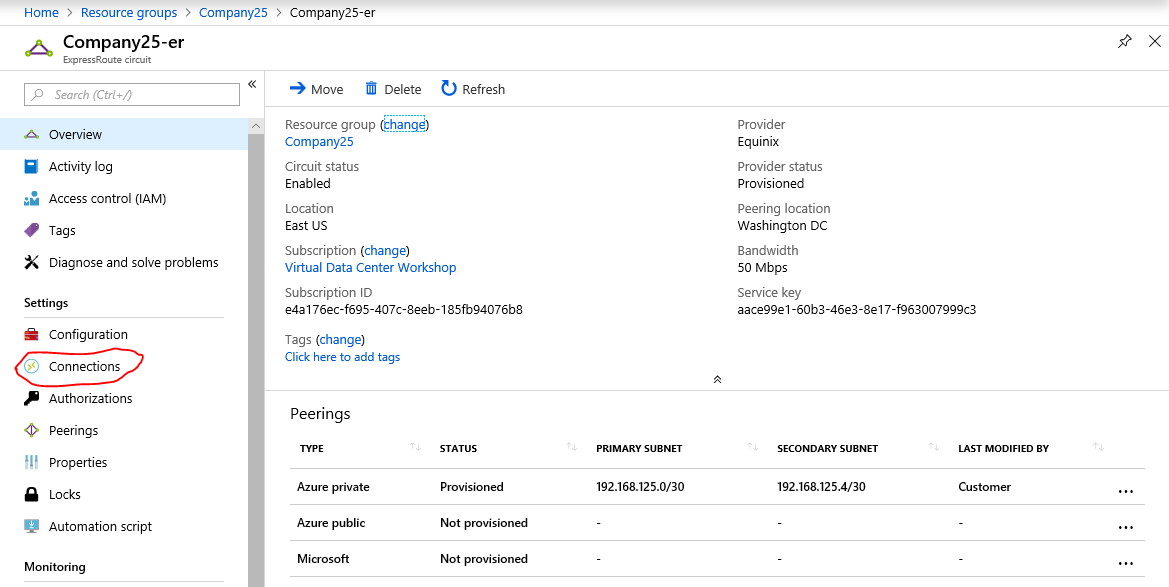
ExpressRoute circuit: **Company***XX***-er**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX*

Location: **East US**

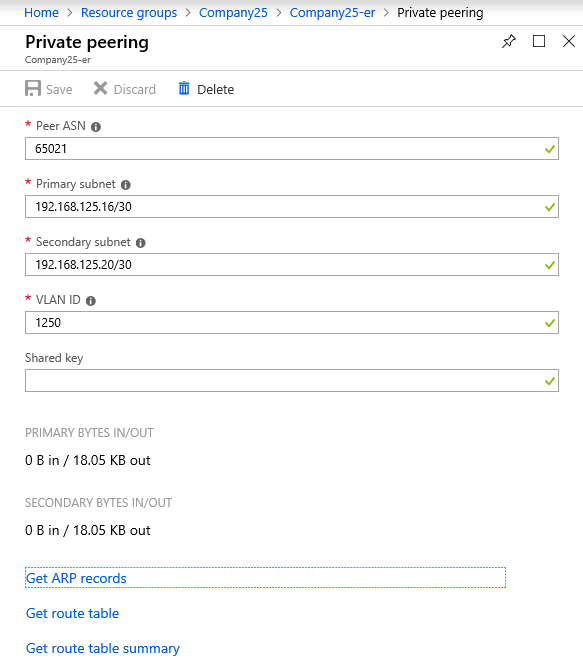
On your *ExpresRoute circuit* blade, click  on the left navigation pane.



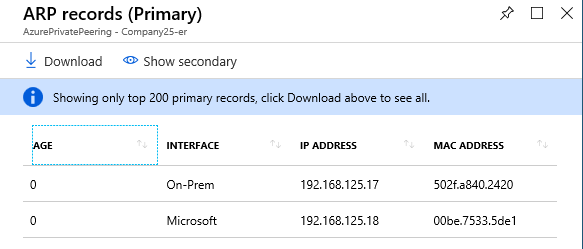
This will open the *Connections* panel. At the top of the panel, click  to open the connection blade. Enter the required information and click  at the bottom of the blade. Following successful creation of the connection, you will receive a notification.

## Validation

On the *ExpressRoute circuit* blade, click on Azure private peering. This will open the *Private peering* blade.



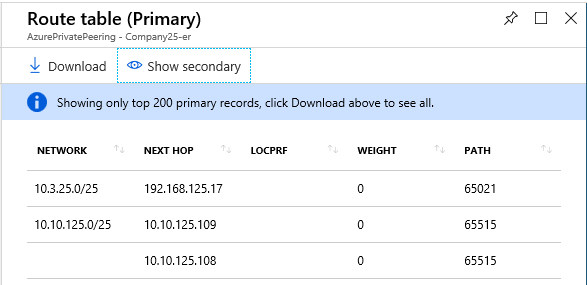
Towards the bottom of the blade, click on *Get ARP records*. This will open the *ARP records* *(Primary)* blade.



On the ARP table, as shown above you should see two entries—one corresponding to the On-Prem side router and the other corresponding to the Microsoft edge router. If either the on-premises side routers are not configured properly or if the peering information (Peer ASN, Primary/Secondary subnet, VLAN ID, or Shared key) mismatches, you will not see the entry corresponding to on-prem.

On the ARP records blade, click and validate the secondary side connectivity ARP table.

Next, on the *Private peering* blade, towards the bottom of the blade click on *Get route table*. This will open the *Route table (Primary)* blade.



The routes with path 65515 come from your VNet, and the ones with path 65021 come from on-premises. On the *Route table* blade, click and validate the secondary side route table.

With that, we have completed Step *3* of the workshop.

# Step 4: Create a VM on the VNet

In this step, let’s create a Network Security Group (NSG) and a public IP address, and associate it with a VM.

Note: A VM with a public IP creates a public endpoint and to mitigate public exposure risks, it is essential to associate a Network Security Group (NSG).

## Create a Network Security Group

Information we need to create an NSG.

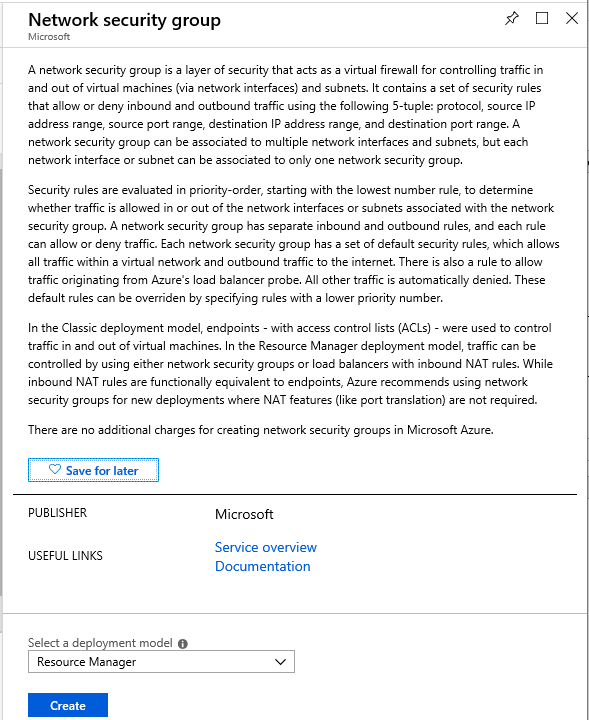
Name: **Hub01-VM01-nic-nsg**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Location: **East US**

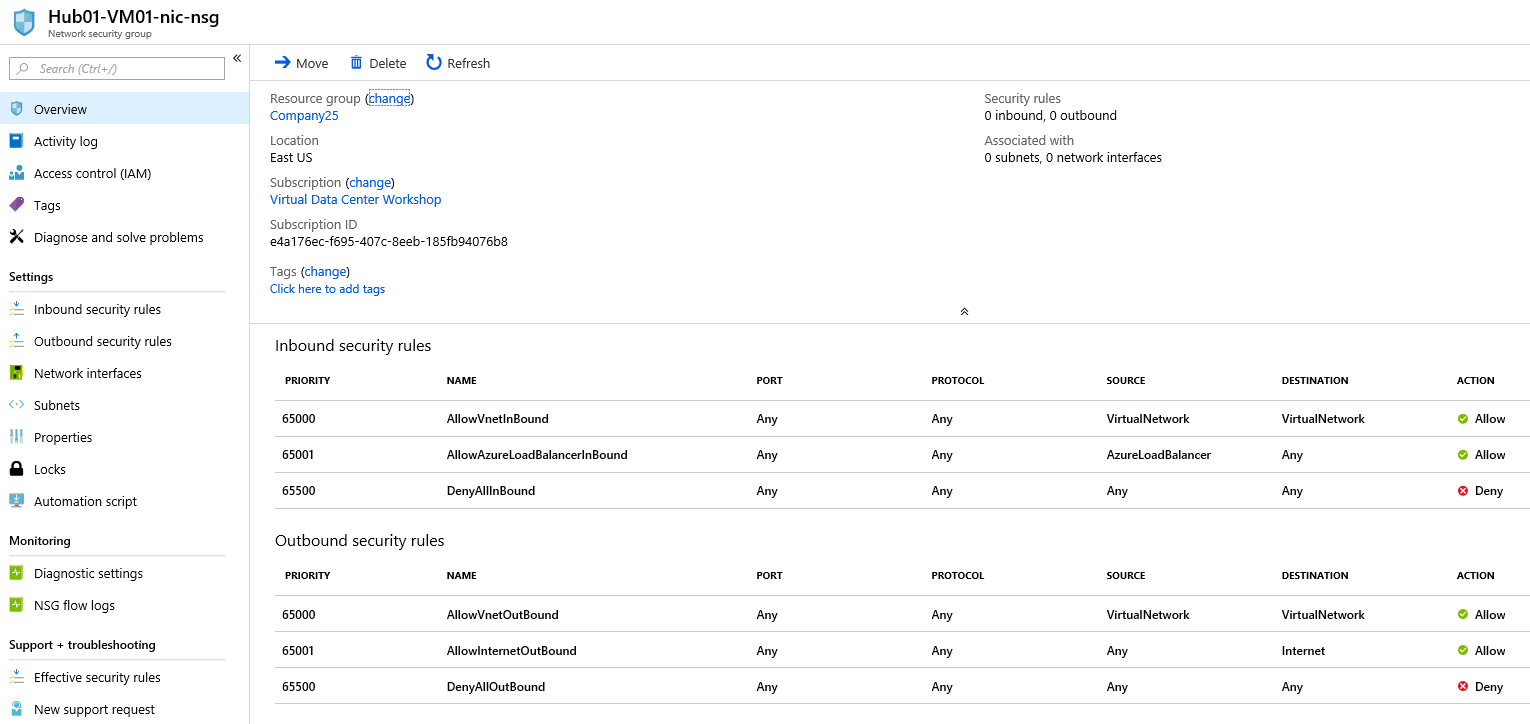
As we did in the previous steps, on your base blade, click at the top. This will open the *Everything* blade. Start typing “network security group” in the search bar on the top and choose *Network security group*. This will open the *Network security group blade*. Click  at the end of the blade.



On the *Network security group* blade, enter the required information and click . Following, a quick validation, the NSG will be created.



On your base blade click on the name of the newly created NSG: *Hub01-VM01-nic-nsg.*



The default setting of the NSG allows Intra-VNet communication and communication between the associated NIC and Azure Load Balancer. All other communication is blocked. Let’s add an inbound security rule to permit Remote Desktop Protocol (RDP) traffic.

Information needed to create an inbound security rule to allow RDP traffic:

Source: **Any**

Source port ranges: **\***

Destination: **Any**

Destination port ranges: **3389**

Protocol: **TCP**

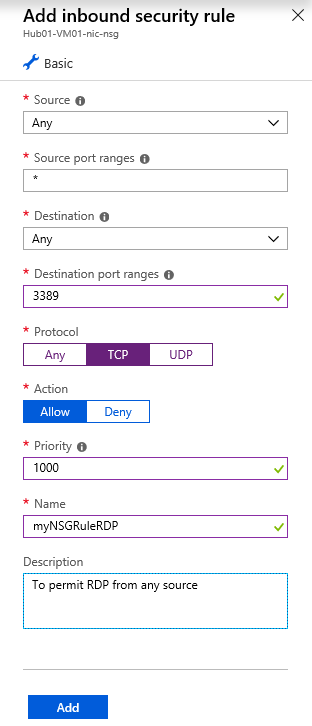
Action: **Allow**

Priority: **1000**

Name: **myNSGRuleRDP**

Description: To permit RDP from any source

On the *Network security group* blade, click  on the left navigation pane. Click at the top of the Inbound security rules pane. In the *Add inbound security rule* blade, enter the required information and click  at the bottom of the blade.



Note: For the sake of learning, we are exposing the RDP port of the VM you are creating to the Internet. However, this is a security risk in the production environments. Therefore, while creating the NSG rule, you will see the following warning from Azure:



With the successful addition of the security rule, you will see the rule listed along with the default inbound security rules of the NSG.

## Create a Public IP Address

Information we need to create a public IP address.

Name: **Hub01-VM01-nic-pip**

SKU: **Basic**

IP Version: **IPv4**

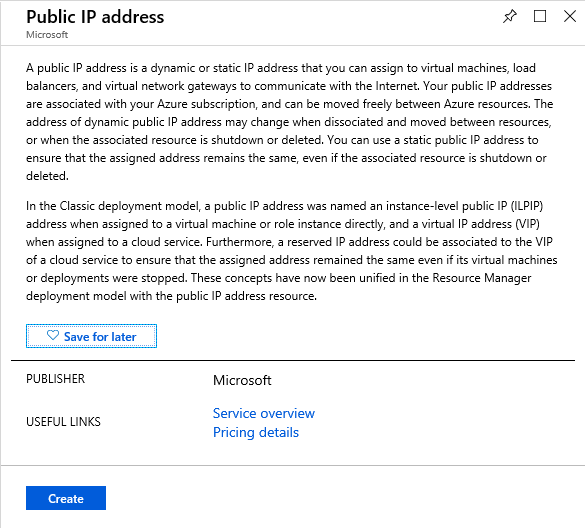
IP address assignment: **Dynamic**

Subscription: **Virtual Data Center Workshop**

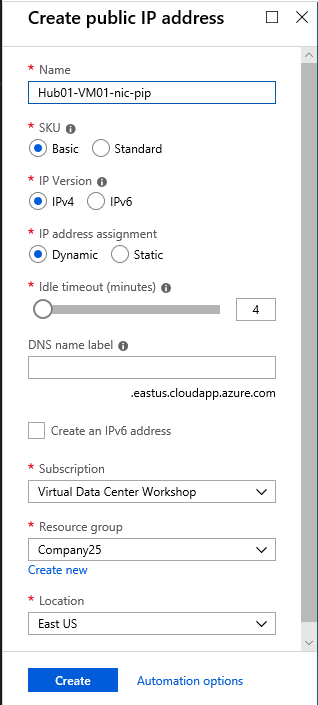
Resource Group: **Company***XX*

Location: **East US**

On your base blade, click  at the top. On the *Everything* blade, type “public IP” on the search bar at the top and choose *Public IP address*. This will open the *Public IP address* introduction blade. Click  button at the end of the blade.



On the *Create public IP address* blade, enter the required information and click . Following a quick validation, the public IP address resource will be created.



Note: Azure will only assign a public IP address to the network interface when the interface is attached to a VM.

## Create a Virtual Machine

Let’s create a virtual machine and associate the public IP address and NSG that we just created. Information we need to create the virtual machine:

Basics

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Virtual machine name: **Hub01-VM01**

Region: **East US**

Image: **Windows Server 2016 Datacenter**

Size: **Standard A4 v2**

Tip: to see Standard A4 v2, you may have to click the *Clear all filter* in Select *VM size* pane.

Username: **Student***XXX*

Password: (if not noted, see “Accessing Azure Key Vault”, at the beginning of this document).

Networking

Virtual network: **Hub01-VNet01**

Subnet: **Tenant**

Public IP: **Hub01-VM01-nic-pip**

Network security group: **Advanced**

Configure network security group: **Hub01-VM01-nic-nsg**

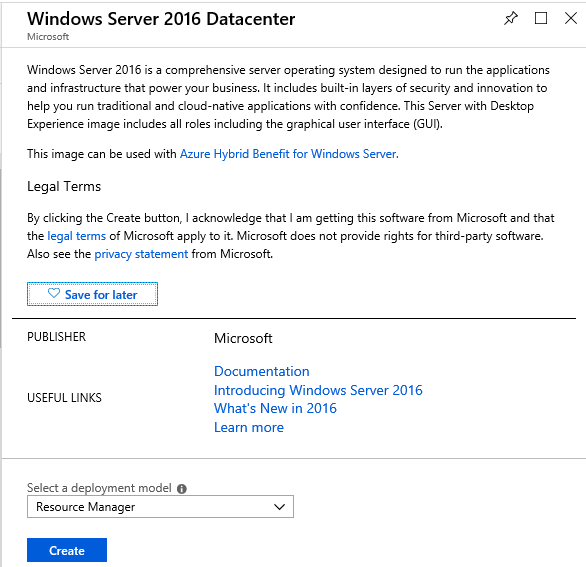
Management

Boot diagnostics: **Off**

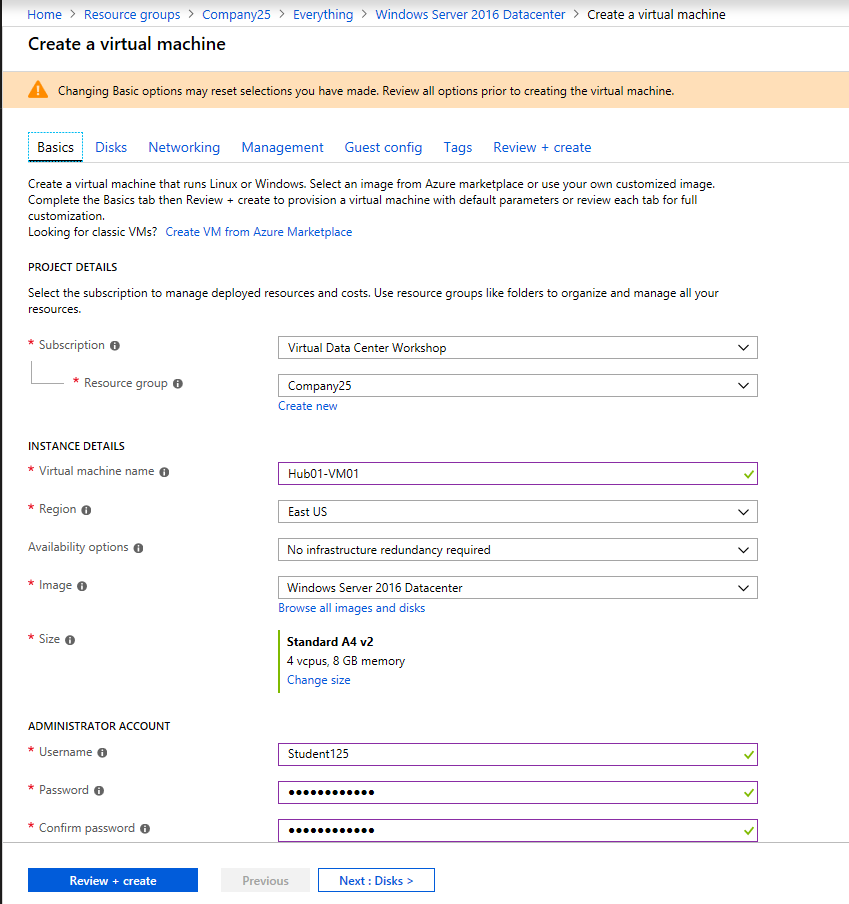
Guest config

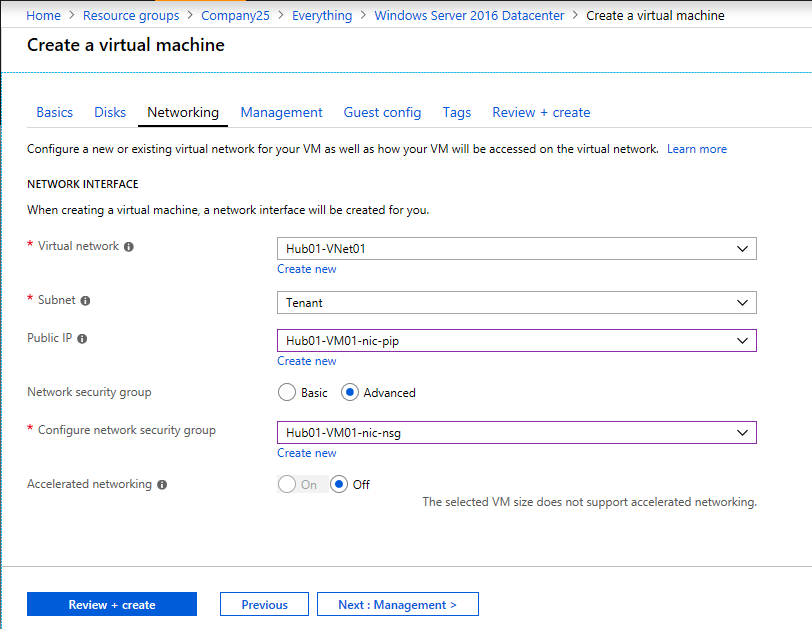
We are going to use *Custom Script Extension* to allow ICMP traffic on the VM that we are creating (Alternatively, after the VM is created you can login to the VM and edit *Inbound Rules of Windows Firewall* with Advanced Security to permit the ICMP traffic).

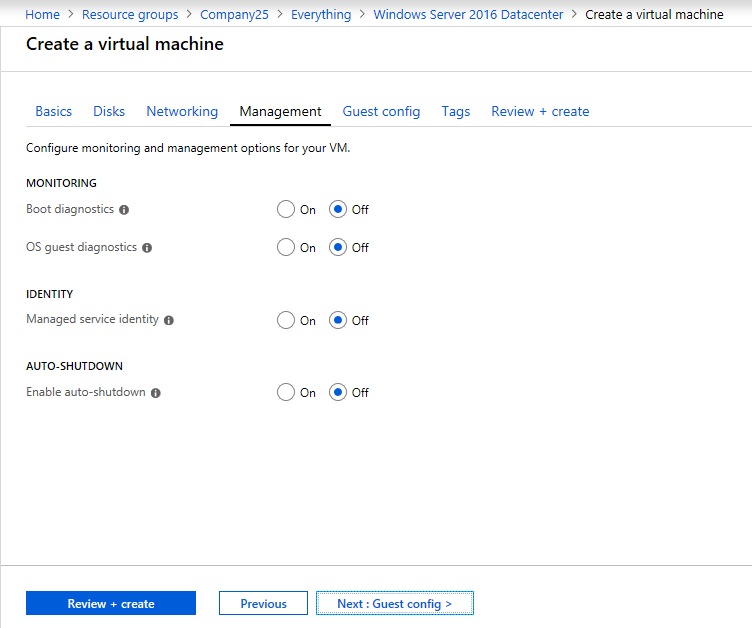
On your base blade, click at the top. On the *Everything* blade, type “virtual machine” on the search bar at the top and choose *Windows Server 2016 Datacenter*. This will open the *Windows Server 2016 Datacenter* introduction blade. Click  at the end of the blade.



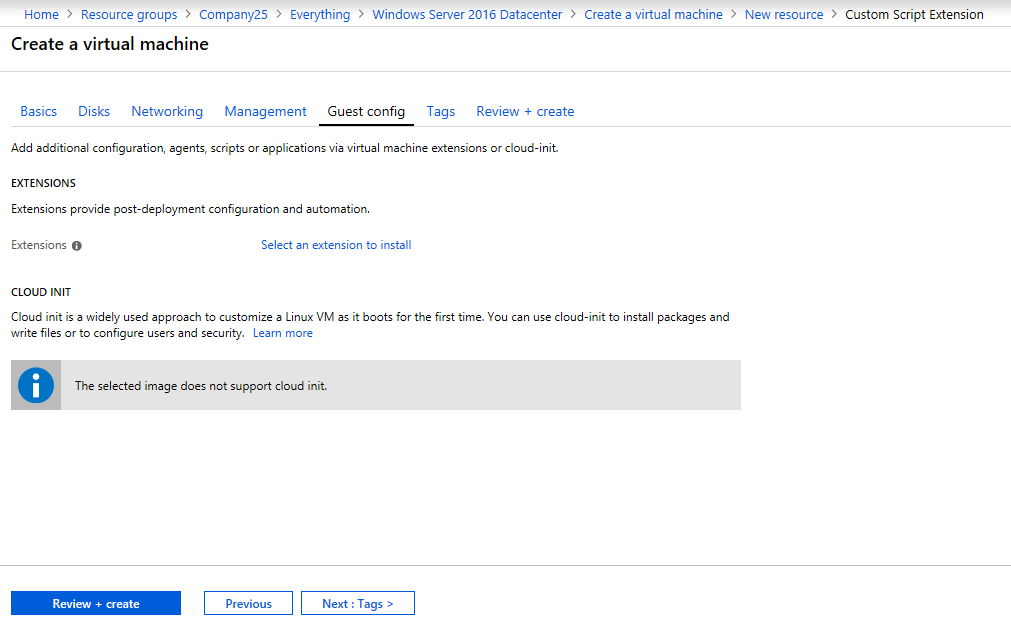
Within the *Create a virtual* blade, the required information is categorized under different tabs. Under *Basics*, *Networking* and *Management* tabs enter the required information. The next three screenshots show an example of values entered for these three tabs.



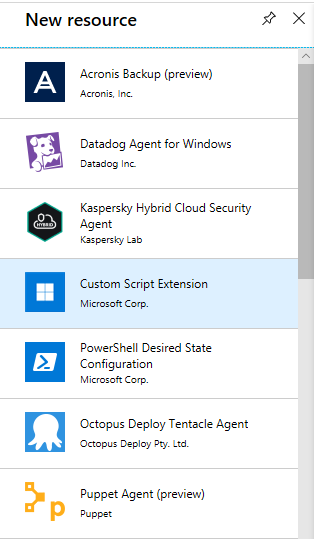




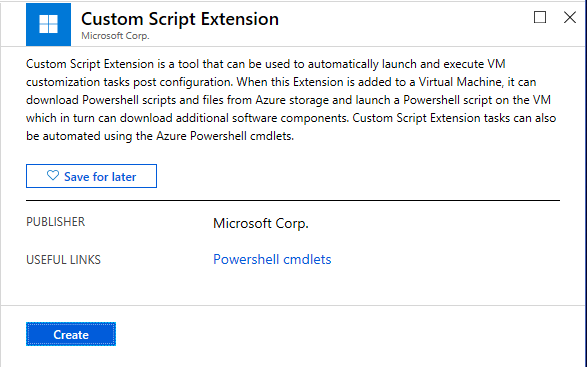
Under the *Guest config* tab, click on *Select an extension to install*.



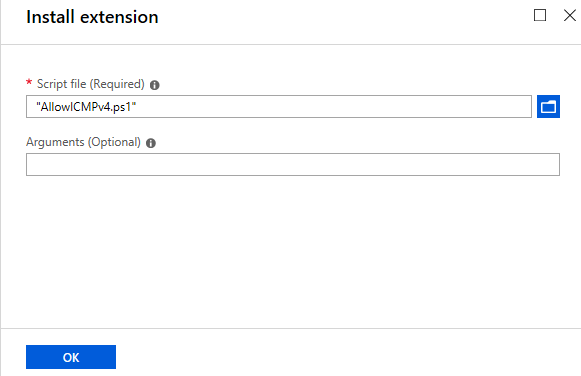
This will open the *New resource* pane:



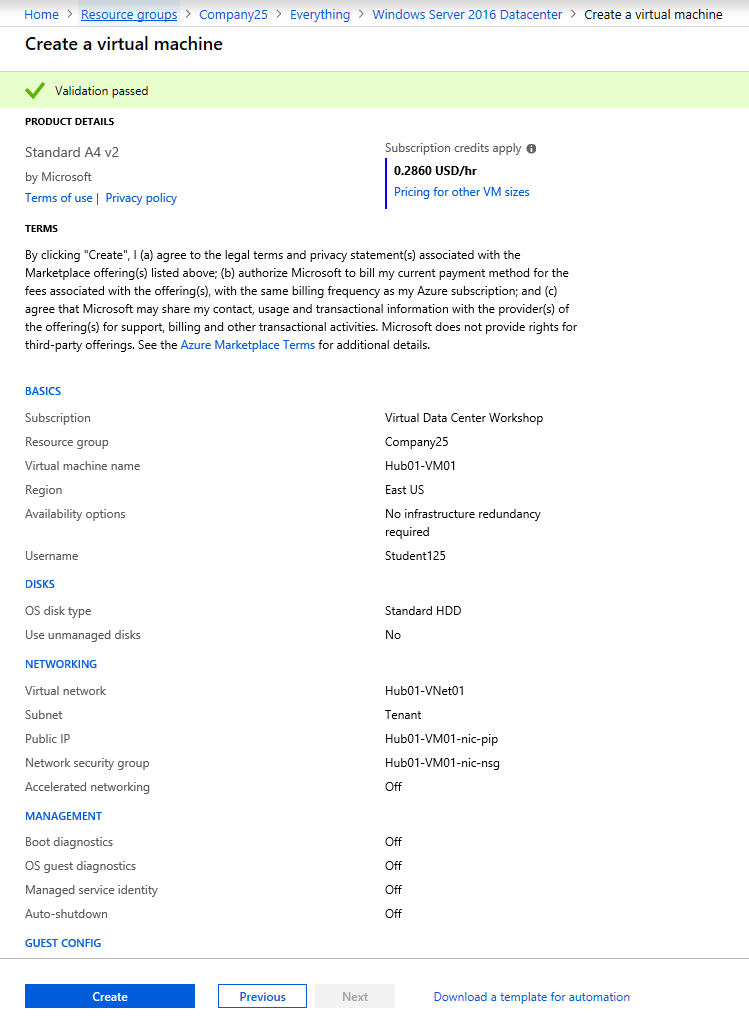
Click on *Custom Script Extension.* This will open the *Custom Script Extension* introduction blade. Click  at the end of the blade.



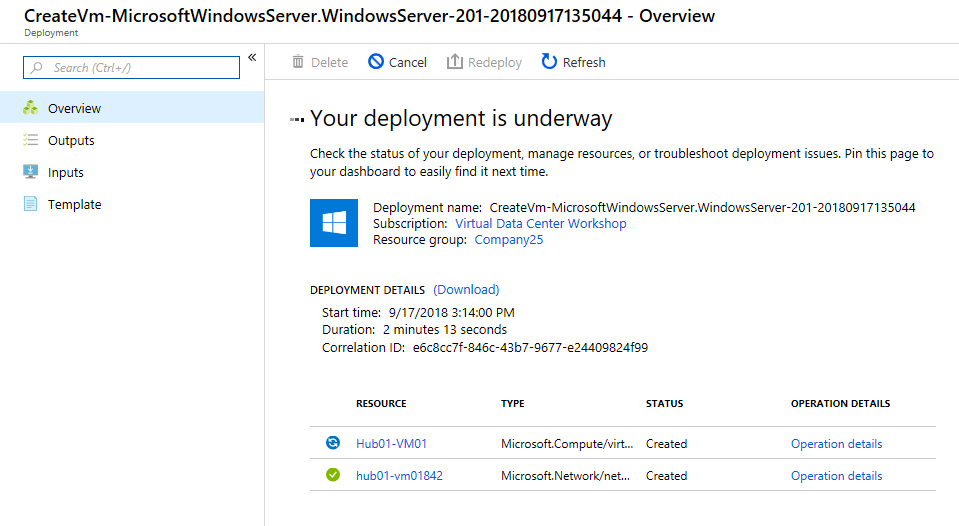
This will open the *Install extension* blade. Choose AllowICMPv4.ps1 file in the Microsoft Azure flash drive given to you at the beginning of the workshop and click  at the bottom of the blade.



On the Create a virtual machine blade, click . Review and ensure that the values are correctly entered. Click  at the bottom of the blade.

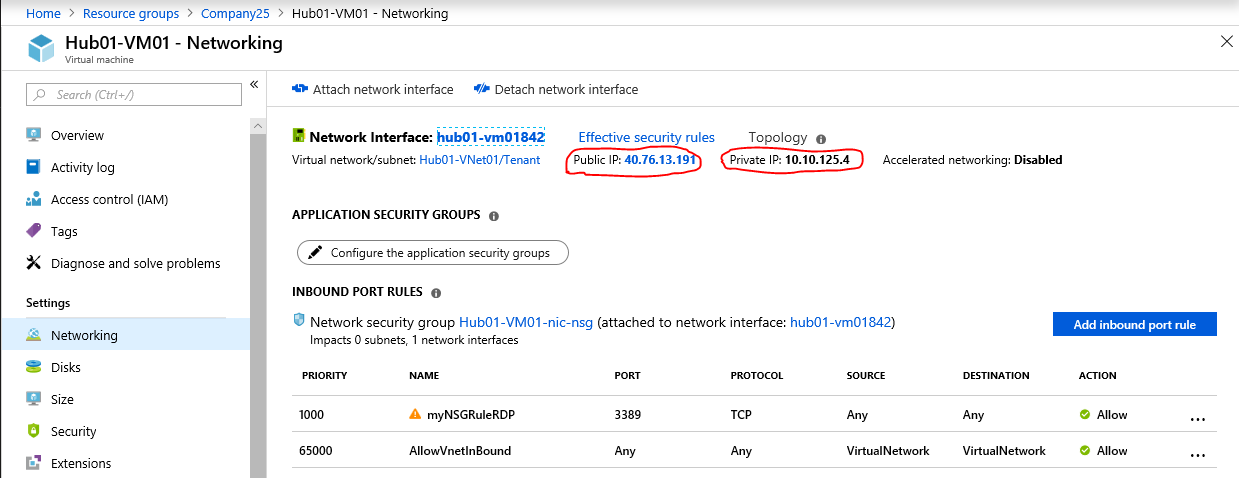


The deployment will take about 8 minutes. As the VM and its associated resources are being created, you will be presented with the deployment progress:

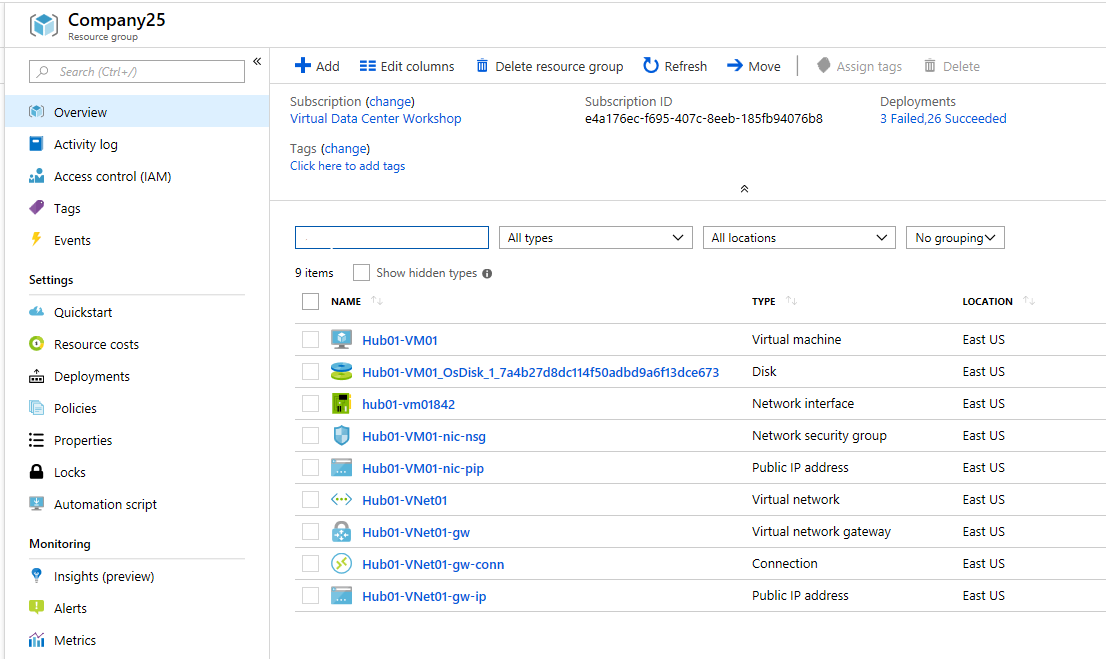


## Validation

Once the VM is successfully created, you can RDP to its public IP address from your Internet-connected local machine and Ping/RDP to its private IP address from your on-premises VM. The on-premises VM RDP URL is on the printed flash card provided to you. You can find the public and private IP addresses of the VM by clicking on the VM name on your base blade and then clicking on on the left navigation pane of the *Virtual machine* blade.



After completing Step 4, your base blade should appear as below, with 9 different resources.



# Step 5: Create a VNet Peering, ILB, and VMSS

In this step, let’s do the following:

* Create a Spoke VNet
* Enable VNet peering between the Hub and the Spoke VNets
* Create a basic Internal Load Balancer (ILB)
* Create a Virtual Machine Scale Set (VMSS)
* Link VMSS to the backend pool of ILB

## Create a (Spoke) VNet

Information needed to create the VNet:

VNet Name: **Spoke01-VNet01**

Address space: **10.10.1***XX***.128/26** (ex. 10.10.125.128/26)

Subscription (Name): **Virtual Data Center Workshop**

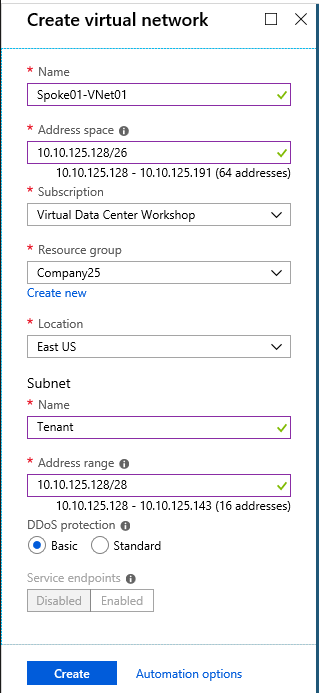
Resource group: **Company***XX* (ex: Company25)

Location (Azure Region): **East US**

Subnet Name: **Tenant**

Subnet Address range: **10.10.1***XX***.128/28** (ex. 10.10.125.128/28)

If you need help creating the VNet, refer to *Create VNet* in *Step 1*. After entering the required information in the *Create virtual network* blade, click  at the bottom of the blade.



## Enable VNet peering

Let’s enable VNet peering between the hub and spoke VNet. This is a two-step process: (1) Enable VNet peering on the hub, (2) Enable VNet peering on the spoke.

As part of the VNet peering configuration, we will enable the spoke VNet to use the hub VNet’s gateway.

Information needed to enable VNet peering on the hub:

Name: **Hub01toSpoke01**

Virtual network deployment model: **Resource manager**

Subscription: **Virtual Data Center Workshop**

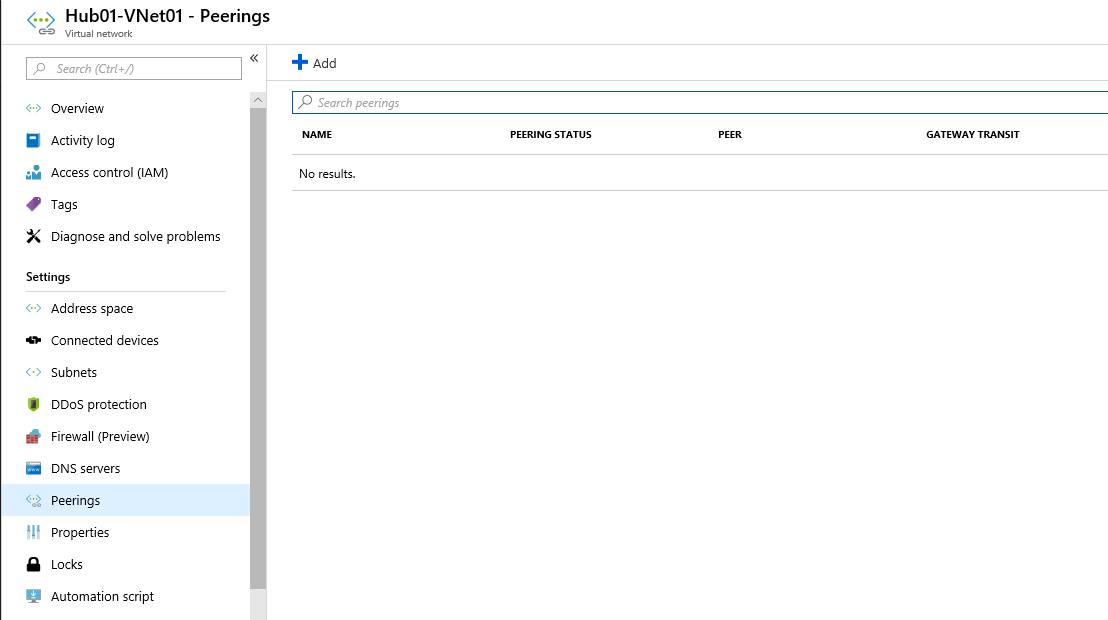
Virtual network: **Spoke01-VNet01**

Allow virtual network access: **Enabled**

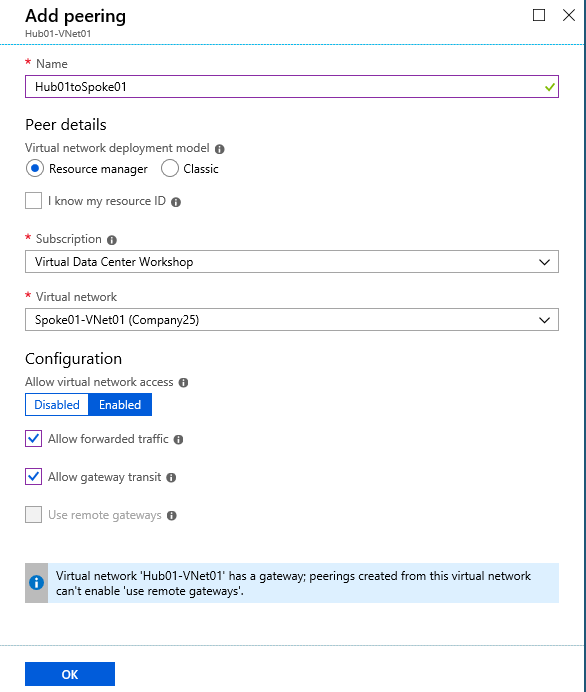
**Allow forwarded traffic**

**Allow gateway transit**

On your base blade, click on the name of the Hub VNet (Hub01-VNet01). This will open the *Virtual network* blade. On the blade click  on the left navigation pane. This will open the *peering* pane.



Click at the top of the pane. This will open the *Add peering* blade. In the blade, enter the required information and click .



Close Hub01-VNet01 blade and land on your base blade.

As we did on the hub VNet, create VNet peering on the spoke VNet. Information needed to enable VNet peering on the spoke:

Name: **Spoke01toHub01**

Virtual network deployment model: **Resource manager**

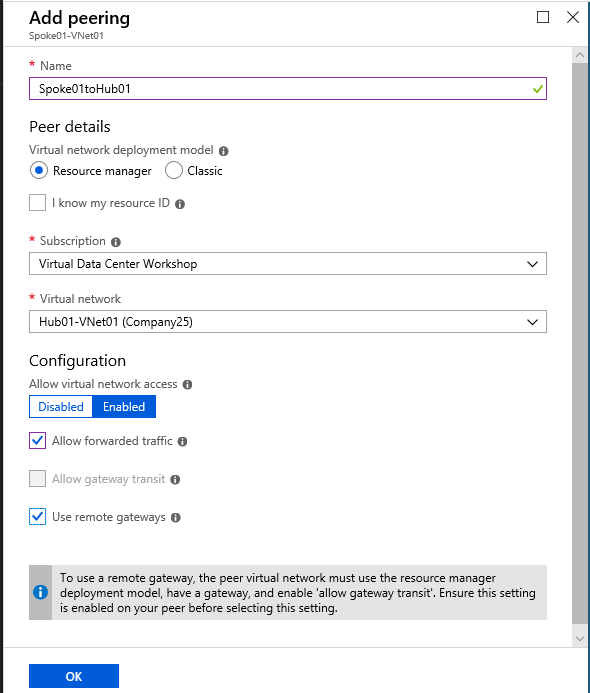
Subscription: **Virtual Data Center Workshop**

Virtual network: **Hub01-VNet01**

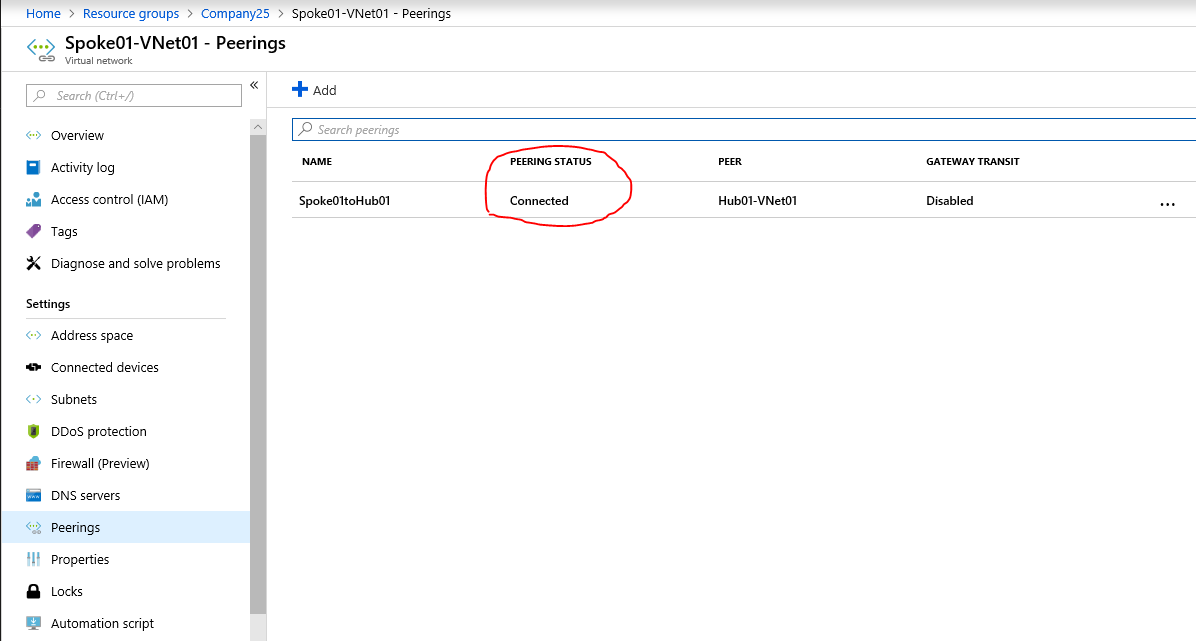
Allow virtual network access: **Enabled**

**Allow forwarded traffic**

**Use remote gateways**



Once the Vnet peering is added, refresh the peering pane. The peering status of the VNet peering should be *Connected*.



## Create an Internal Load Balancer

Information needed to create the load balancer.

Name: **Spoke01-lb**

Type: **Internal**

SKU: **Basic**

Virtual network: **Spoke01-VNet01**

Subnet: **Tenant**

IP address assignment: **Static**

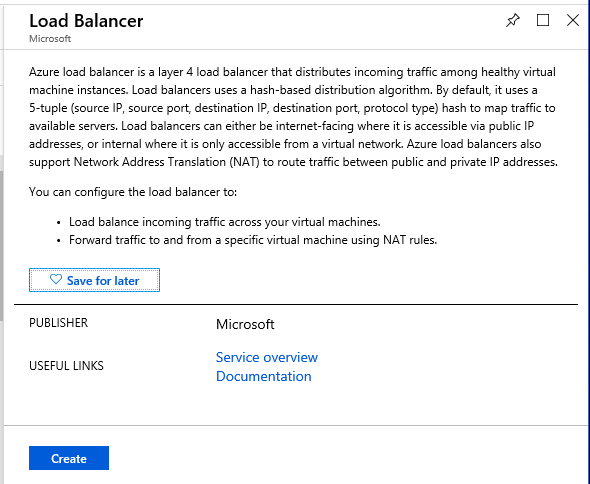
Private IP address: **10.10.1***XX***.142**

Subscription: **Virtual Data Center Workshop**

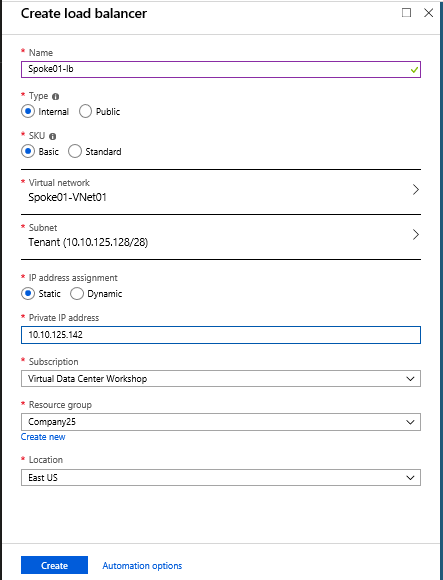
Resource group: **Company***XX*

Location: **East US**

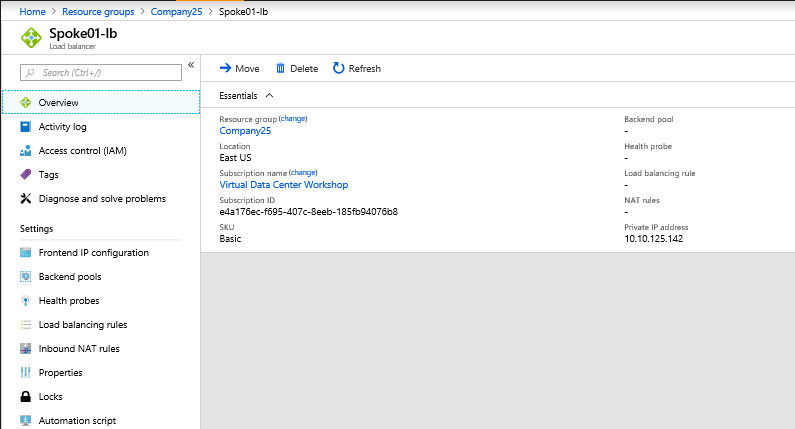
On your base blade, click  at the top. On the *Everything* blade, type “load balancer” on the search bar at the top and choose *Load Balancer*. This will open the *Load Balancer blade*. Click  at the end of the blade.



On the *Create load balancer* blade, enter the required information and click  at the bottom of the blade.

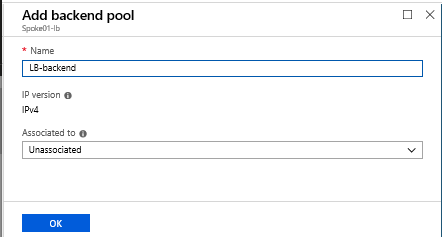


Once the load balancer is successfully deployed, click on the name of the load balancer on your base blade and open the *Load balancer* blade to further configure it.



### Configure the Backend Pools

On the Load Balancer blade, click  on the left navigation pane. On the Backend pools pane, click  at the top of the pane. On the *Add backend pool* blade, enter the Name of the pool as *LB-backend* and click  at the bottom of the blade.



Information needed to configure the health probe of the load balancer:

Name: **HealthProbe**

IP version: IPv4

Protocol: **TCP**

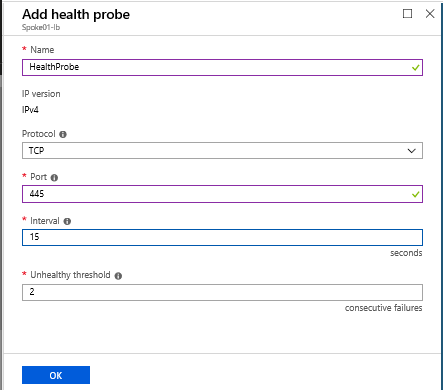
Port: **445**

Interval: **15** (seconds)

Unhealthy threshold: **2**

### Configure the Health probe

On the Load Balancer blade, click  on the left navigation pane. On the Backend pools pane, click  at the top of the pane. On the *Add health probe* blade, enter the required information and click  at the bottom of the blade.



### Configure the Load balancing rules

Information needed to configure load balancing rules:

Rule 1:

Name: **SMB445**

IP version: IPv4

Frontend IP address: **10.10.1***XX***.142**

Protocol: **TCP**

Port: **445**

Backend port: **445**

Backend pool: **LB-backend**

Health probe: **HealthProbe (TCP:445)**

Idle timeout (minutes): **15**

Rule 2:

Name: **SMB137**

IP version: IPv4

Frontend IP address: **10.10.1***XX***.142**

Protocol: **TCP**

Port: **137**

Backend port: **137**

Backend pool: **LB-backend**

Health probe: **HealthProbe (TCP:445)**

Idle timeout (minutes): **15**

Rule 3:

Name: **SMB139**

IP version: IPv4

Frontend IP address: **10.10.1***XX***.142**

Protocol: **TCP**

Port: **139**

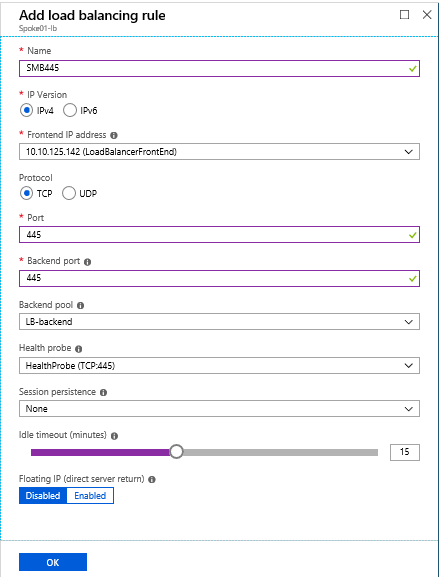
Backend port: **139**

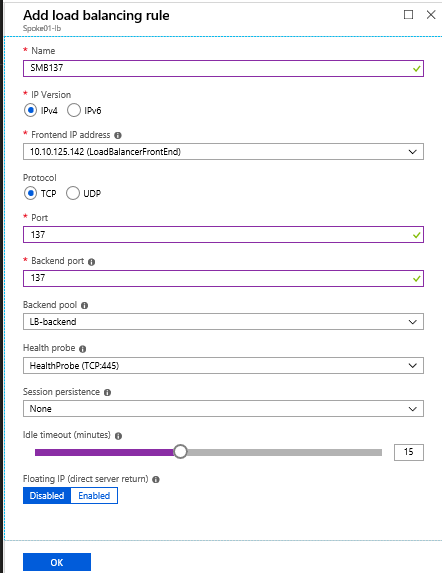
Backend pool: **LB-backend**

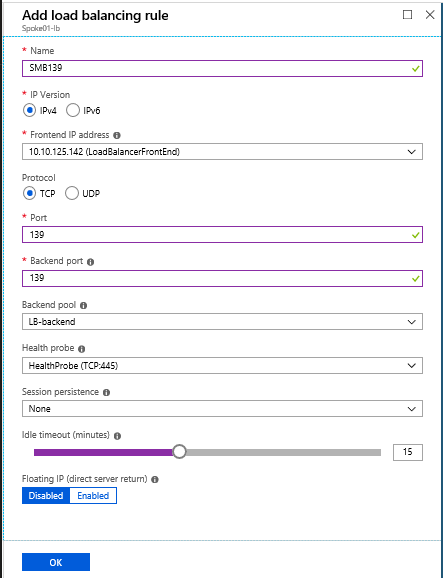
Health probe: **HealthProbe (TCP:445)**

Idle timeout (minutes): **15**

On the Load Balancer blade, click  on the left navigation pane. Click  at the top of the pane. On the *Add load balancing rule* blade, enter the required information for each rule and click  at the bottom of the blade.







## Create a Virtual machine scale set

Information needed to create the create virtual machine scale set (for the fields not mentioned below, default value is preferred).

Virtual machine scale set name: **Spoke01VM**

Operating system disk image: **Windows Server 2016 Datacenter**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX*

Location: **East US**

Username: **Student***XXX*

Password: (if haven’t noted, see “Accessing Azure Key Vault”, at the beginning of this document).

Instance count: **2**

Instance size: **A4\_v2 (4 vCPUs, 8 GB)**

Deploy as low priority: **No**

Use managed disks: **Yes**

Autoscale: **Disabled**

SKU: **Basic**

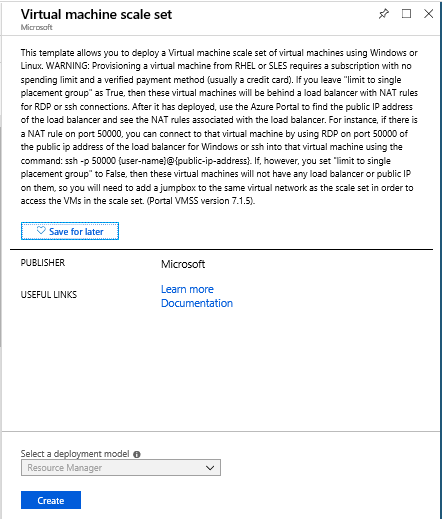
Virtual network: **Spoke01-VNet01**

Subnet: **Tenant**

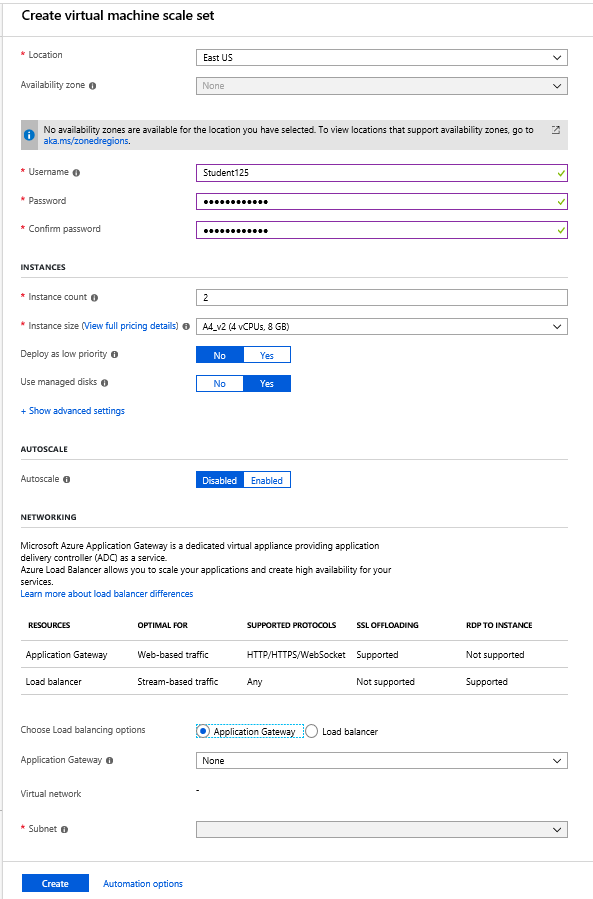
IP address assignment: **Static**

Private IP address: **10.10.125.142**

On your base blade, click  at the top. On the *Everything* blade, start typing “virtual machine scale set” on the search bar at the top and choose *Virtual machine scale set*. This will open the *Virtual machine scale set introduction* blade. Click  button at the end of the blade.

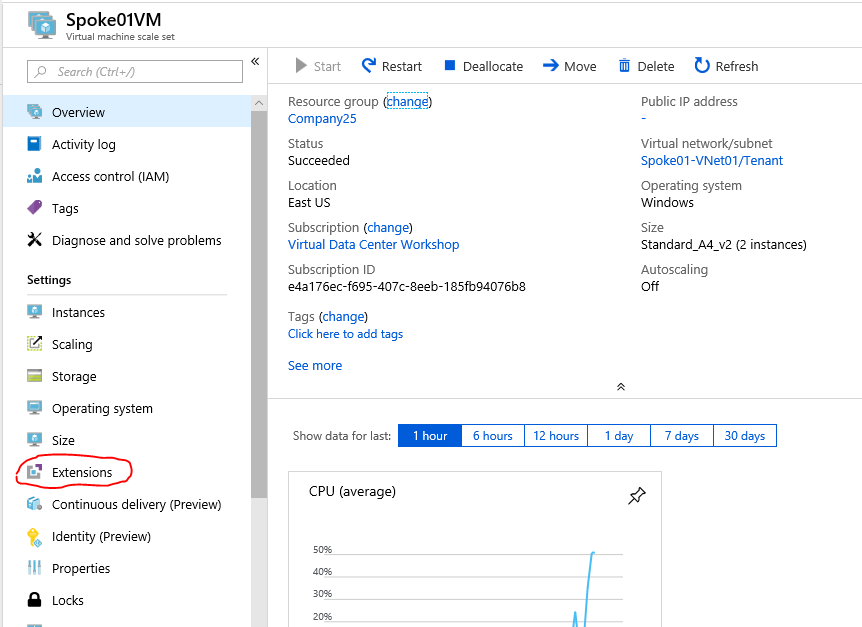


On the *Create virtual machine scale set* blade, enter the required information and click  at the end of the blade.

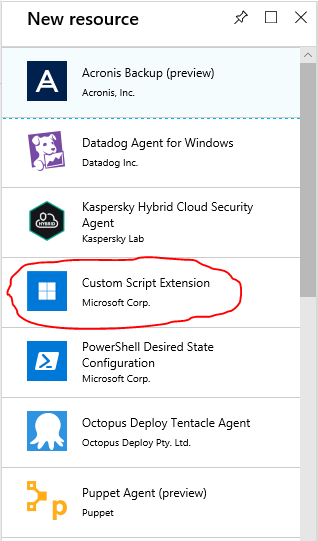


### Post deploy configuration

Once the scale set is successfully deployed, on your base blade click *Spoke01VM* to open the *Virtual machine scale set* blade.



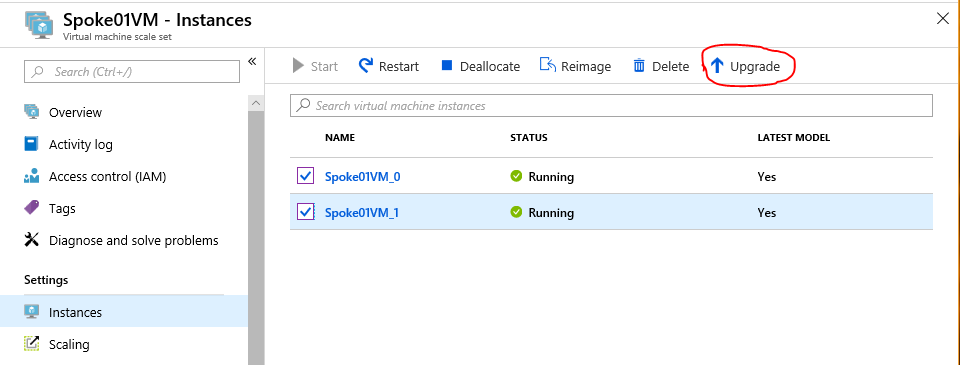
On the blade, click on the left navigation pane. This will open *Extensions* pane. Click on top of the pane. This will open the *New resource* blade.



On the *New resource* blade, select *Custom Script Extension*. This will open the *Custom Script Extension* introduction blade. Click  at the end of the blade. This will open the *Install extension* blade. In the blade, click  and choose *FSBuild.ps1* file in the Microsoft Azure flash drive given to you at the beginning of the workshop and click  at the bottom of the blade.

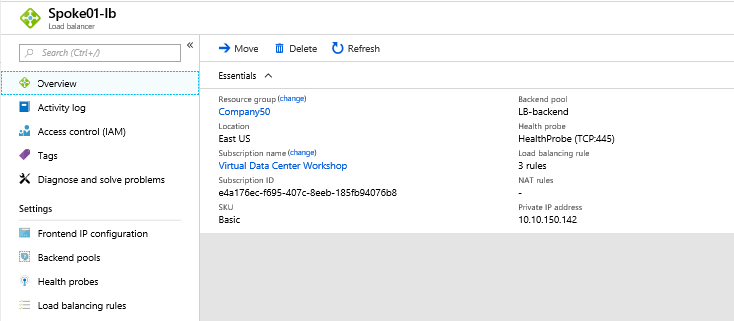


On the Virtual machine scale set, click on the left navigation pane. This will open the *Instances* pane. Choose all the instances and hit at the top of the pane.

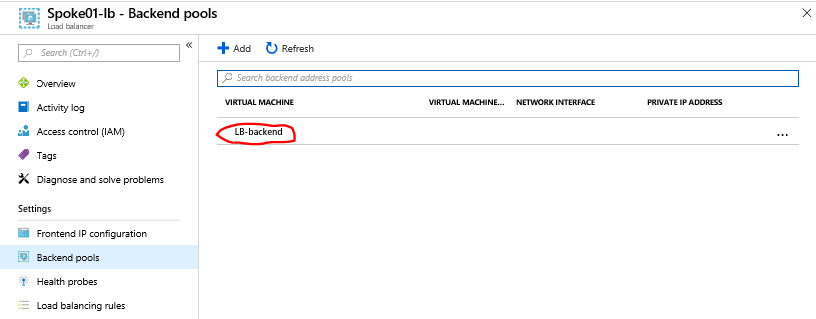


## Link VMSS to the backend pool of the iLB

On your base blade, click on the name of iLB (“Spoke01-lb”). This will open the *Load balancer* blade.



Click on the left navigation pane. This will open the *Backend pools* pane.



Click on the listed backend pool name (“LB-backend”). This will open the *LB-backend* blade. In the blade, under *Associated to* choose *Virtual machine scale set*. Under *Virtual machine scale set* choose the VMSS we just created. After linking VMSS with the backend pool click at the top of the blade.

## Validation

On your base blade, click *Spoke01-VNet01* and open the Virtual network blade. Note the private IP addresses assigned to your Spoke01VM instances.

# Step 6: Deploy an AppGateway

In this step, let’s do the following:

* Create a second Spoke VNet
* Between the Hub and the second Spoke VNets, enable VNet peering
* Create an NSG
* Create multiple VMs
* Create AppGateway

## Create a (Spoke) VNet

Information needed to create the VNet:

VNet Name: **Spoke02-VNet01**

Address space: **10.10.1***XX***.192/26** (ex. 10.10.125.128/26)

Subscription (Name): **Virtual Data Center Workshop**

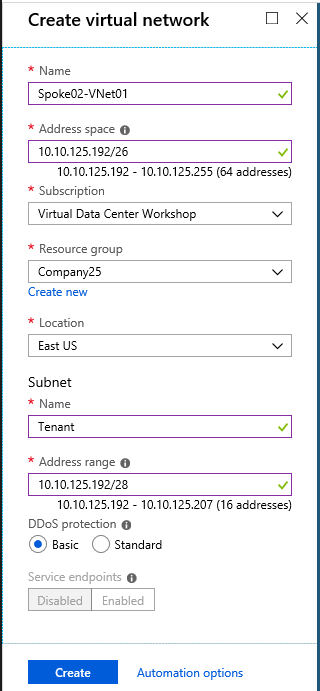
Resource group: **Company***XX* (ex: Company25)

Location (Azure Region): **East US**

Subnet Name: **Tenant**

Subnet Address range: **10.10.1***XX***.192/28** (ex. 10.10.125.192/28)

Recall you created a Vnet in *Step 1.*



### Create an additional Subnet on the VNet

To create an additional subnet within the VNet, click on the name of the VNet (*Spoke02-VNet01*) in your base blade. This will open the VNet blade.

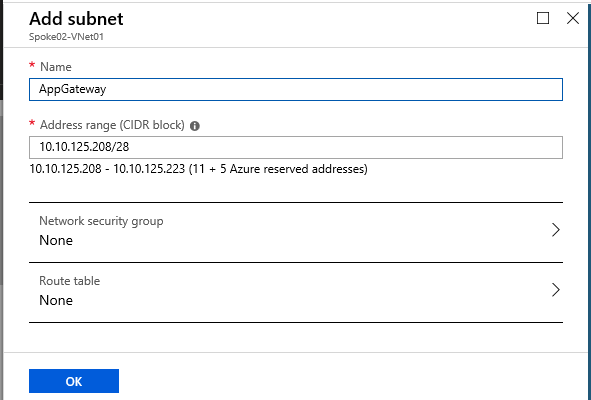
On the *Virtual network* blade, click  on the left navigation pane, to open the *Subnets* blade.

Using the following information, add the subnet.

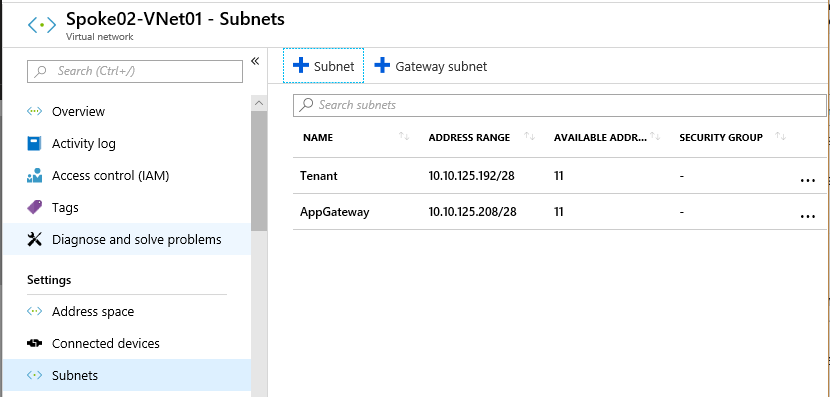
Subnet Name: **AppGateway**

Address Range: **10.10.1***XX***.208/28** (ex: 10.10.125.208/28)

To add the subnet, click on the top of the blade.



After entering the address range, click  at the bottom of the *Add subnet* blade. Once the subnet is created, you will receive the following notification: *Successfully added subnet*. The new Subnet will be listed on the *Subnets* blade of the VNet.



## Enable VNet peering

Recall you enabled Vnet peering in *Step 5.* Information needed to enable VNet peering on the hub:

Name: **Hub01toSpoke02**

Virtual network deployment model: **Resource manager**

Subscription: **Virtual Data Center Workshop**

Virtual network: **Spoke02-VNet01**

Allow virtual network access: **Enabled**

**Allow forwarded traffic**

**Allow gateway transit**

Information needed to enable VNet peering on the spoke:

Name: **Spoke02toHub01**

Virtual network deployment model: **Resource manager**

Subscription: **Virtual Data Center Workshop**

Virtual network: **Hub01-VNet01**

Allow virtual network access: **Enabled**

**Allow forwarded traffic**

**Use remote gateways**

Confirm, the peering status of the VNet peering you just created is *Connected*.

## Create a Network Security Group

Information we need to create an NSG.

Name: **Spoke02-VM-nic-nsg**

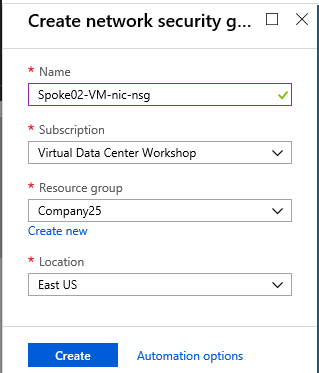
Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Location: **East US**

On your base blade, click  at the top. This will open the *Everything* blade. Start typing “network security group” on the search bar on the top and choose *Network security group*. This will open the *Network security group* introduction blade. Click  at the end of the blade.

On the *Create network security group* blade, enter the required information and click  . Following, a quick validation, the NSG will be created.



In the newly created NSG, add an inbound security rule to allow RDP traffic.

Information needed to create an inbound security rule to allow RDP traffic:

Source: **Any**

Source port ranges: **\***

Destination: **Any**

Destination port ranges: **3389**

Protocol: **TCP**

Action: **Allow**

Priority: **1000**

Name: **myNSGRuleRDP**

Description: To permit RDP from any source

With the successful addition of the security rule, you will see the rule listed along with the default inbound security rules of the NSG.

## Create Virtual Machines

Let’s create three virtual machines. Information we need to create the virtual machine.

Basics

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Virtual machine name: **Spoke02-VM01** / **Spoke02-VM02** / **Spoke02-VM03**

Region: **East US**

Image: **Windows Server 2016 Datacenter**

Size: **Standard A4 v2**

Username: **Student***XXX*

Password: (if not noted, see “Accessing Azure Key Vault”, at the beginning of this document).

Networking

Virtual network: **Spoke02-VNet01**

Subnet: **Tenant**

Public IP: **None**

Network security group: **Advanced**

Configure network security group: **Spoke02-VM-nic-nsg**

Management

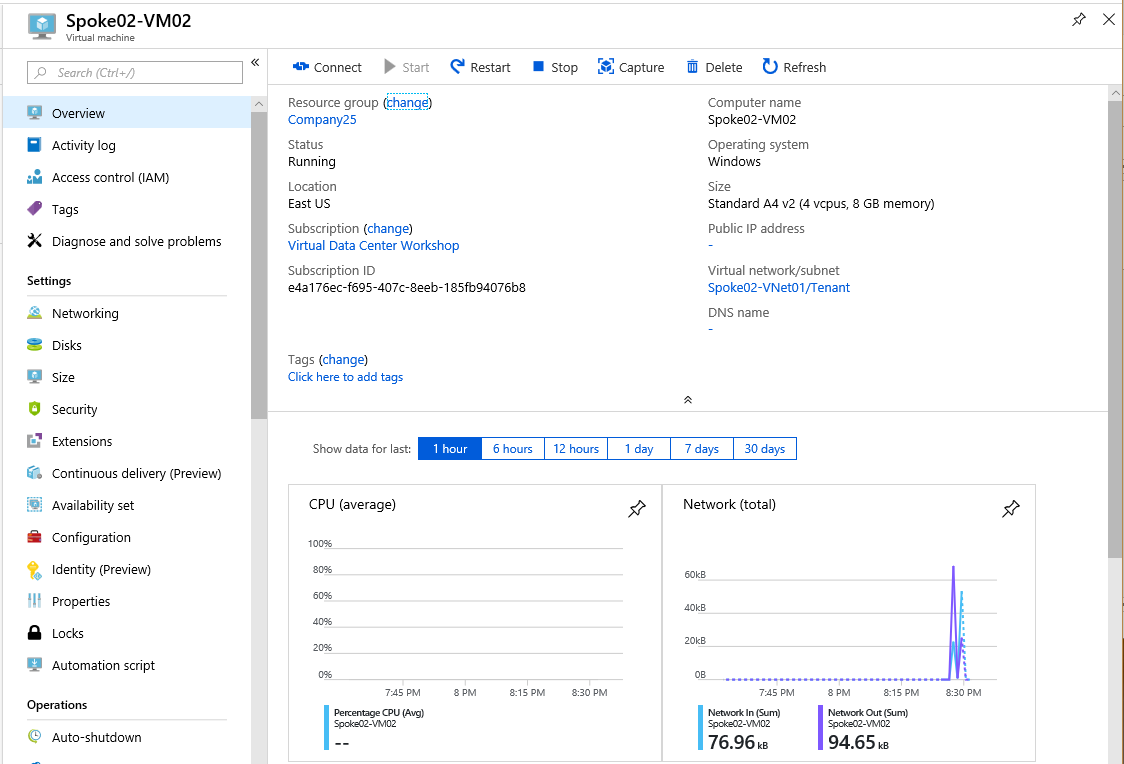
Boot diagnostics: **Off**

If you need help creating the virtual machines, refer to *Create Virtual Machines* in *Step 4* of this document. Unlike in *Step 4*, we are not going to run extensions as part of the VM creation and instead run it after the VM creation.

### Post Deploy Build

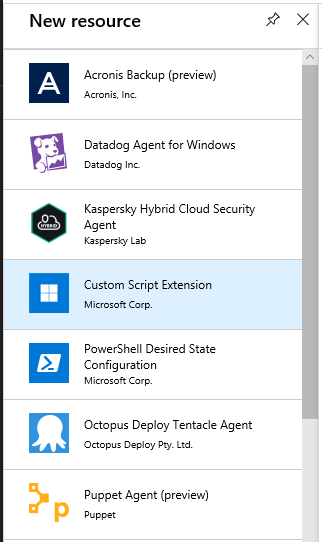
After successful deployment of virtual machines, use *Custom Script Extension* to install IIS and create web app pages on the VMs.

On your base blade, click on the name of a VM we just created. This will open the *Virtual machine* blade.

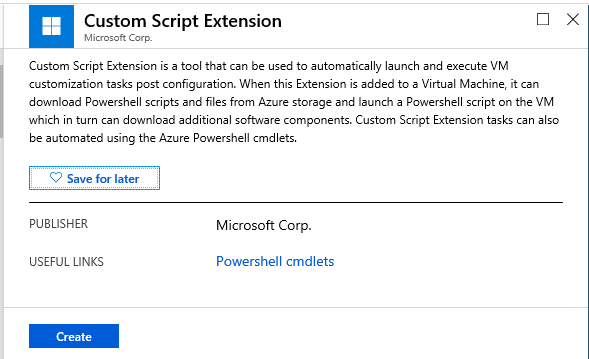


Click  on the left navigation menu. This will open the *Extensions* pane.

Click at the top of the pane. This will open the *New resource* blade. In the blade choose *Custom Script Extension*.



This will open *Custom Script Extension* introduction blade.



Click  at the bottom of the blade. This will open the *Install extension* blade. In the blade click and choose IISBuild.ps1 file in the Microsoft Azure flash drive given to you at the beginning of the workshop and click  at the bottom of the blade.



Repeat the post deploy build for the other two VMs create.

## Validation

Once the VM is successfully created, you should be able to RDP to its public IP address from your Internet-connected local machine and Ping/RDP to its private IP address from your on-premises VM. The on-premises VM RDP URL is on the printed flash card provided to you. You can find the public and private IP addresses of the VM by clicking the VM name on your base blade and then clicking on the left navigation pane of the Virtual machine blade.

## Create an AppGateway

Information we need to create the AppGateway.

Basics

Name: **Spoke02-AppGW**

Tier: **WAF**

SKU size: **Medium**

Instance count: **2**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Region: **East US**

Settings

Virtual network: **Spoke02-VNet01**

Subnet: **AppGateway**

Frontend IP configuration: **Public**

Public IP address: **Create new**

**Spoke02-AppGW-pip**

Protocol: **HTTP**

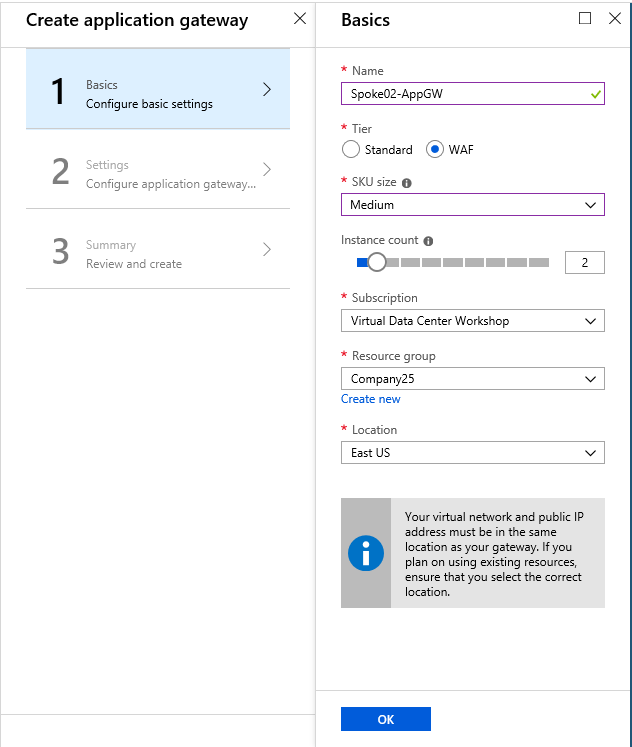
Port: **80**

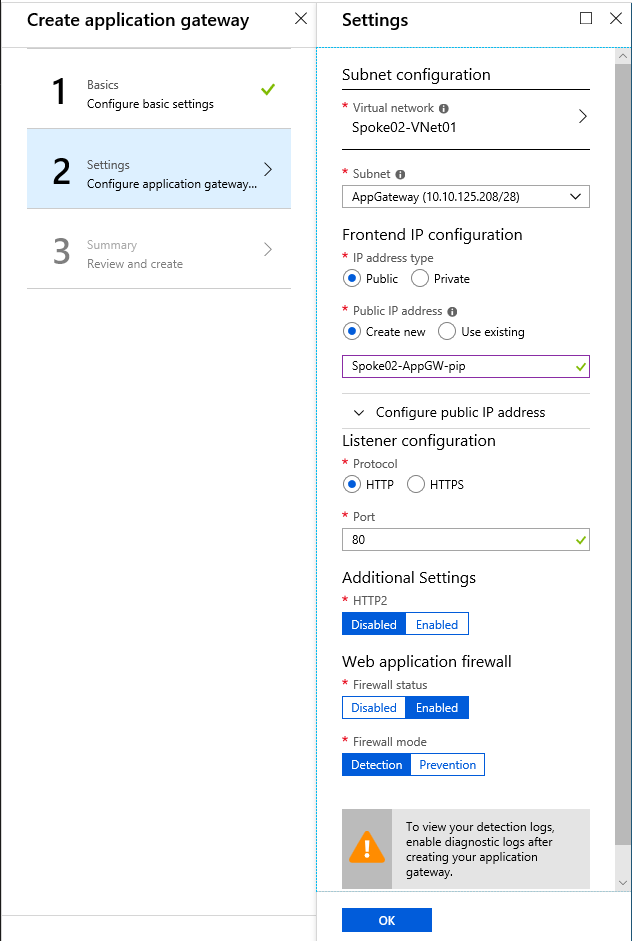
HTTP2: **Disabled**

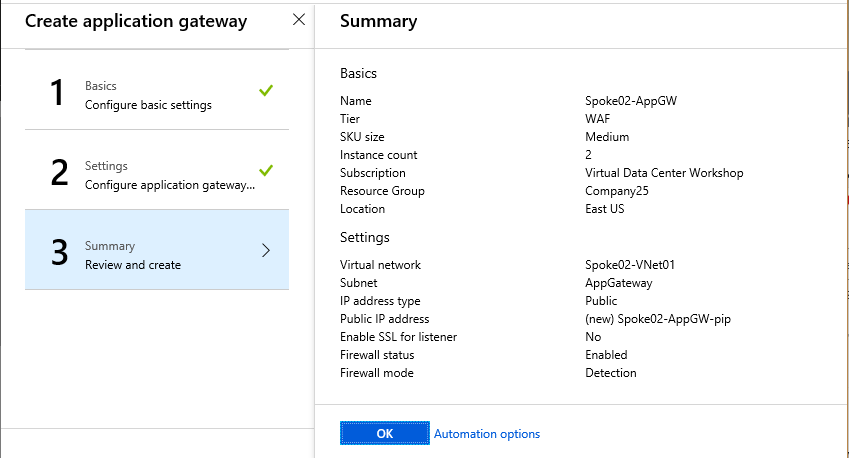
Firewall status: **Disabled**

On your base blade, click  button at the top. On the *Everything* blade, start typing “Application Gateway” on the search bar at the top and choose *Application Gateway*. This will open the *Application Gateway* introduction blade. Click  at the end of the blade.

This will open the *Create application gateway* blade. Enter the required information on the blade. The blade requires you to enter the information in three steps. After entering the information in each step, click  at the end of each pane to progress further.





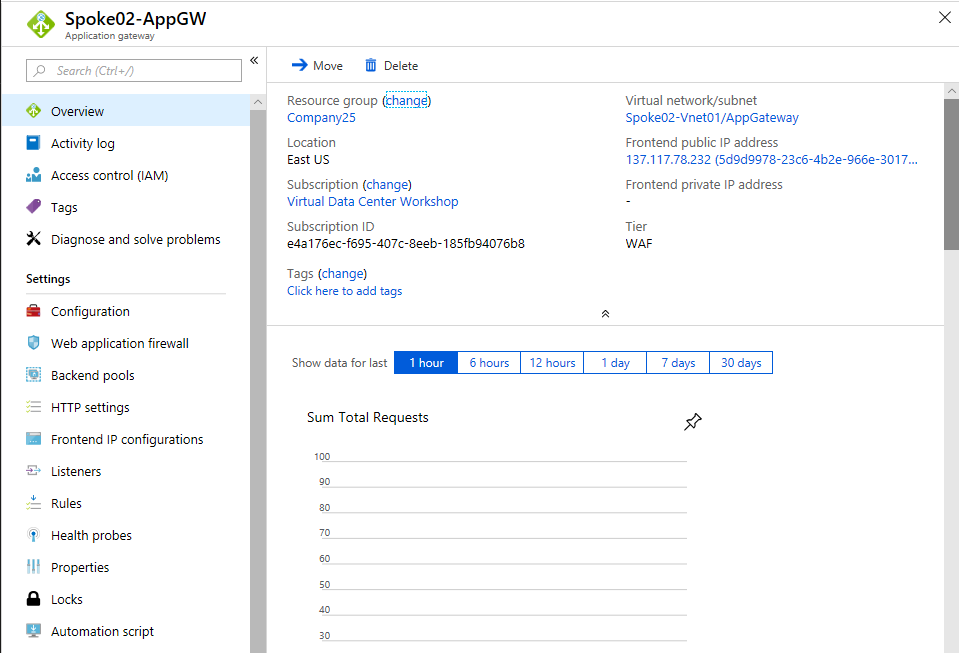


The WAF tier AppGateway creation will take about 20 minutes. The new tier—WAF 2—available for preview will be provisioned in 5 to 6 minutes. The tier name may change following the preview period. The new tier has the following benefits:

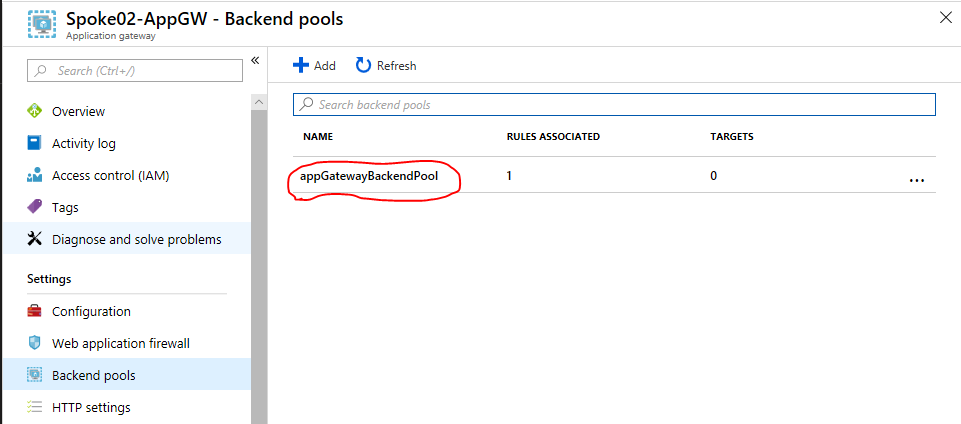
* Autoscaling: no longer need to overprovision at anticipated peak capacity or guess SKU size/instance count
* Better SSL perf: 5X increase in SSL offloads per core
* Better provisioning and update time: 5 to 6 mins to provision compared to 20 mins, to update via REST API
* Zone redundancy: same AppGw deployment spans 3 zones. Is user configurable – could be two zones or single zone as well
* Support for static frontend IP
* Ingress Controller support for AKS

### Post deployment configuration

Following successful deployment of the AppGateway, click on the name of the AppGateway (*Spoke02-AppGW*) in your base blade. This will open the *Application gateway* blade.



Click  on the left navigation pane. This will open the *Backend pools* pane.



Click on the backend pool—*appGatewayBackendPool*—listed. This will open the *Edit backend pool* blade. Use the following information to edit the backend pool:

Targets: **Virtual machine**

Virtual Machine Network Interfaces

**Spoke02-VM01**  **Spoke02-VM01**yyyy (yyyy differs)

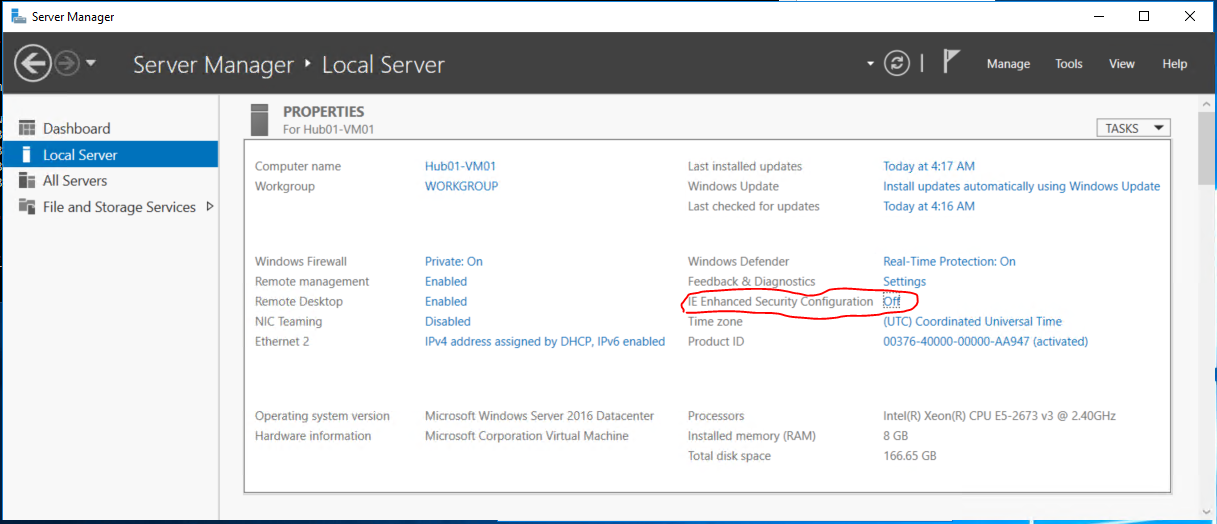
**Spoke02-VM02** **Spoke02-VM02**yyyy

**Spoke02-VM03** **Spoke02-VM03**yyyy

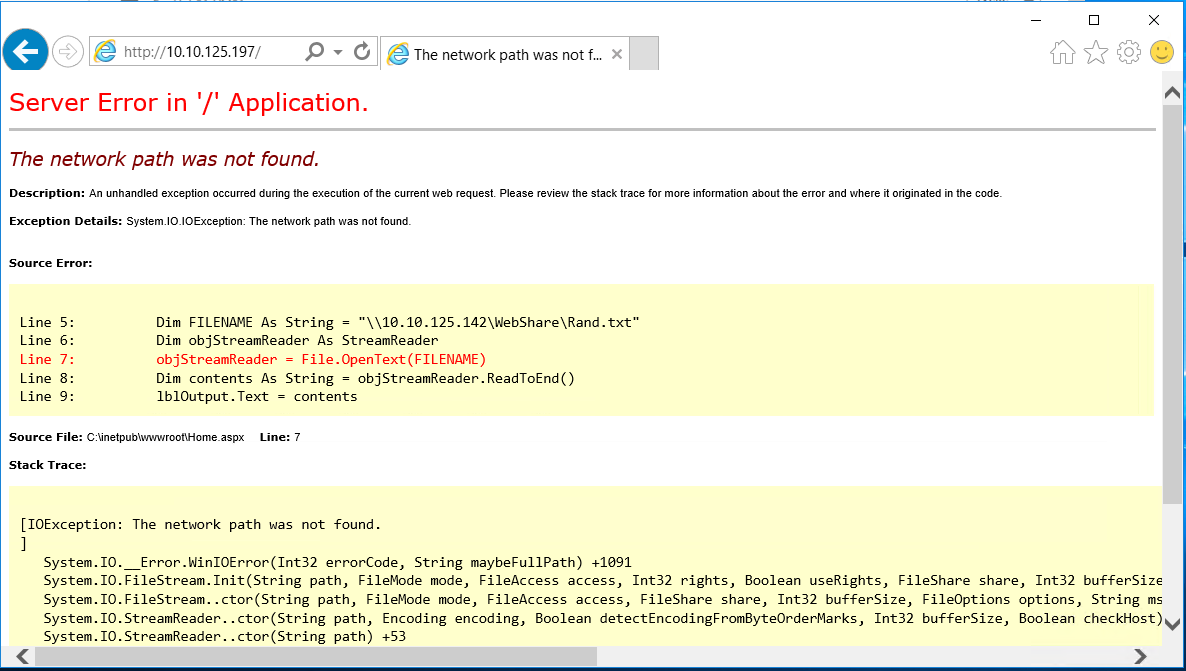
Click  at the top of the blade.

### Validation

On the Hub01-VM01 go to *Server Manager*. In the *Server Manager*, under *Local Server*, turn off *IE Enhanced Security Configuration*.

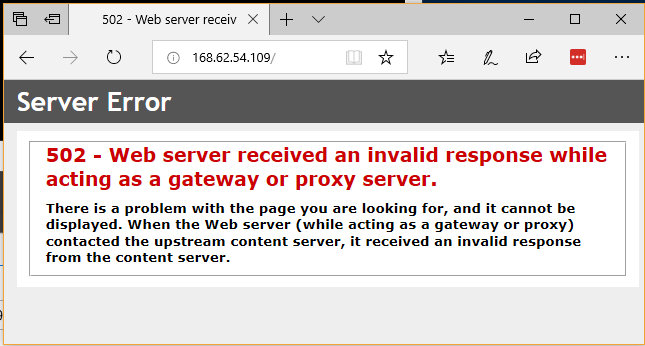


Now open Internet Explorer and navigate to the private IP address of Spoke02-VM01 (e.g. <http://10.10.125.197/>). You should get the following error message in response.



The reason we see the error message is because the VMs behind the AppGW are unable (see Line 5 in the above screenshot) to fetch the Rand.txt file from IIS servers sitting behind the internal Load Balancer in Spoke01-VNet01. In the next step, we will fix the issue.

Let’s check how the AppGW handles the request. Note the AppGW public IP address from the AppGW blade and navigate to the IP from any browser that has Internet connection.



Note: The error message you receive through AppGW is abstract and the AppGW masks the issue details.

# Step 7: Deploy a Firewall in High Available Configuration

In this step, we will accomplish the following:

* Create a HA load balancer
* Create an NSG
* Create an availability set for firewall
* Create virtual machines within an availability set and associate the NSG
* Link virtual machines to a load balancer backend pool
* Create UDR rules to force traffic between Spoke01 and Spoke02 to go via the firewall

## Create a HA load balancer

Information needed to create the load balancer.

Name: **Hub01-FW-lb**

Type: **Internal**

SKU: **Standard**

Virtual network: **Hub01-VNet01**

Subnet: **Firewall**

IP address assignment: **Static**

Private IP address: **10.10.1***XX***.30**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX*

Location: **East US**

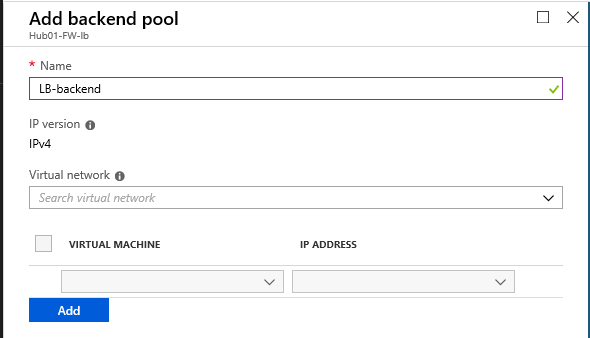
Recall, you have configured an internal load balancer in *Step 5*. On your base blade, click on the button at the top. On the *Everything* blade, type “load balancer” on the search bar at the top and choose *Load Balancer*. This will open the *Load Balancer* introduction blade. Click  at the end of the blade.

On the *Create load balancer* blade, enter the required information and click  at the bottom of the blade.

Once the load balancer is successfully deployed, click on the name of the load balancer (*Hub01-FW-lb*) on your base blade and open the *Load balancer* blade to further configure it.

### Configure the Backend Pools

On the *Load Balancer* blade, click  on the left navigation pane. On the *Backend pools* pane, click  at the top of the pane. On the *Add backend pool* blade, enter the Name of the pool as “LB-backend” and click  at the bottom of the blade.



### Configure the Health probe

Information needed to configure the health probe of the load balancer:

Name: **HealthProbe**

IP version: IPv4

Protocol: **HTTP**

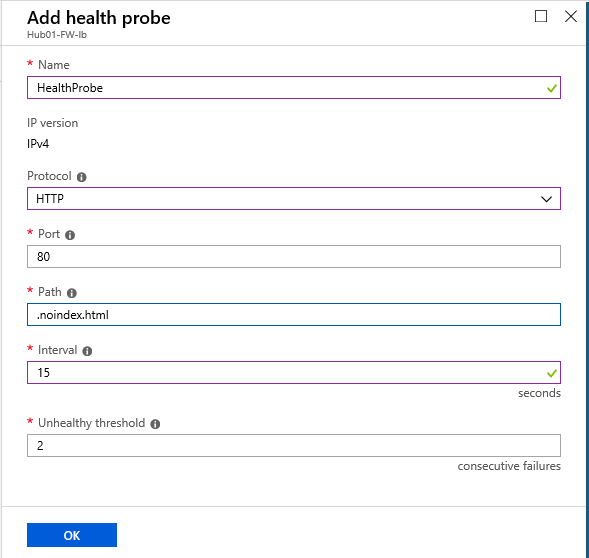
Port: **80**

Path: **.noindex.html**

Interval: **15** (seconds)

Unhealthy threshold: **2**

On the *Load Balancer* blade, click  on the left navigation pane. On the *Backend pools* pane, click  at the top of the pane. On the *Add health probe* blade, enter the required information and click  at the bottom of the blade.



### Configure the Load balancing rules

Information needed to configure load balancing rules:

Rule 1:

Name: **HAPortsRule**

IP version: IPv4

Frontend IP address: **10.10.1***XX***.30 (LoadBalancerFrontEnd)**

**HA Ports**

Backend pool: **LB-backend**

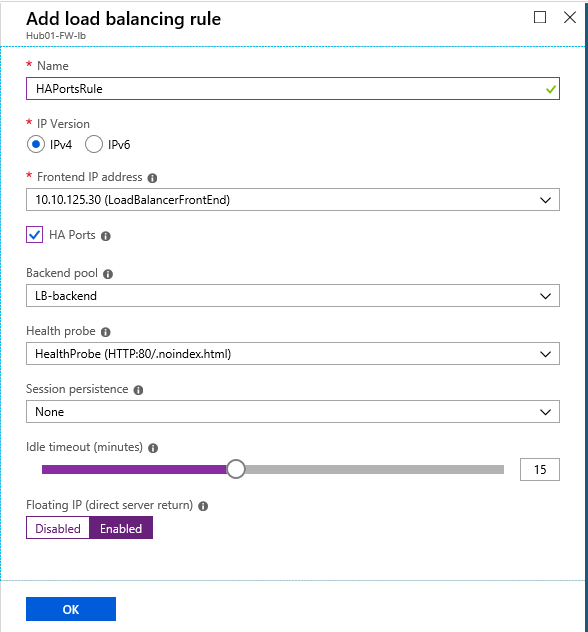
Health probe: **HealthProbe (HTTP:80/.noindex.html)**

Session persistence: **None**

Idle timeout (minutes): **15**

Floating IP: **Enabled**

On the *Load Balancer* blade, click  on the left navigation pane. On the *Load balancing rule* pane, click  at the top of the pane. On the *Add load balancing rule* blade, enter the required information for each rule and click  at the bottom of the blade.



## Create a Network Security Group

Information we need to create an NSG.

Name: **Hub01-FW-nic-nsg**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Location: **East US**

Recall, you have configured an NSG in *Step 4*. On your base blade, click  at the top. This will open the *Everything* blade. Start typing “network security group” in the search bar on the top and choose *Network security group*. This will open the *Network security group* introduction blade. Click  at the end of the blade.

On the *Create network security group* blade, enter the required information and click the  button. Following a quick validation, the NSG will be created.

On your base blade click on the name of the newly created NSG: *Hub01-FW-nic-nsg*.

Let’s add an inbound security rule to permit SSH traffic.

Information needed to create an inbound security rule to allow SSH traffic:

Source: **Any**

Source port ranges: **\***

Destination: **Any**

Destination port ranges: **22**

Protocol: **TCP**

Action: **Allow**

Priority: **1000**

Name: **myNSGRuleSSH**

Description: To permit SSH from any source

On the *Network security group* blade, click  on the left navigation pane. Next, click at the top of the Inbound security rules pane. On the *Add inbound security rule* blade, enter the required information and click  at the bottom of the blade.

## Create a VM Availability Set

Information we need to create an NSG.

Name: **Hub01-FW-as**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

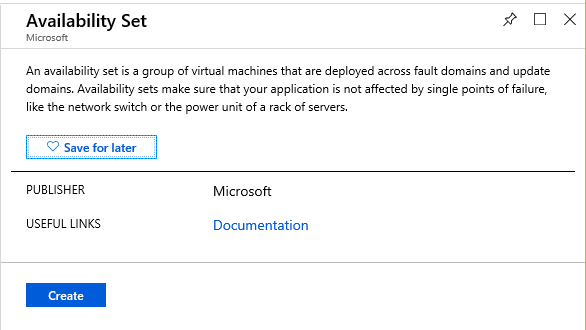
Location: **East US**

Fault domains**: 2**

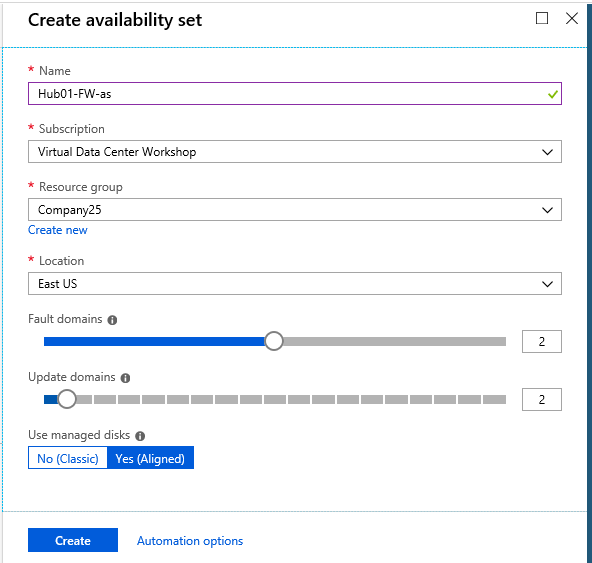
Update domains: **2**

Use managed disks: **Yes (Aligned)**

On your base blade, click  at the top. This will open the *Everything* blade. Start typing “availability set” in the search bar on the top and choose *Availability Set*. This will open the *Availability Set* introduction blade. Click  at the end of the blade.



On the *Create availability set* blade, enter the required information and click  at the bottom of the blade. Following a quick validation, the availability set will be created.



## Create Virtual Machines

Let’s create two virtual machines and associate the availability set and NSG that we just created. Information we need to create virtual machines.

Basics

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Virtual machine name: **Hub01-FW01** / **Hub01-FW02**

Region: **East US**

Availability options: **Availability set**

Availability set: **Hub01-FW-as**

Image: **CentOS-based 7.5**

Size: **Standard A4 v2** (you may have to clear all filters)

Username: **Student***XXX*

Password: (if not noted, see *Accessing Azure Key Vault*, at the beginning of this document).

Networking

Virtual network: **Hub01-VNet01**

Subnet: **Firewall**

Public IP: **none**

Network security group: **Advanced**

Configure network security group: **Hub01-FW-nic-nsg**

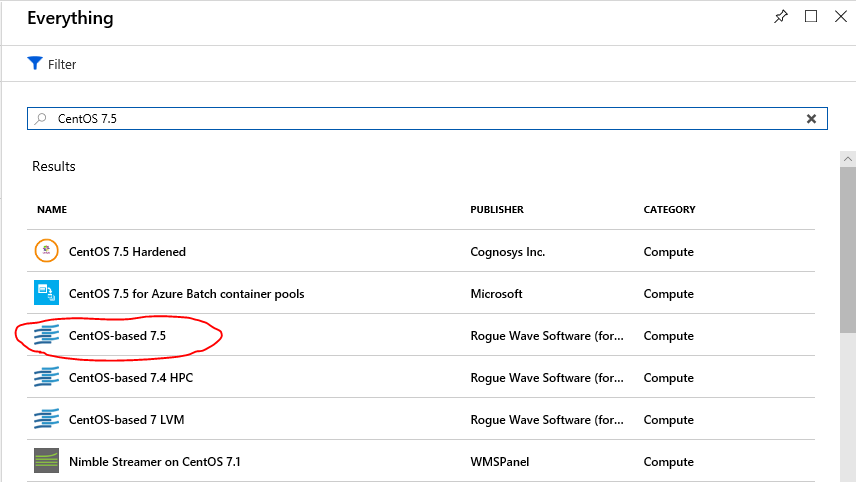
Management

Boot diagnostics: **Off**

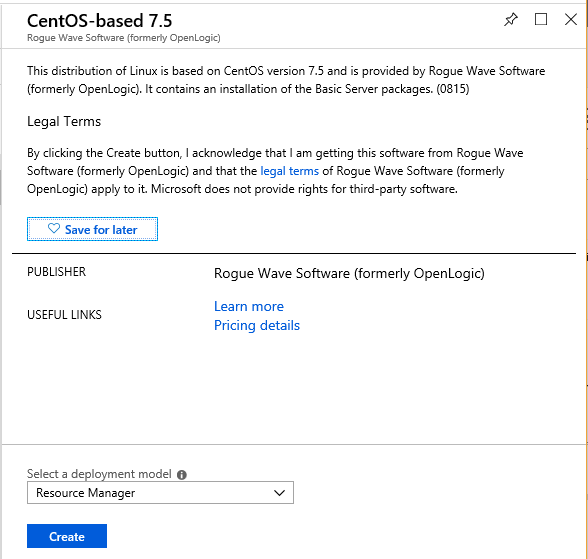
Guest config

We are going to use *Custom Script Extension* to allow ICMP traffic on the VM that we are creating (Alternatively, after the VM is created you can login to the VM and edit Inbound Rules of Windows Firewall with Advanced Security to permit the ICMP traffic).

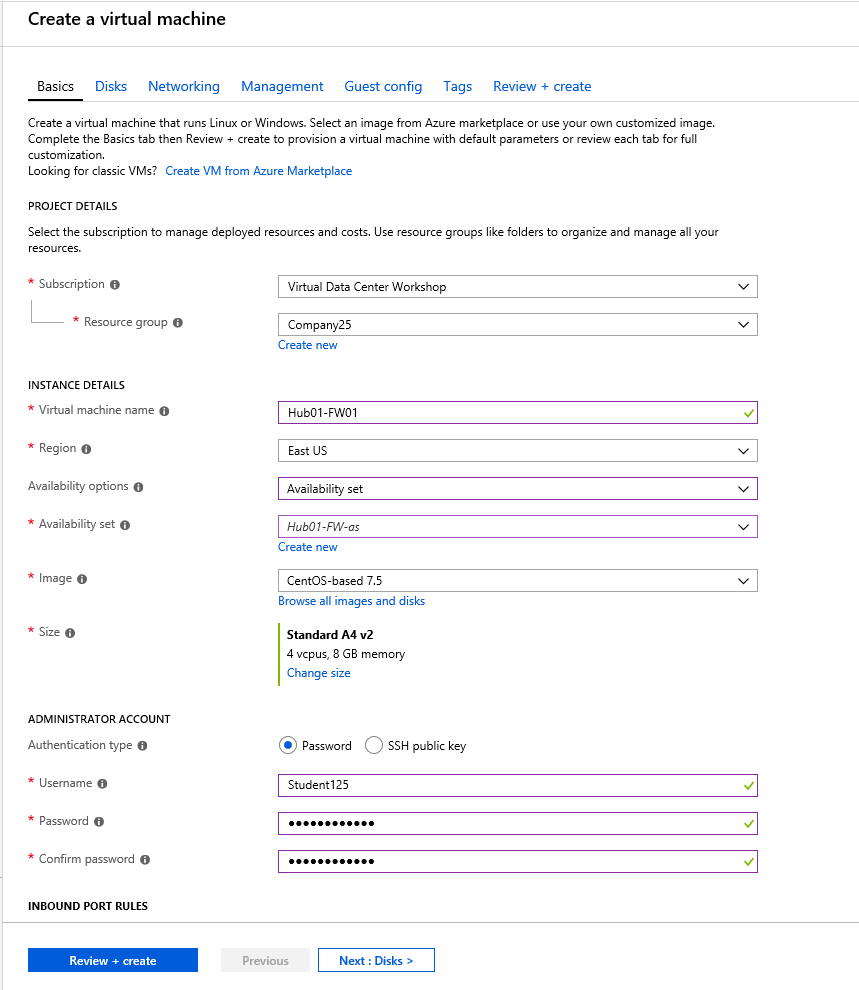
On your base blade, click  at the top. On the *Everything* blade, type “CentOS 7.5” in the search bar at the top and choose *CentOS-based 7.5* by Rogue Wave Software (formerly OpenLogic).

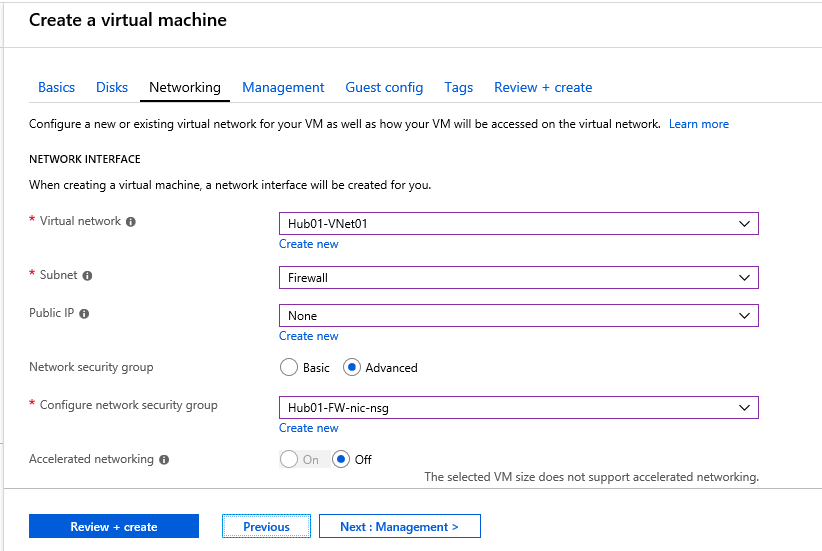


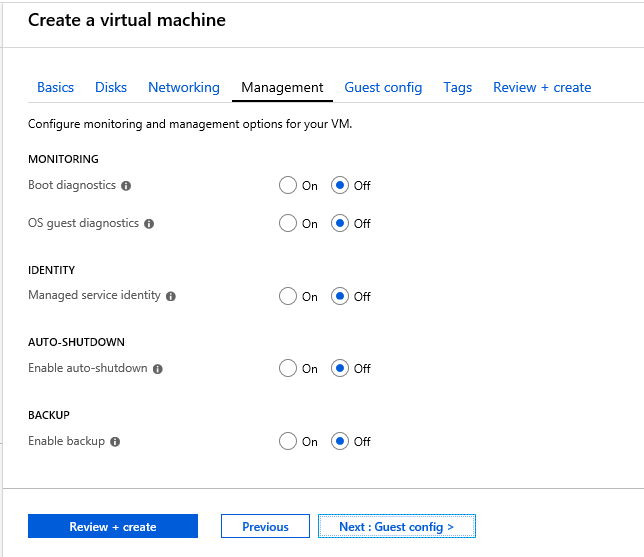
This will open the *CentOS-based 7.5* introduction blade. Click  at the end of the blade.



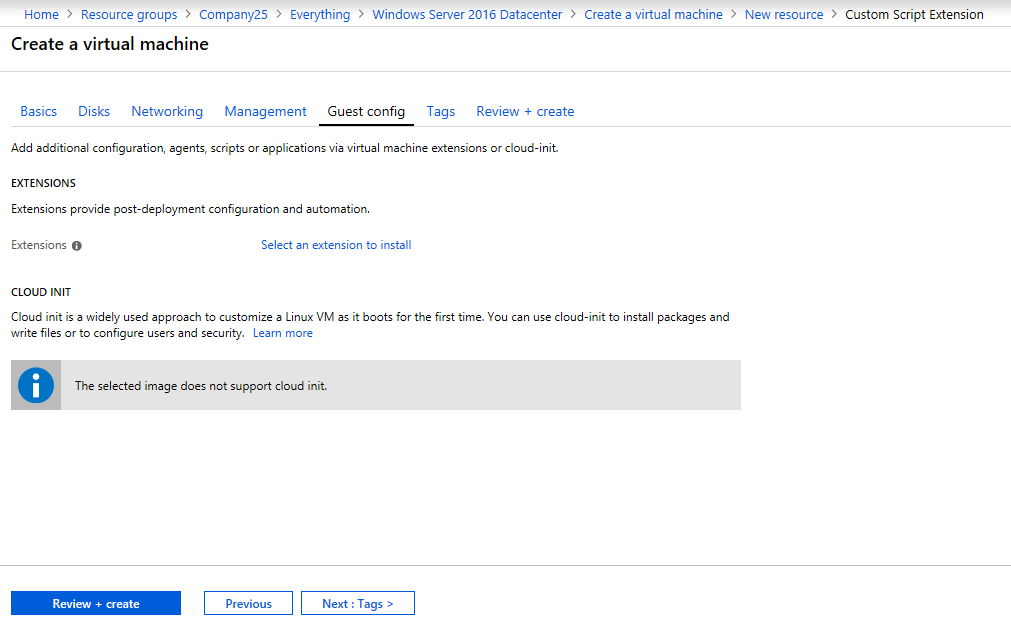
Within the *Create a virtual* blade, the required information is categorized under different tabs. Under *Basics*, *Network*, and *Management* tabs enter the required information. The next three screenshots show an example of values entered for these three tabs.



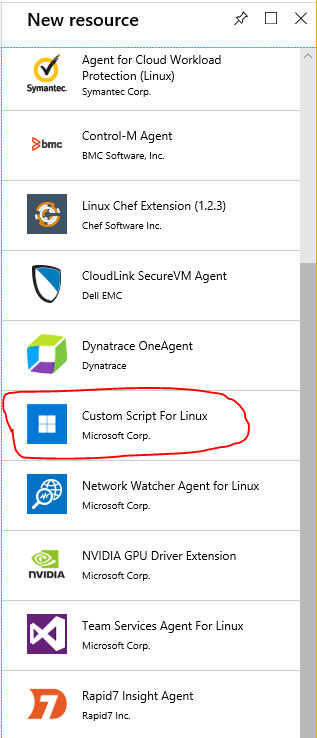




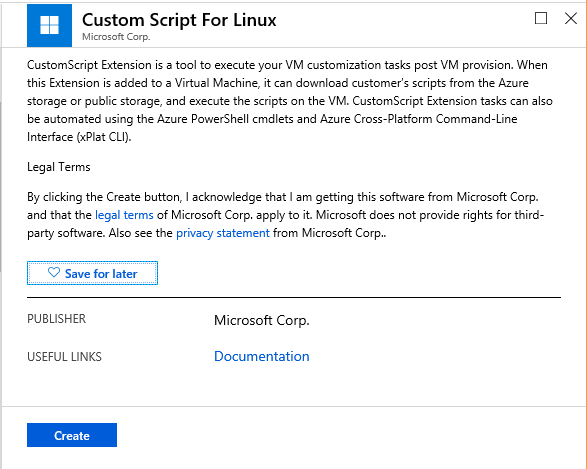
Under the *Guest config* tab, click on *Select an extension to install*.



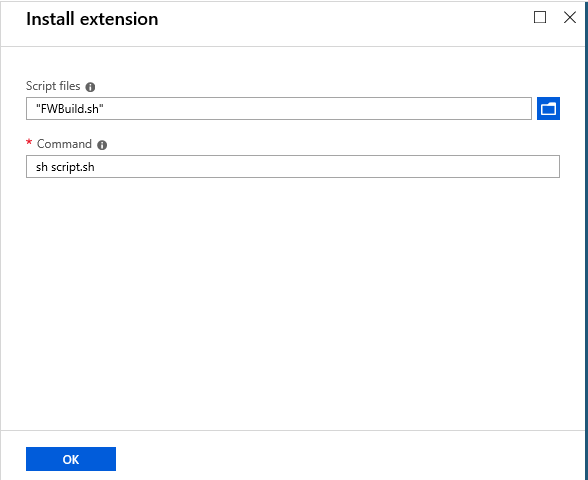
This will open the *New resource* pane:



Click on *Custom Script For Linux – Microsoft Corp*. This will open the *Custom Script Extension* introduction blade. Click  at the end of the blade.

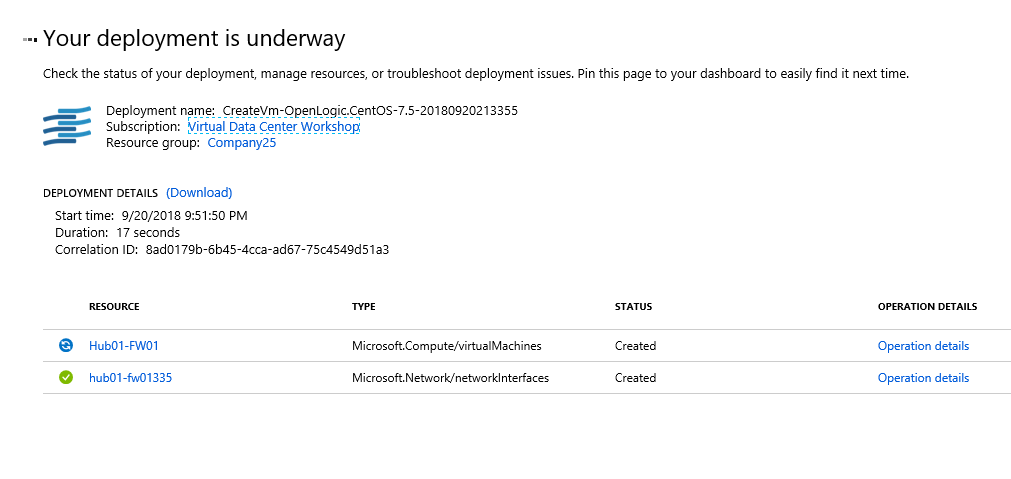


This will open the *Install extension* blade. Choose *FWBuild.sh* file in the Microsoft Azure flash drive given to you at the beginning of the workshop and click  at the bottom of the blade.



On the *Create a virtual machine* blade, click on . Review and ensure that the values are correctly entered. Click  at the bottom of the blade.

The deployment will take about 4 minutes. As the VM and its associated resources are being created, you will be presented with the deployment progress:



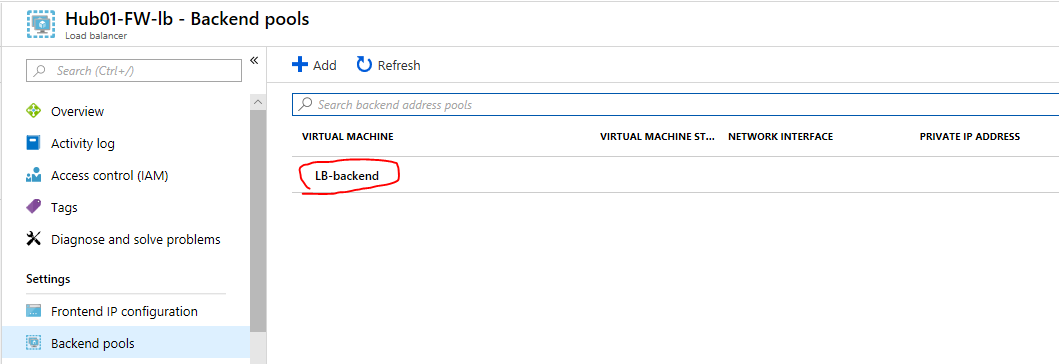
While the first VM is being deployed, you can configure the second VM that differs from the first VM only by the name.

Note: While creating the second VM, you must choose the VM size prior to choosing the Availability set.

## Link VMs to LB

Following successful deployment of both the VMs, link both the VMs to the backend pool of the load balancer we created in this step.

Click on the load balancer name (*Hub01-FW-lb*) in your base blade. On the *Load balancer* blade, click  on the left navigation pane. In the *Backend pools* pane click on the listed backend pool (*LB-backend*).



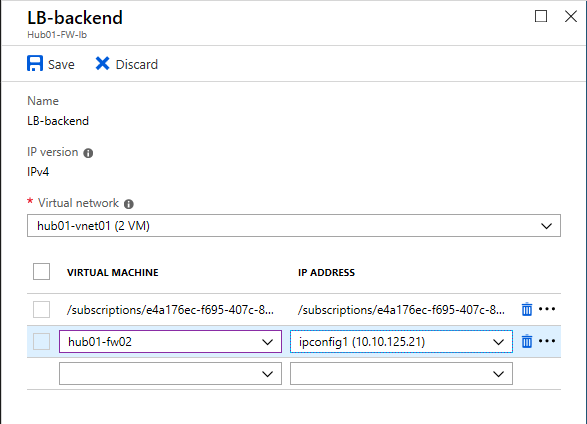
Configure the LB-backend blade with the following information:

Virtual network: hub01-vnet01 (2 VM)

VIRTUAL MACHINE IP ADDRESS

Hub01-fw01 **10.10.125**.*yy* (last octet would differ)

Hub01-fw02 **10.10.125.***zz*



Click  at the top of the blade.

## Create UDR rules to force traffic to the firewall

We need to create two route tables—one in Spoke01-VNet01 and the other in Spoke02-VNet01—for specifying *User Defined Routes.* We need *UDR* to force the traffic to the firewall in the hub.

Information we need to create the Route Tables:

Name: **Spoke01-VNet01-rt** / **Spoke02-VNet01-rt**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

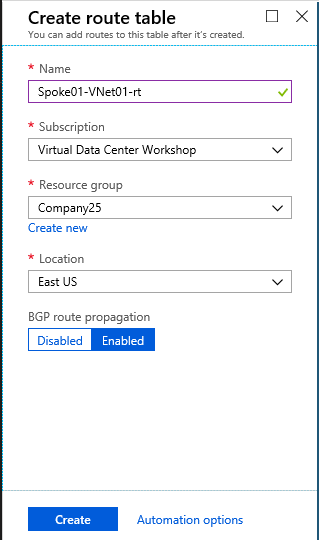
Region: **East US**

BGP route propagation: **Enabled**

On your base blade, click  at the top. On the *Everything* blade, start “typing route table” in the search bar at the top and choose *Route Table*. This will open the *Route table* introduction blade. Click at the end of the blade.



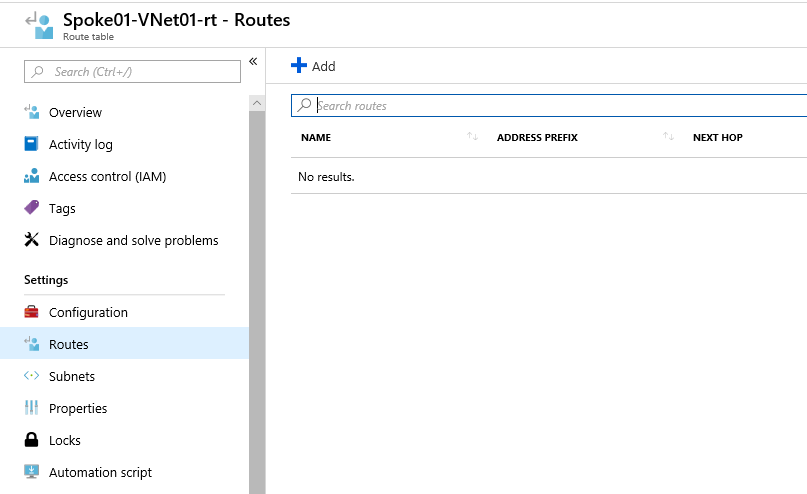
This will open the *Create route table* blade. Enter the required information on the blade and Click  at the end of the blade.



Following a quick validation, the route table will be created in about a minute.

### Adding Routes to the Route table

Open the *Route table* blade and click to open the *Routes* pane.



Information needed to add routes:

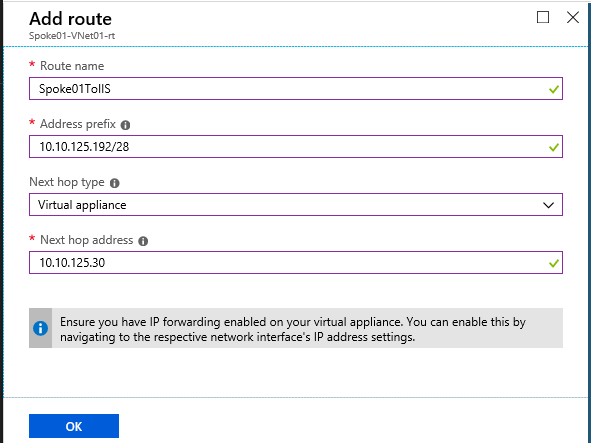
Route name: **Spoke01ToIIS** / **Spoke02ToFS**

Address prefix: **10.10.1***XX***.192/28** / **10.10.1***XX***.128/28**

Next hop type: VirtualAppliance

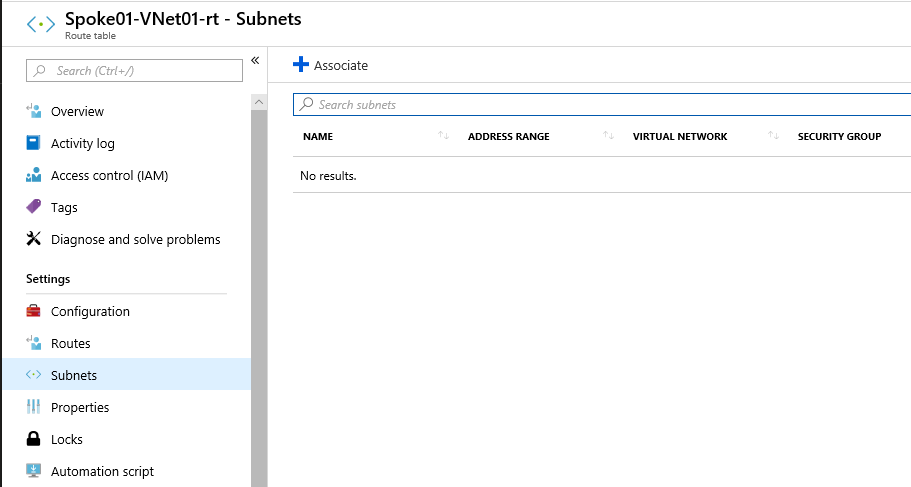
Next hop address: **10.10.1***XX***.30**

On the *Routes* pane, click on the top of the pane. This will open the *Add route* blade. Add the required information on the blade and click  at the bottom of the blade.



### Associating Subnets to Route table

On the *Route table* blade click to open the *Subnets* pane.

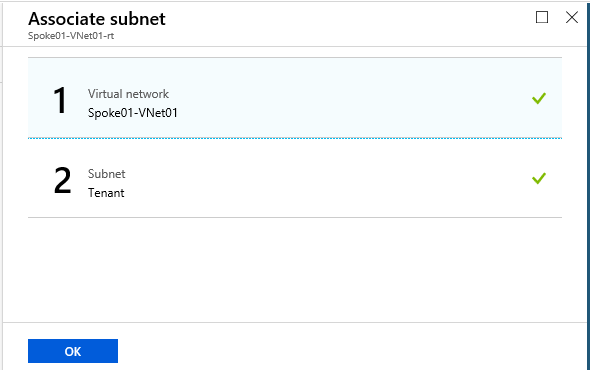


Information needed to associate subnets:

Virtual network: **Spoke01-VNet01** / **Spoke02-VNet01**

Subnet: **Tenant**

On the *Subnets* pane, click on the top of the pane. This will open the *Associate subnet* blade. Add the required information on the blade and click  at the bottom of the blade.



With the creation of the first route table, you are forcing the traffic from the Tenant subnet of *Spoke01-VNet01* to that of *Spoke02-VNet01* to go via the firewall in the hub. Similarly, create the second route table to force the traffic in the other direction to flow through the firewall as well.

## Validation

Find the App Gateway’s public IP address, from your on-premises VM, open the browser and browse to that IP address.

# Step 8: Create a Remote-VNet with Hub Connectivity

In this step, let’s create a new VNet in Europe with a VM. Let’s connect the remote VNet in Europe to the ExpressRoute circuit in Washington DC.

Specifically let’s to do the following:

* Create a Virtual Network
* Create an NSG
* Create an availability set for firewall
* Create virtual machines within an availability set and associate the NSG
* Link virtual machines to load balancer backend pool
* Create UDR rules to force traffic between Spoke01 and Spoke02 to go via the firewall

## Create a Virtual Network

To create a virtual network, we need the following information:

VNet Name: **Remote-VNet01**

Address space: **10.40.1***XX***.0/24**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX* (ex: Company25)

Location (Azure Region): **West Europe**

Subnet Name: **Tenant**

Subnet Address Space: **10.40.1***XX***.0/28**

Recall, you have configured VNets in *Step 1*, *Step 5*, and *Step 6*.

### Create a Gateway Subnet on the VNet

Let’s add a Gateway subnet, with the following information:

Subnet Name: **GatewaySubnet**

Address Range: **10.40.1***XX***.96/27**

## Create an ExpressRoute Gateway

Information we need to create the ExpressRoute gateway:

Name: **Remote-VNet01-gw**

Gateway type: **ExpressRoute**

SKU: **Standard**

Virtual network: **Remote-VNet01**

Public IP address: **Create new**

**Remote-VNet01-gw-ip** (name of the public IP address)

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX* (ex: Company25)

Location: **West Europe**

Recall, you have configured an ExpressRoute Gateway in Step 3.

## Create a Network Security Group

Information we need to create an NSG.

Name: **Remote-VM01-nic-nsg**

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Location: **West Europe**

After deploying the NSG, add an inbound security rule on the NSG.

Information needed to create an inbound security rule to allow RDP traffic:

Source: **Any**

Source port ranges: **\***

Destination: **Any**

Destination port ranges: **3389**

Protocol: **TCP**

Action: **Allow**

Priority: **1000**

Name: **myNSGRuleRDP**

Description: To permit RDP from any source

## Create a Virtual Machine

Let’s create a virtual machine and associate the NSG that we just created. Information we need to create the virtual machine.

Basics

Subscription: **Virtual Data Center Workshop**

Resource Group: **Company***XX*

Virtual machine name: **Remote-VM01**

Region: **West Europe**

Image: **Windows Server 2016 Datacenter**

Size: **Standard A4 v2**

Username: **Student***XXX*

Password: (if not noted, see *Accessing Azure Key Vault*, at the beginning of this document).

Networking

Virtual network: **Remote-VNet01**

Subnet: **Tenant**

Public IP: **Remote-VM01-nic-ip**

Network security group: **Advanced**

Configure network security group: **Remote-VM01-nic-nsg**

Management

Boot diagnostics: **Off**

Guest config

We are going to use *Custom Script Extension* to allow ICMP traffic.

For script file, choose AllowICMPv4.ps1 file.

Note that the deployment will take about 8 minutes. In the meantime, proceed further and create a connection object.

## Create a Connection Object

Let’s create a connection object to connect the VNet GW of the remote VNet to the ExpressRoute circuit we created in *Step 2*. Information we need to create the connection object:

Name: **Remote-VNet01-gw-conn**

Connection type: **ExpressRoute**

Virtual network gateway: **Remote-VNet01**

ExpressRoute circuit: **Company***XX***-er**

Subscription: **Virtual Data Center Workshop**

Resource group: **Company***XX*

Location: **West Europe**

## Validation

From the on-prem VM and from the Hub VM you should be able to ping the private IP address of the VM that we created in the Remote VNet