

Lab 06 - Implement Traffic Management

Task 1: Provision the lab environment

In this task, we will deploy four virtual machines into the same Azure region. The first two will reside in a hub virtual network, while each of the remaining two will reside in a separate spoke virtual network.

Making sure I've uploaded the files \\Allfiles\\Labs\\06\\az104-06-vms-loop-template.json and \\Allfiles\\Labs\\06\\az104-06-vms-loop-parameters.json into the Cloud Shell home directory:

```
PS /home/andrijana> ls -l
total 36
-rw-r--r-- 1 andrijana andrijana 447 Mar 23 14:38 az104-02a-customRoleDefinition.json
-rw-r--r-- 1 andrijana andrijana 373 Mar 23 22:38 az104-05-vnetvm-loop-parameters.json
-rw-r--r-- 1 andrijana andrijana 7842 Mar 23 22:37 az104-05-vnetvm-loop-template.json
-rw-r--r-- 1 andrijana andrijana 388 Mar 23 23:27 az104-06-vms-loop-parameters.json
-rw-r--r-- 1 andrijana andrijana 9867 Mar 23 23:27 az104-06-vms-loop-template.json
lrwxrwxrwx 1 andrijana andrijana 22 Mar 23 21:32 clouddrive -> /usr/csuser/clouddrive
drwxr-xr-x 3 andrijana andrijana 4096 Mar 20 22:09 Microsoft
```

Creating the first resource group:

```
PS /home/andrijana> $location = 'eastus'
PS /home/andrijana> $rgName = 'az104-06-rg1'
PS /home/andrijana> New-AzResourceGroup -Name $rgName -Location $location

ResourceGroupName : az104-06-rg1
Location           : eastus
ProvisioningState  : Succeeded
Tags               :
ResourceId         : /subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1
```

From the Cloud Shell pane, run the following to create the three virtual networks and four Azure VMs into them by using the template and parameter files you uploaded:

```
New-AzResourceGroupDeployment `
    -ResourceGroupName $rgName `
    -TemplateFile $HOME/az104-06-vms-loop-template.json `
```

```
-TemplateParameterFile $HOME/az104-06-vm-loop-parameters.json
```

```
PS /home/andrijana> New-AzResourceGroupDeployment `
>> -ResourceGroupName $rgName `
>> -TemplateFile $HOME/az104-06-vm-loop-template.json `
>> -TemplateParameterFile $HOME/az104-06-vm-loop-parameters.json

DeploymentName      : az104-06-vm-loop-template
ResourceGroupName   : az104-06-rg1
ProvisioningState    : Succeeded
Timestamp           : 3/23/2023 11:34:27 PM
Mode                : Incremental
TemplateLink         :
Parameters           :
                        Name                Type                Value
                        =====
                        vmSize              String              "Standard_D2s_v3"
                        vmName              String              "az104-06-vm"
                        vmCount              Int                 4
                        adminUsername        String              "Student"
                        adminPassword        SecureString         null

Outputs             :
DeploymentDebugLogLevel :
```

Next, we install the Network Watcher extension on the Azure VMs deployed in the previous step by running the following command in PowerShell:

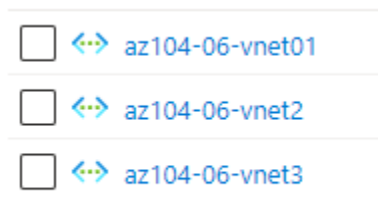
```
PS /home/andrijana> $rgName = 'az104-06-rg1'
PS /home/andrijana> $location = (Get-AzResourceGroup -ResourceGroupName $rgName).location
PS /home/andrijana> $vmNames = (Get-AzVM -ResourceGroupName $rgName).Name
PS /home/andrijana>
PS /home/andrijana> foreach ($vmName in $vmNames) {
>> Set-AzVMExtension `
>> -ResourceGroupName $rgName `
>> -Location $location `
>> -VMName $vmName `
>> -Name 'networkWatcherAgent' `
>> -Publisher 'Microsoft.Azure.NetworkWatcher' `
>> -Type 'NetworkWatcherAgentWindows' `
>> -TypeHandlerVersion '1.4'
>> }

RequestId IsSuccessStatusCode StatusCode ReasonPhrase
-----
True      OK OK
True      OK OK
True      OK OK
True      OK OK
```

Task 2: Configure the hub and spoke network topology

In this task, we will configure local peering between the virtual networks you deployed in the previous tasks in order to create a hub and spoke network topology.

We've created 3 Virtual Networks:



Record the Resource ID of the second virtual network:

```
/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1/providers/Microsoft.Network/virtualNetworks/az104-06-vnet2
```

Next, we add Peerings to vnet01:

[Home](#) > [Virtual networks](#) > [az104-06-vnet01 | Peerings](#) >

Add peering ...

az104-06-vnet01

This virtual network

Peering link name *

az104-06-vnet01_to_az104-06-vnet2 ✓

Traffic to remote virtual network ⓘ

- ☒ Allow (default)
- ☐ Block all traffic to the remote virtual network

Traffic forwarded from remote virtual network ⓘ

- ☐ Allow (default)
- ☒ Block traffic that originates from outside the remote virtual network

Virtual network gateway or Route Server ⓘ

- ☐ Use this virtual network's gateway or Route Server
- ☐ Use the remote virtual network's gateway or Route Server
- ☒ None (default)

Remote virtual network

Peering link name *

az104-06-vnet2_to_az104-06-vnet01 ✓

Virtual network deployment model ⓘ

- ☒ Resource manager
- ☐ Classic

☒ I know my resource ID ⓘ

Resource ID *

/subscriptions/836f56df-cca0-4866-b552-adbe26a742da/resourceGroups/az104-06-rg1/providers/Microsoft.Network/vir... ✓

Traffic to remote virtual network ⓘ

- ☒ Allow (default)
- ☐ Block all traffic to the remote virtual network

Traffic forwarded from remote virtual network ⓘ

- ☒ Allow (default)
- ☐ Block traffic that originates from outside the remote virtual network

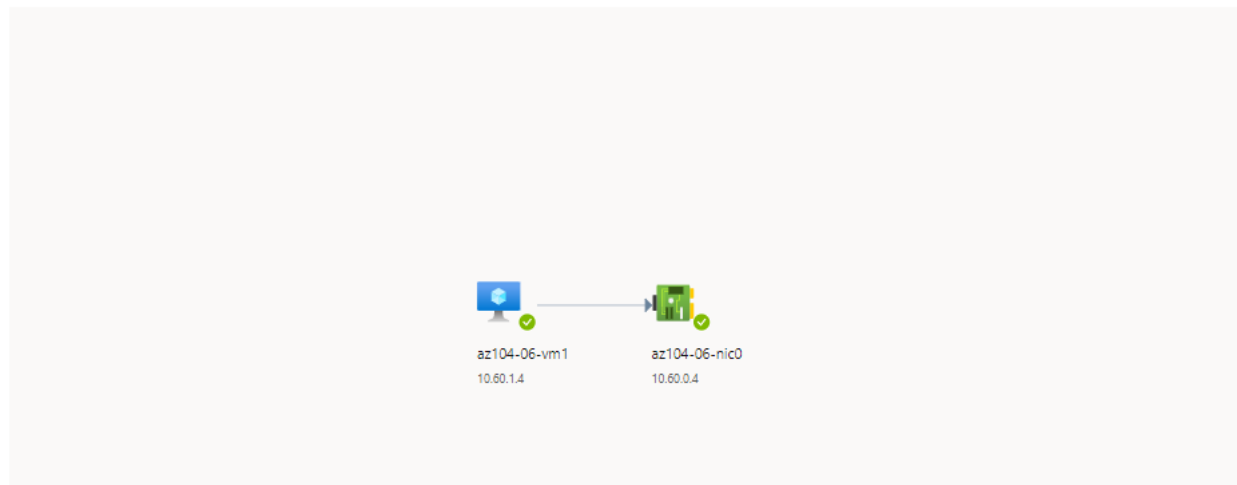
This step establishes two local peerings - one from az104-06-vnet01 to **az104-06-vnet2** and the other from **az104-06-vnet2** to **az104-06-vnet01**.

Test	Status	Details	Suggestions
Connectivity Test	❌ Fail	Probes Sent: 0 , Probes Failed: 0	-
NSG Outbound (from source)	✅ Success	Outbound communication from source is allowed	None
Next Hop (from source)	✅ Success	Next Hop Type: VirtualNetwork Route Table Id: System Route	None

Hop by hop details

Name	Status	IP address	Next hop	RTT	Errors
 az104-06-vm1	🔍 Info	10.60.1.4	10.60.0.4	-	-
 az104-06-nic0	🔍 Info	10.60.0.4	-	-	-

Topology view





In the topology view, we can see the virtual networks are peered with each other.

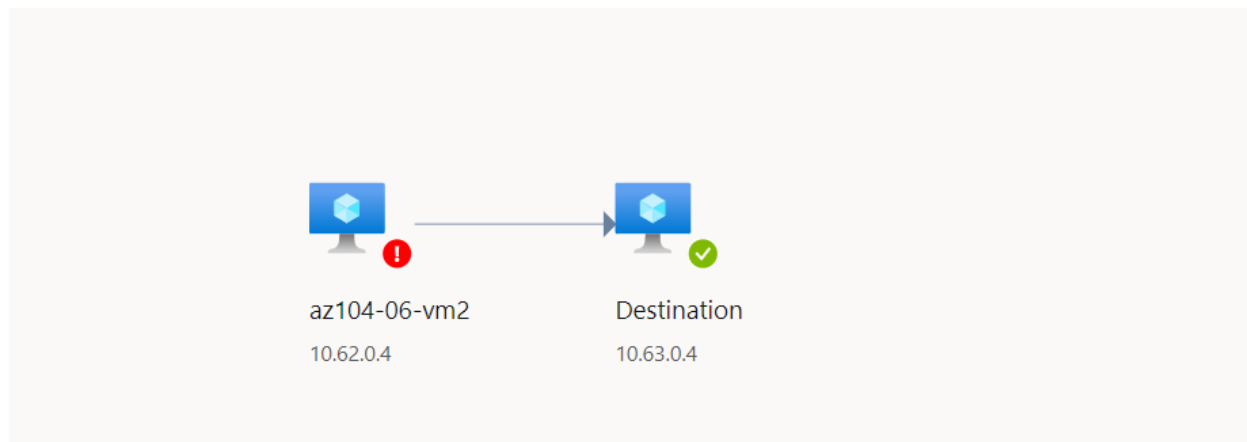
We initialize a Connection troubleshooting from **az104-06-vm2** to 10.63.0.4, which belongs to **az104-06-vm3**. We run the Diagnostics test:

Test	Status	Details	Suggestions
Connectivity Test	❌ Fail	Probes Sent: 0 ,Probes Failed: 0	-
NSG Outbound (from source)	❌ Fail	There are failed tests in the following NSGs: <ul style="list-style-type: none"> az104-06-nsg2 	Go to VM > Update the networking rule Read docs
Next Hop (from source)	✅ Success	Next Hop Type: None Route Table Id: System Route	None

Hop by hop details

Name	Status	IP address	Next hop	RTT	Errors
 az104-06-vm2	❌ Fail	10.62.0.4	10.63.0.4	-	Routes for the destination are missing in the virtual network gateway.
 Destination (10.63.0.4)	ℹ Info	10.63.0.4	-	-	-

Topology view



The test fails because the two virtual networks are not peered with each other. Virtual peering is not transitive, and this is an expected behavior when running the diagnostics test. Note that the status is Fail.

10.62.0.4 represents the private IP address of az104-06-vm2

Task 4: Configure routing in the hub and spoke topology

In this task, we will configure and test routing between the two spoke virtual networks by enabling IP forwarding on the network interface of the az104-06-vm0 virtual machine, enabling routing within its operating system, and configuring user-defined routes on the spoke virtual network.

Set IP forwarding to Enabled:

IP forwarding settings

IP forwarding

Disabled **Enabled**

Virtual network


az104-06-vnet01

Next, we need to configure an operating system for the **az104-06-vm0** Virtual Machine by running the following command in Overview -> Run Command -> RunPowerShellScript:

```
Install-WindowsFeature RemoteAccess -IncludeManagementTools
```


Run Command Script

RunPowerShellScript

 Script execution complete

PowerShell Script

```
1 Install-WindowsFeature RemoteAccess -IncludeManagementTools
```

Run

Output


Success	Restart Needed	Exit Code	Feature Result
True	No	Success	{Remote Access}

We need to configure the operating system because this virtual machine will support routing.

Install the routing role service:

Run Command Script

RunPowerShellScript

 Script execution complete

PowerShell Script

```
1  Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFe
2
3  Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"
4
5  Install-RemoteAccess -VpnType RoutingOnly
6
7  Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled
```

Run

Output

Success	Restart Needed	Exit Code	Feature Result
True	No	Success	{RAS Connection Manager Administration Kit...
True	No	NoChangeNeeded	{}

Create route tables:

[Home](#) / [Route tables](#) /

Create Route table ...

Basics Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ

Azure Pass - Sponsorship



Resource group * ⓘ

az104-06-rg1

[Create new](#)

Instance details

Region * ⓘ

East US

Name * ⓘ

az104-06-rt23

Propagate gateway routes * ⓘ

☐ Yes
☒ No



Microsoft.RouteTable-20230324014743 | Overview

Deployment

Search

[Delete](#) [Cancel](#) [Redeploy](#) [Download](#) [Refresh](#)

Overview

[Inputs](#)

[Outputs](#)

[Template](#)

✓ Your deployment is complete



Deployment name: Microsoft.RouteTable-20230324014743
Subscription: [Azure Pass - Sponsorship](#)
Resource group: [az104-06-rg1](#)

Start time: 3/24/2023, 1:48:54 AM

Correlation ID: 1ce3c261-5078-4b25-8b69-6455cf4efc8a [Copy](#)

Deployment details

Next steps

[Go to resource](#)

Give feedback

[Tell us about your experience with deployment](#)

After the deployment is finished, we click Go to resource where we add a route with the following settings:

Add route

×

az104-06-rt23

Route name *

az104-06-route-vnet2-to-vnet3 ✓

Destination address prefix * ⓘ

IP Addresses ✓

Destination IP addresses/CIDR ranges * ⓘ


10.63.0.0/20 ✓

Next hop type * ⓘ

Virtual appliance ✓

Next hop address * ⓘ

10.60.0.4 ✓

 Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.

Then, we add an associate subnet:

Associate subnet

×

az104-06-rt23

Virtual network * ⓘ

az104-06-vnet2 (az104-06-rg1) ✓

Subnet * ⓘ

subnet0 ✓

We created another route table:

Create Route table ...

Basics Tags Review + create

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ Azure Pass - Sponsorship ▼

Resource group * ⓘ az104-06-rg1 ▼

[Create new](#)

Instance details

Region * ⓘ East US ▼

Name * ⓘ az104-06-rt32 ✓

Propagate gateway routes * ⓘ ☐ Yes ☒ No

Microsoft.RouteTable-20230324015331 | Overview ⓘ ...

Deployment

Search << [Delete] [Cancel] [Redeploy] [Download] [Refresh]

Overview

Inputs

Outputs

Template

✓ Your deployment is complete

Deployment name: Microsoft.RouteTable-20230324015331
Subscription: Azure Pass - Sponsorship
Resource group: az104-06-rg1

Start time: 3/24/2023, 1:54:38 AM
Correlation ID: 7801b183-2885-45a0-9e9b-b4b9ebaf83ad

Deployment details

Next steps

[Go to resource](#)

Then, we create another route from the third virtual machine to the first virtual machine:

Add route



az104-06-rt32

Route name *

az104-06-route-vnet3-to-vnet2



Destination address prefix * ⓘ

IP Addresses



Destination IP addresses/CIDR ranges * ⓘ

10.62.0.0/20



Next hop type * ⓘ

Virtual appliance



Next hop address * ⓘ

10.60.0.4



Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.

And we add an associate subnet:

Associate subnet



az104-06-rt32

Virtual network * ⓘ

az104-06-vnet3 (az104-06-rg1)



Subnet * ⓘ

subnet0



We run diagnostics test with the following settings:

| Connection troubleshoot ...

« Network Watcher connection troubleshoot provides the capability to check a direct TCP or ICMP connection from a virtual machine (VM), application gateway, or Bastion host to a VM, fully qualified domain name (FQDN), URI, or IP address. To start, choose a source to start the connection from, and the destination you wish to connect to and select "Run diagnostic tests". [Learn more](#)

Source

Subscription * ⓘ Azure Pass - Sponsorship ✓

Resource group * ⓘ az104-06-rg1 ✓

Source type * ⓘ Virtual machine ✓

Virtual machine * ⓘ az104-06-vm2 ✓

Destination

Destination type ⓘ ☐ Select a virtual machine ☒ Specify manually

URI, FQDN or IP address * ⓘ 10.63.0.4 ✓

Probe settings

Protocol ⓘ ☒ TCP ☐ ICMP

Destination port * ⓘ 3389 ✓

Source port (optional) ⓘ ✓

Connection diagnostic

Diagnostics tests * ⓘ 4 selected ✓

Run diagnostic tests




So, it's basically a test from the third virtual machine to the fourth one.

Click Check and wait until results of the connectivity check are returned. Verify that the status is Reachable. Review the network path and note that the traffic was routed via 10.60.0.4, assigned to the az104-06-nic0 network adapter. If status is Unreachable, we should stop and then start az104-06-vm0.

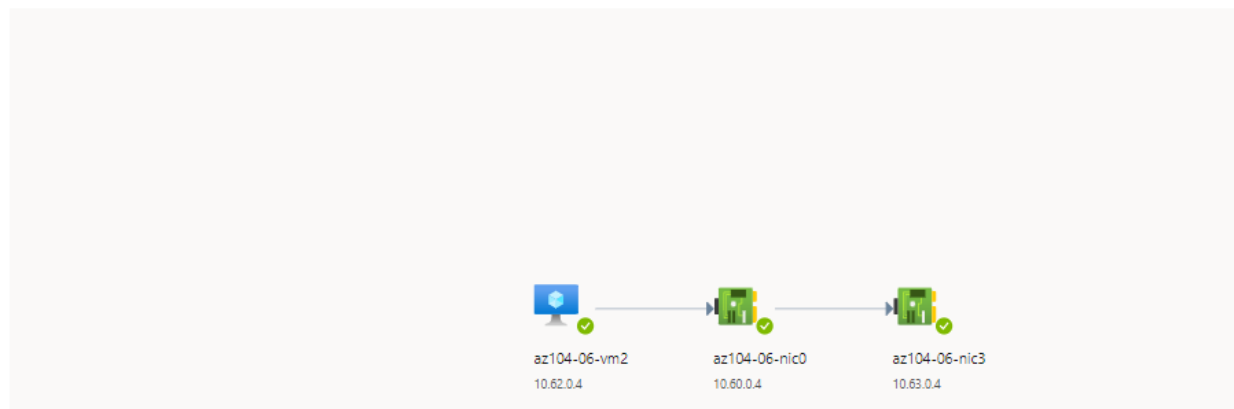
Diagnostic tests

Test	Status	Details	Suggestions
Connectivity Test	✖ Fail	Probes Sent: 0 , Probes Failed: 0	-
NSG Outbound (from source)	✔ Success	Outbound communication from source is allowed	None
Next Hop (from source)	✔ Success	Next Hop Type: VirtualAppliance Next Hop IP: 10.60.0.4	None

Hop by hop details

Name	Status	IP address	Next hop	RTT	Errors
 az104-06-vm2	ℹ Info	10.62.0.4	10.60.0.4	-	-
 az104-06-nic0	ℹ Info	10.60.0.4	10.63.0.4	-	-
 az104-06-nic3	ℹ Info	10.63.0.4	-	-	-

Topology view



The connection is established.

Task 5: Implement Azure Load Balancer

Create load balancer ...

Basics Frontend IP configuration Backend pools Inbound rules Outbound rules Tags Review + create

Azure load balancer is a layer 4 load balancer that distributes incoming traffic among healthy virtual machine instances. Load balancers uses a hash-based distribution algorithm. By default, it uses a 5-tuple (source IP, source port, destination IP, destination port, protocol type) hash to map traffic to available servers. Load balancers can either be internet-facing where it is accessible via public IP addresses, or internal where it is only accessible from a virtual network. Azure load balancers also support Network Address Translation (NAT) to route traffic between public and private IP addresses. [Learn more.](#)

Project details

Subscription *

Resource group *
[Create new](#)

Instance details

Name *

Region *

SKU * ⓘ
☒ Standard
☐ Gateway
☐ Basic

i Microsoft recommends Standard SKU load balancer for production workloads. [Learn more about pricing differences between Standard and Basic SKU](#)

Type * ⓘ
☐ Public
☒ Internal

Tier *
☒ Regional
☐ Global

Type * ⓘ

☒ Public

☐ Internal

Type should be Public (I've set this as internal previously - my mistake, and got some errors)

We add the backend pools:

BasicsFrontend IP configurationBackend poolsInbound rulesOutbound rulesTagsReview + create

A backend pool is a collection of resources to which your load balancer can send traffic. A backend pool can contain virtual machines, virtual mach

+ Add a backend pool

Name	Virtual network	Resource Name
az104-06-lb4-be1		
az104-06-lb4-be1	az104-06-vnet01	az104-06-vm0
az104-06-lb4-be1	az104-06-vnet01	az104-06-vm1

Adding a health probe:

Add health probe



Health probes are used to check the status of a backend pool instance. If the health probe fails to get a response from a backend instance then no new connections will be sent to that backend instance until the health probe succeeds again.

Name *

az104-06-lb4-hp1



Protocol *

TCP



Port * ⓘ

80

Interval * ⓘ

5

seconds

Used by ⓘ

Not used






OK

Cancel

We create the Load Balancer:

Microsoft.LoadBalancer-20230324020324 | Overview

Deployment


<<  Delete  Cancel  Redeploy  Download  Refresh


Overview


Inputs


Outputs

Template

 **Your deployment is complete**

 Deployment name: Microsoft.LoadBalancer-20230324020324 St
Subscription: [Azure Pass - Sponsorship](#) C
Resource group: [az104-06-rg4](#)

 Deployment details

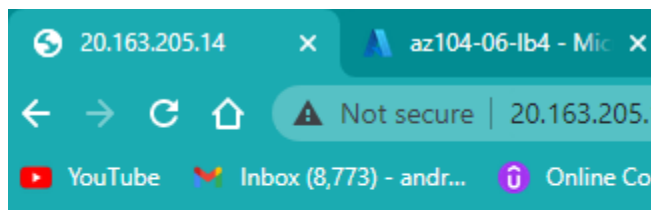
 Next steps

[Go to resource](#)

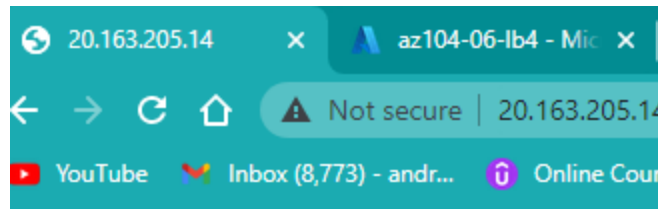
[Give feedback](#)

Select Frontend IP configuration from the Load Balancer resource page. Copy the IP address.

Then we open another browser tab and navigate to the IP address. Verify that the browser window displays the message Hello World from az104-06-vm0 or Hello World from az104-06-vm1.



Hello World from az104-06-vm1



Hello World from az104-06-vm0

This demonstrates the load balancer rotating through the virtual machines.

Task 6: Implement Azure Application Gateway

In this task, we will implement an Azure Application Gateway in front of the two Azure virtual machines in the spoke virtual networks.

Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications. Traditional load balancers operate at the transport layer (OSI layer 4 - TCP and UDP) and route traffic based on source IP address and port, to a destination IP address and port.

Add a subnet to az104-06-vnet01:

Add subnet

Name *

subnet-appgw

Subnet address range * ⓘ

10.60.3.224/27

10.60.3.224 - 10.60.3.255 (27 + 5 Azure reserved addresses)

This subnet will be used by the Azure Application Gateway instances, which you will deploy later in this task. The Application Gateway requires a dedicated subnet of /27 or larger size.

Then, we add a backend pool.

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, app services, IP addresses, or fully qualified domain names (FQDN).

Add a backend pool.

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machines scale sets, IP addresses, domain names, or an App Service.

Name *

az104-06-appgw5-be1

Add backend pool without targets

Yes

No

Backend targets

2 items

Target type	Target	
IP address or FQDN	10.62.0.4	**
IP address or FQDN	10.63.0.4	**
IP address or FQDN		

And we add a routing rule so we can send traffic from the given IP address to one or more backend targets:

[illegible]

As the final step, we deploy the Application Gateway.