Networking
Lecture 11
Deep.Azure@McKesson

Zoran B. Djordjević

Azure Networking

Azure provides a variety of networking capabilities:

- Connectivity between Azure resources: Connect Azure resources together in a secure, private virtual network in the cloud.
- Internet connectivity: Communicate to and from Azure resources over the Internet.
- On-premises connectivity: Connect an on-premises network to Azure resources through a virtual private network (VPN) over the Internet, or through a dedicated connection to Azure.
- Load balancing and traffic direction: Load balance traffic to servers in the same location and direct traffic to servers in different locations.
- Security: Filter network traffic between network subnets or individual virtual machines (VM).
- Routing: Use default routing or fully control routing between your Azure and on-premises resources.
- Manageability: Monitor and manage your Azure networking resources.
- Deployment and configuration tools: Use a web-based portal or cross-platform command-line tools to deploy and configure network resources.

Connectivity between Azure Resources

- Azure resources such as Virtual Machines, Cloud Services, Virtual Machines Scale Sets, and Azure App Service Environments can communicate privately with each other through an Azure Virtual Network (VNet).
- A VNet is a logical isolation of the Azure cloud dedicated to your subscription. You can implement multiple VNets within each Azure subscription and Azure region. Each VNet is isolated from other VNets.
- For each VNet you can:
  - Specify a custom private IP address space using public and private addresses.
  - Segment the VNet into one or more subnets. VNet address space is allocated to each subnet.
  - Use Azure-provided name resolution or your own DNS server for resources connected to a VNet.
- You can connect VNets to each other, enabling resources connected to either VNet to communicate with each other across VNets.
- You can use either or both of the following options to connect VNets to each other:
  - **Peering:** Enables resources connected to different Azure VNets within the same Azure region to communicate with each other.
  - **VPN Gateway:** Enables resources connected to different Azure VNets within different Azure regions to communicate with each other. Traffic between VNets flows through an Azure VPN Gateway.

Internet Connectivity

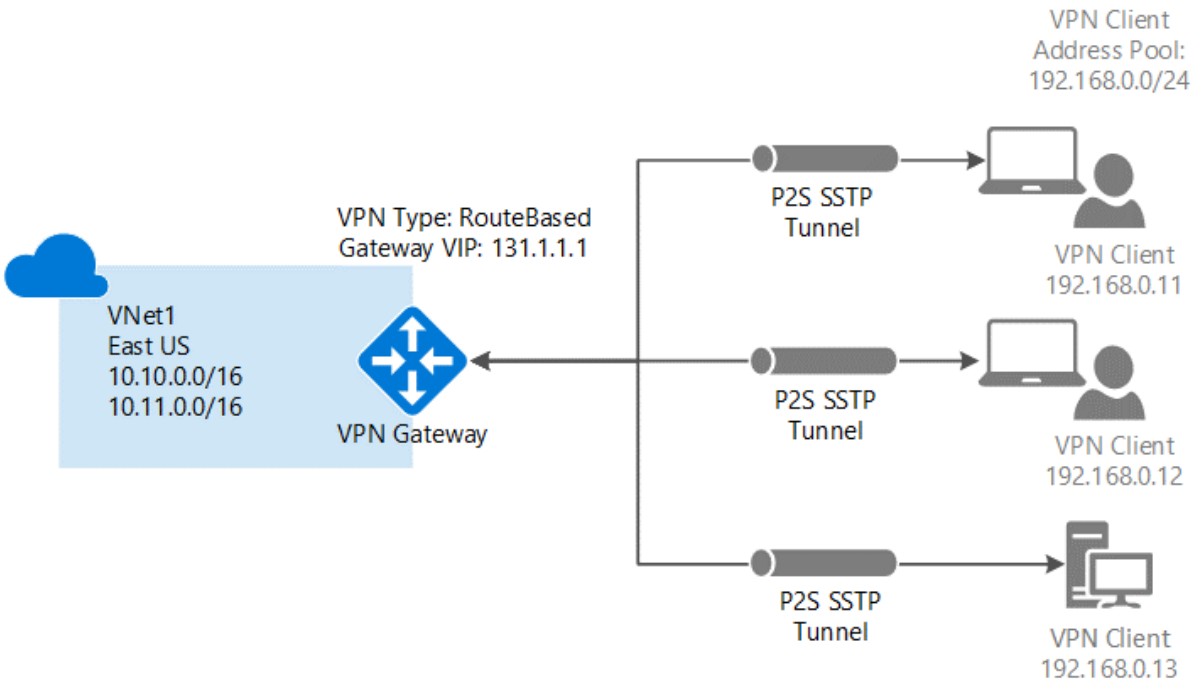
- All Azure resources connected to a VNet have outbound connectivity to the Internet by default.
- The private IP address of the resource is source network address translated (SNAT) to a public IP address by the Azure infrastructure.
- To communicate inbound to Azure resources from the Internet, or to communicate outbound to the Internet without SNAT, a resource must be assigned a public IP address.

Connectivity to on-premise Resources

- You can access resources in your VNet securely over either a VPN connection, or a direct private connection.
- To send network traffic between your Azure virtual network and your on-premises network, you must create a virtual network gateway.
- Virtual network gateway can support:
  - Point-to-site VPN
  - Site-to-site VPN or
  - ExpressRoute.

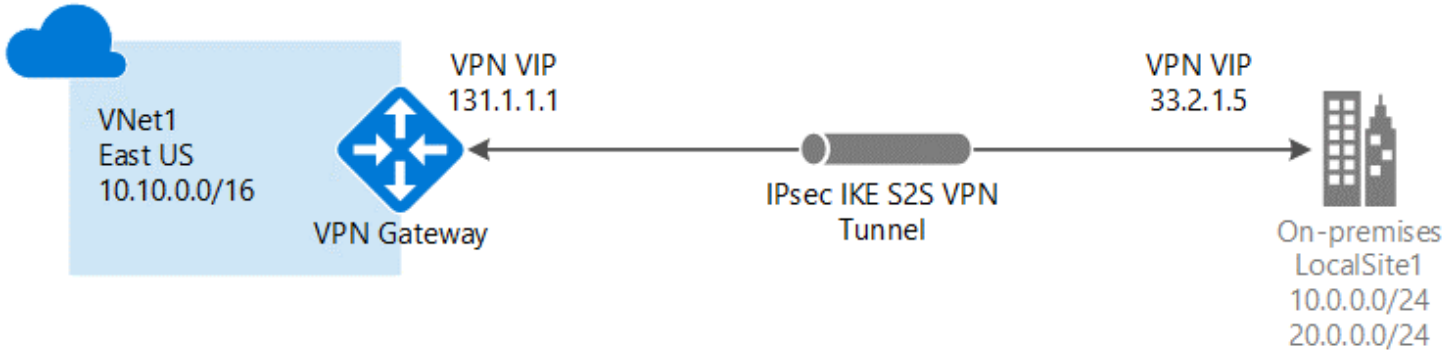
Point-to-site (VPN over SSTP)

- Point-to-site connection is established between a single computer and a VNet.
- Point-to-site connections are often coupled with a site-to-site connection through the same virtual network gateway.
- The connection uses **Secure Socket Tunneling Protocol** (**SSTP**) protocol that provides a mechanism to transport point-to-point traffic through an SSL/TLS channel. SSL/TLS provides transport-level security with key negotiation, encryption and traffic integrity checking.



Site-to-site (IPsec/IKE VPN tunnel)

- Site-to-site connection is established between your on-premises VPN device and an Azure VPN Gateway.
- This connection type enables any on-premises resource that you authorize to access the VNet. The connection is an IPSec/IKE VPN that provides encrypted communication over the Internet between your on-premises device and the Azure VPN gateway.
- You can connect multiple on-premises sites to the same VPN gateway.
- The on-premises VPN device at each site must have an externally-facing public IP address that is not behind a NAT.

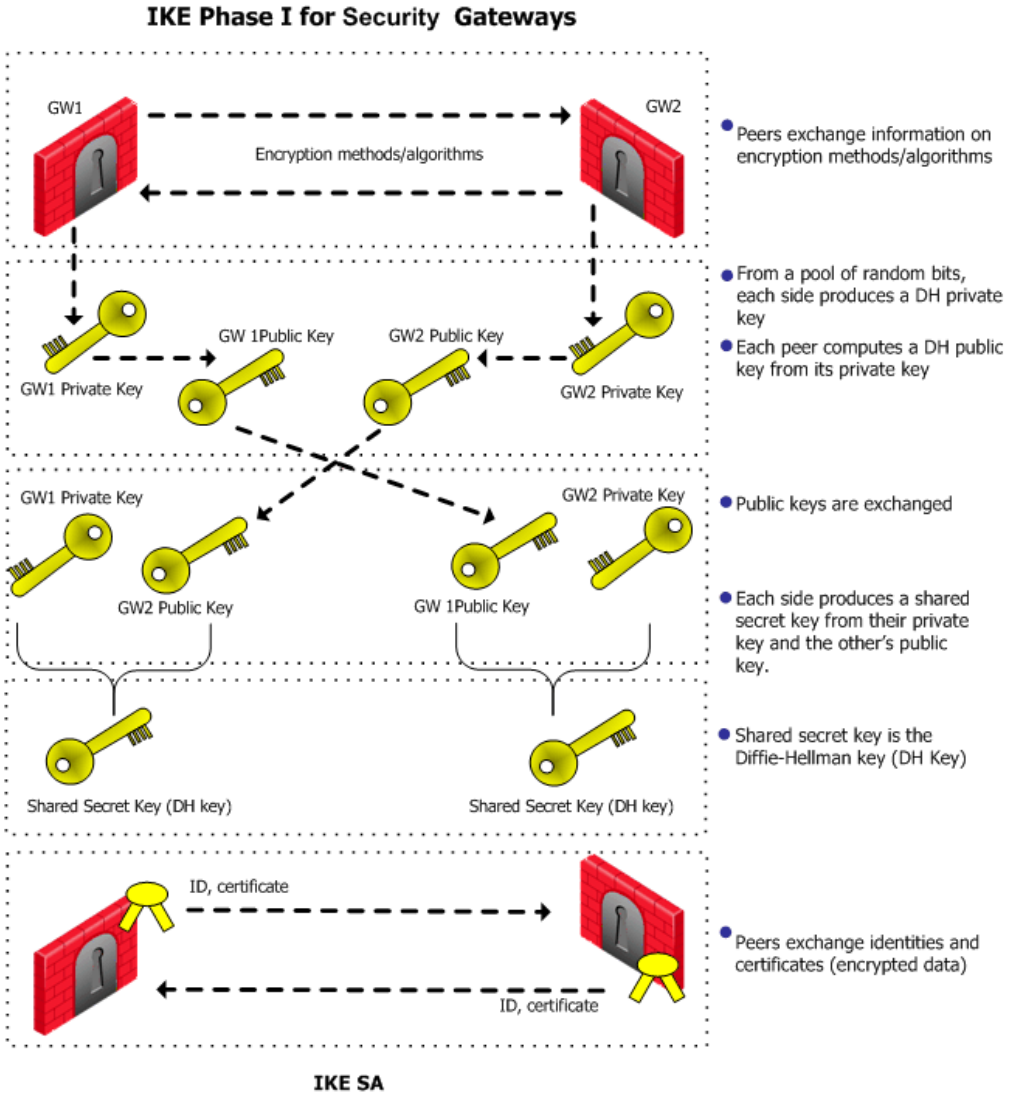


IPsec & IKE

- In symmetric cryptographic systems, both communicating parties use the same key for encryption and decryption. The material used to build these keys must be exchanged in a secure fashion. Information can be securely exchanged only if the key belongs exclusively to the communicating parties.
- The goal of the *Internet Key Exchange* (IKE) is for both sides to independently produce the same symmetrical key. This key then encrypts and decrypts the regular IP packets used in the bulk transfer of data between VPN peers. IKE builds the VPN tunnel by authenticating both sides and reaching an agreement on methods of encryption and integrity. The outcome of an IKE negotiation is a *Security Association* (SA).
- This agreement upon keys and methods of encryption must also be performed securely. For this reason, IKE is composed of two phases. The first phase lays the foundations for the second. Both IKEv1 and IKEv2 are supported in Security Gateways of version R71 and higher.
- Diffie-Hellman (DH) is that part of the IKE protocol used for exchanging the material from which the symmetrical keys are built. The Diffie-Hellman algorithm builds an encryption key known as a "shared secret" from the private key of one party and the public key of the other. Since the IPsec symmetrical keys are derived from this DH key shared between the peers, at no point are symmetric keys actually exchanged.

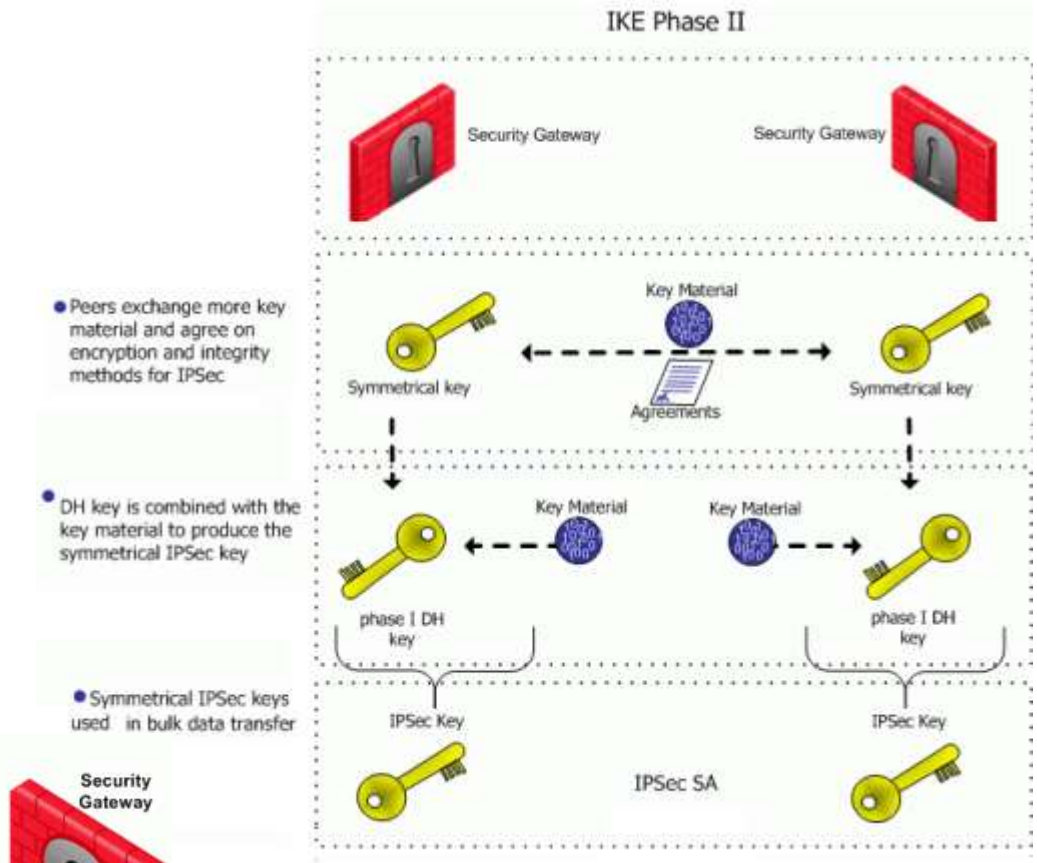
IKE Phase I

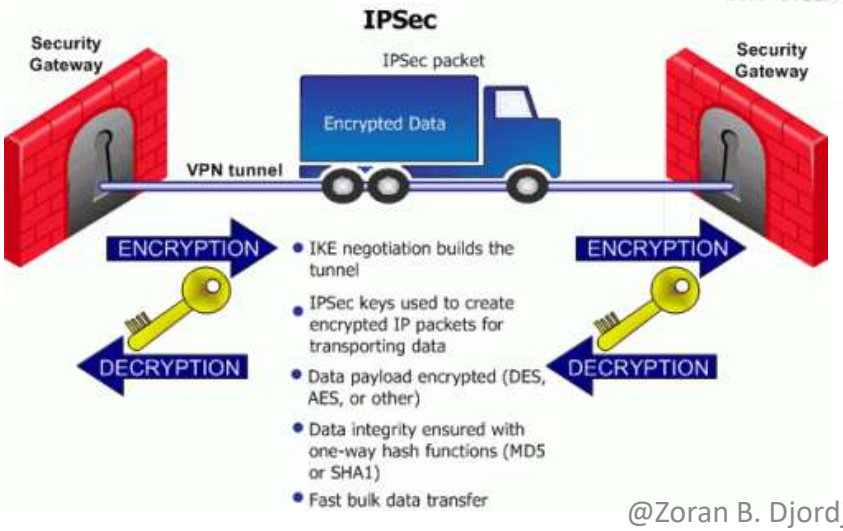
- During IKE Phase I:
- The peers authenticate, either by certificates or via a pre-shared secret. (More authentication methods are available when one of the peers is a remote access client.)
- A Diffie-Hellman key is created. The nature of the Diffie-Hellman protocol means that both sides can independently create the shared secret, a key which is known only to the peers.
- Key material (random bits and other mathematical data) as well as an agreement on methods for IKE phase II are exchanged between the peers.
- In terms of performance, the generation of the Diffie-Hellman Key is slow and heavy. The outcome of this phase is the IKE SA, an agreement on keys and methods for IKE phase II. Figure below illustrates the process that takes place during IKE phase I.



IKE Phase II (Quick mode or IPSec Phase)

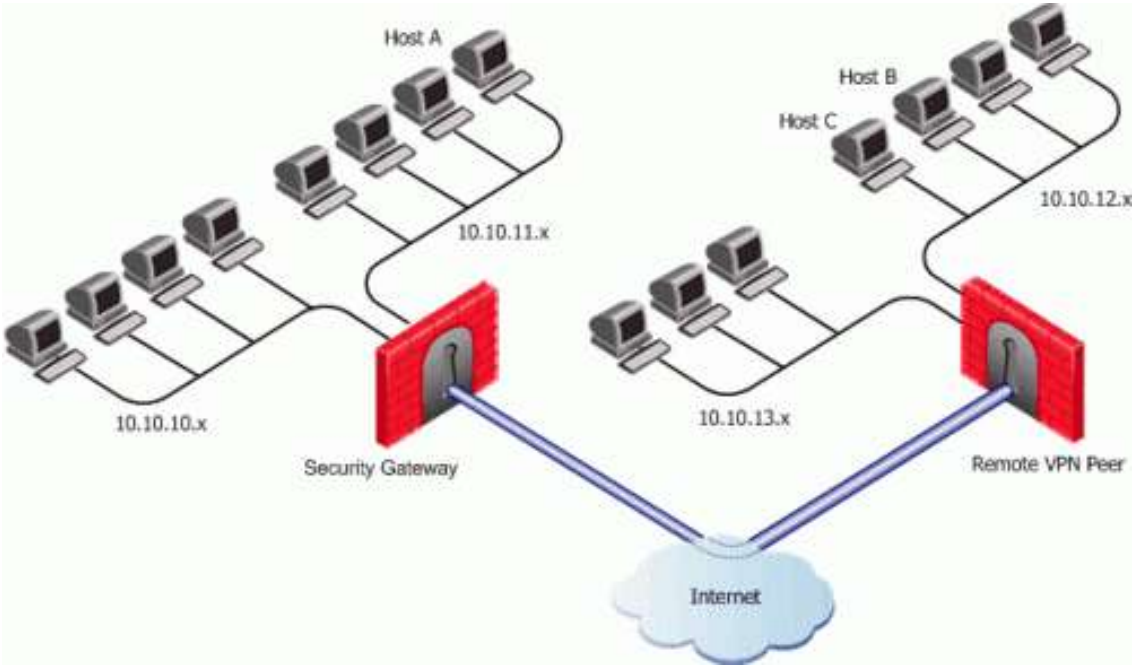
- IKE phase II is encrypted according to the keys and methods agreed upon in IKE phase I. The key material exchanged during IKE phase II is used for building the IPsec keys.
- The outcome of phase II is the IPsec Security Association. The IPsec SA is an agreement on keys and methods for IPsec, thus IPsec takes place according to the keys and methods agreed upon in IKE phase II.
- After the IPsec keys are created, bulk data transfer takes place:





Subnets and Security Associations

- By default, a VPN tunnel is created for the complete subnets that host computers reside on, and not just for the host computers involved in the communication.
- A Security Gateway protects a network consisting of two subnets (10.10.10.x, and 10.10.11.x, with netmask 255.255.255.0 for both). A second Security Gateway, the remote peer, protects subnets 10.10.12.x and 10.10.13.x, with netmask 255.255.255.0.
- Because a VPN tunnel is created by default for complete subnets, four SA's exist between the Security Gateway and the peer Security Gateway. When Host A communicates with Host B, an SA is created between Host A's subnet and Host B's subnet.

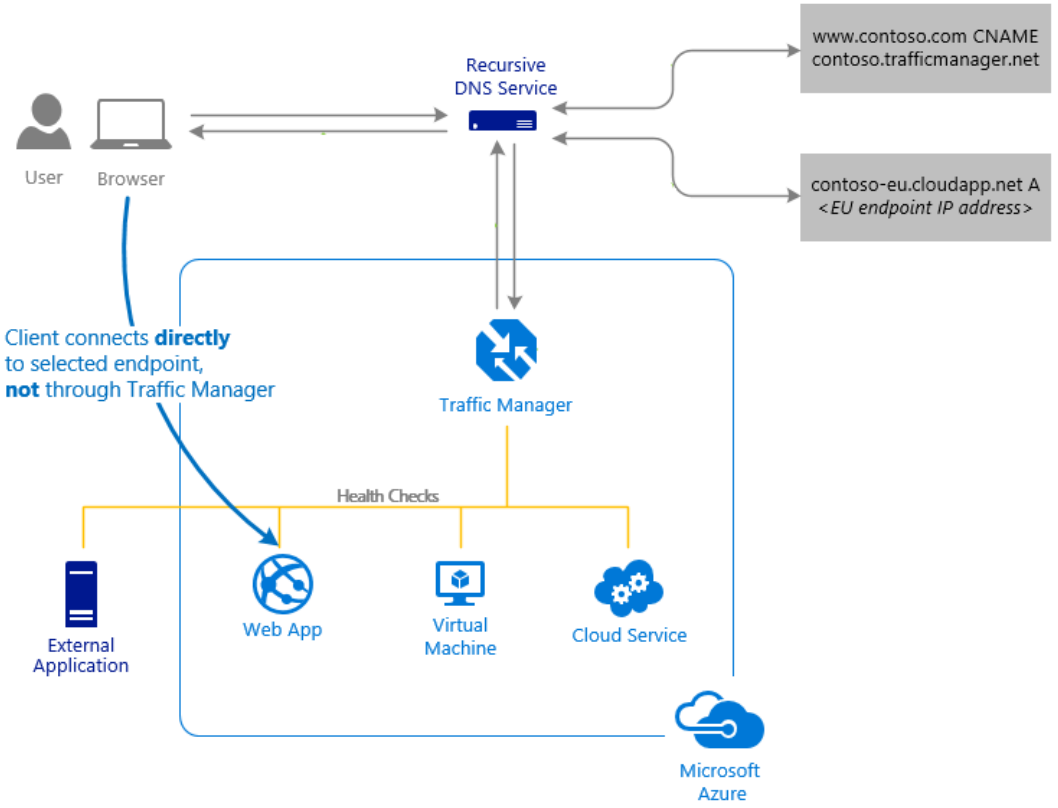


DNS Load Balancing & Azure Traffic Manager

- The Azure Traffic Manager service provides global DNS load balancing. Traffic Manager responds to clients with the IP address of a healthy endpoint, based on one of the following routing methods:
- **Geographic:** Clients are directed to specific endpoints (Azure, external or nested) based on which geographic location their DNS query originates from. This method enables complying with data sovereignty mandates, localization of content & user experience, and measuring traffic from different regions.
  - **Performance:** The IP address returned to the client is the "closest" to the client. Traffic Manager maintains an Internet latency table to track the round-trip time between IP address ranges and each Azure datacenter.
  - **Priority:** Traffic is directed to the primary (highest-priority) endpoint. If the primary endpoint is not available, Traffic Manager routes the traffic to the second endpoint, and so on. Availability of the endpoint is based on the configured status (enabled or disabled) and the ongoing endpoint monitoring.
  - **Weighted round-robin:** For each request, Traffic Manager randomly chooses an available endpoint. The probability of choosing an endpoint is based on the weights assigned to all available endpoints. Using higher or lower weights on specific endpoints causes those endpoints to be returned more or less frequently in the DNS responses.

Traffic Manager

- The picture shows a request for a web application directed to a Web App endpoint. Endpoints can also be other Azure services such as VMs and Cloud Services.
- The client connects directly to that endpoint.
- Azure Traffic Manager detects when an endpoint is unhealthy and then redirects clients to a different, healthy endpoint.

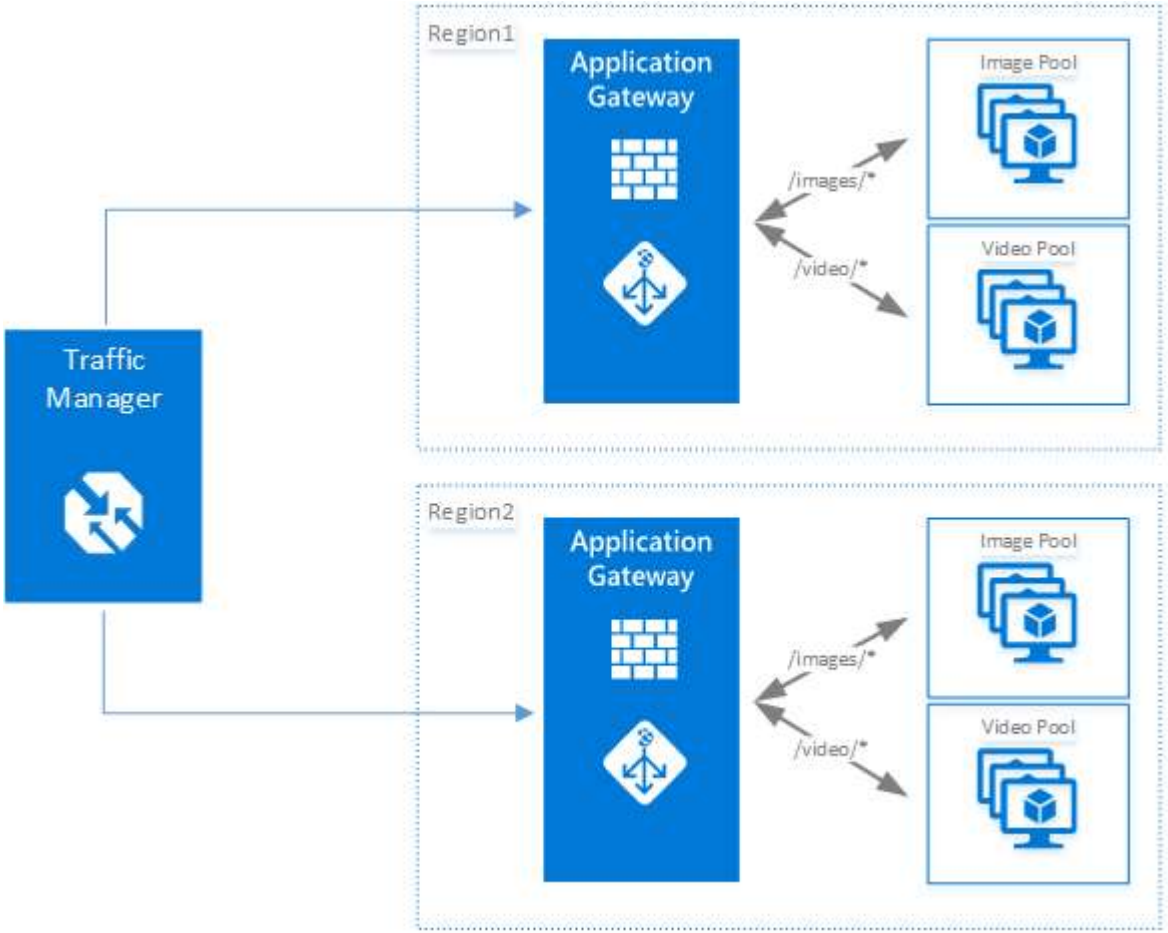


Application Load Balancing

- The Azure Application Gateway service provides application delivery controller (ADC) as a service.
- Application Gateway offers various Layer 7 (HTTP/HTTPS) load-balancing capabilities for your applications, including a web application firewall to protect your web applications from vulnerabilities and exploits.
- Application Gateway also allows you to optimize web farm productivity by offloading CPU-intensive SSL termination to the application gateway.
- Other Layer 7 routing capabilities include round-robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single application gateway.
- Application Gateway can be configured as an Internet-facing gateway, an internal-only gateway, or a combination of both.
- Application Gateway is fully Azure managed, scalable, and highly available. It provides a rich set of diagnostics and logging capabilities for better manageability.

URL path-based Routing

- The picture illustrates URL path-based routing with Application Gateway:

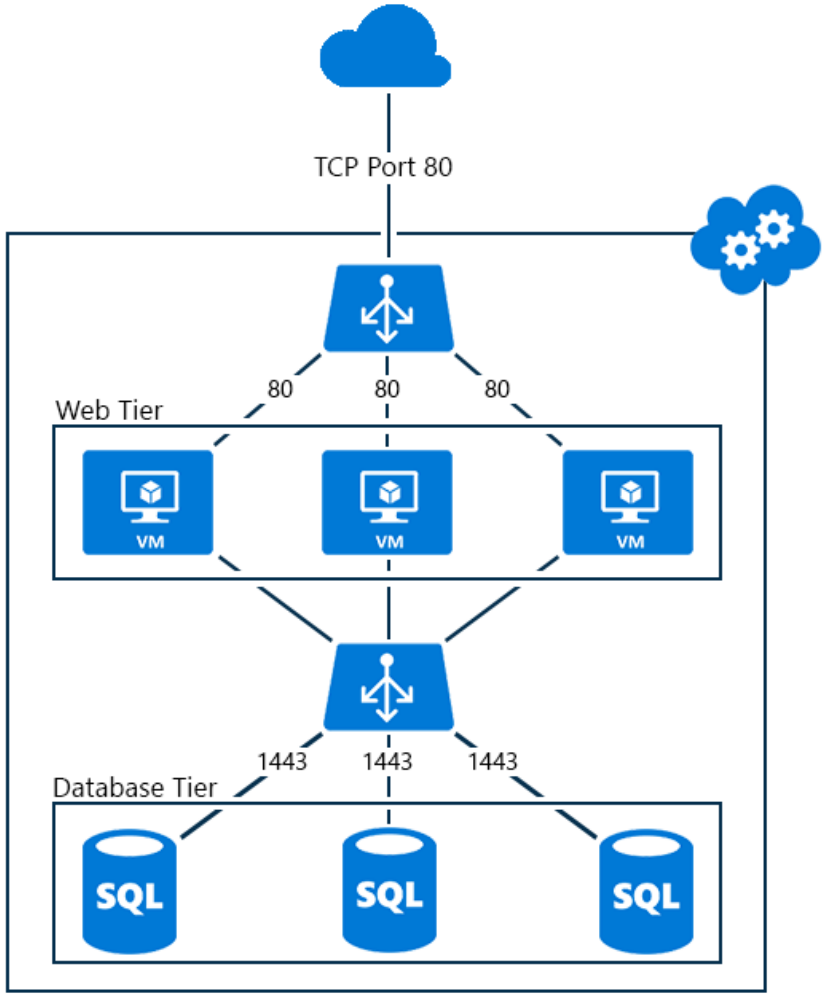


Layer 4 and Layer 7 Load Balancing

- **Layer 4 load balancing** operates at the intermediate *transport* layer, which deals with delivery of messages with no regard to the content of the messages. Transmission Control Protocol (TCP) is the Layer 4 protocol for Hypertext Transfer Protocol (HTTP) traffic on the Internet.
- Layer 4 load balancers simply forward network packets to and from the upstream server without inspecting the content of the packets. Layer 4 load balancers can make limited routing decisions by inspecting the first few packets in the TCP stream.
- **Layer 7 load balancing** operates at the high-level *application* layer, which deals with the actual content of each message. HTTP is the predominant Layer 7 protocol for website traffic on the Internet.
- Layer 7 load balancers route network traffic in a much more sophisticated way than Layer 4 load balancers, particularly applicable to TCP-based traffic such as HTTP. A Layer 7 load balancer terminates the network traffic and reads the message within. It can make a load-balancing decision based on the content of the message (the URL or cookie, for example). It then makes a new TCP connection to the selected upstream server (or reuses an existing one, by means of HTTP keepalives) and writes the request to the server.
- Layer 7 load balancing is more CPU-intensive than packet-based Layer 4 load balancing, but rarely causes degraded performance on a modern server.
- Layer 7 load balancing enables the load balancer to make smarter load-balancing decisions, and to apply optimizations and changes to the content (such as compression and encryption). It uses buffering to offload slow connections from the upstream servers, which improves performance.

Network Load Balancer

- The Azure Load Balancer provides high-performance, low-latency **Layer 4** load-balancing for all UDP and TCP protocols. It manages inbound and outbound connections. You can configure public and internal load-balanced endpoints. You can define rules to map inbound connections to back-end pool destinations by using TCP and HTTP health-probing options to manage service availability.
- The picture shows an Internet-facing multi-tier application that utilizes both external and internal load balancers:



Network Security

You can filter traffic to and from Azure resources using the following options:

- **Network:** You can implement Azure network security groups (NSGs) to filter inbound and outbound traffic to Azure resources.
- Each NSG contains one or more inbound and outbound rules. Each rule specifies the source IP addresses, destination IP addresses, port, and protocol that traffic is filtered with.
- NSGs can be applied to individual subnets and individual VMs.
- **Application:** By using an Application Gateway with web application firewall you can protect your web applications from vulnerabilities and exploits.
- Common examples are SQL injection attacks, cross site scripting, and malformed headers.
- Application gateway filters out this traffic and stops it from reaching your web servers. You are able to configure what rules you want enabled. The ability to configure SSL negotiation policies is provided to allow certain policies to be disabled.

Routing

Azure creates default route tables that enable resources connected to any subnet in any VNet to communicate with each other.

We can implement the following types of routes to override the default routes Azure creates:

- **User-defined:** We can create custom route tables with routes that control where traffic is routed to for each subnet.
- **Border gateway protocol (BGP):** If we connect your VNet to our on-premises network using an Azure VPN Gateway or ExpressRoute connection, we can propagate BGP routes to your VNets.
- BGP is the standard routing protocol commonly used in the Internet to exchange routing and reachability information between two or more networks. When used in the context of Azure Virtual Networks, BGP enables the Azure VPN Gateways and our on-premises VPN devices, called BGP peers or neighbors, to exchange "routes" that inform both gateways on the availability and reachability for those prefixes to go through the gateways or routers involved.
- BGP can also enable transit routing among multiple networks by propagating routes a BGP gateway learns from one BGP peer to all other BGP peers.

Network Monitoring and Managing

Azure provides many tools to monitor and manage networking:

- **Activity logs:** All Azure resources have activity logs which provide information about operations taken place, status of operations and who initiated the operation.
- **Diagnostic logs:** Periodic and spontaneous events are created by network resources and logged in Azure storage accounts, sent to an Azure Event Hub, or sent to Azure Log Analytics. Diagnostic logs provide insight to the health of a resource. Diagnostic logs are provided for Load Balancer (Internet-facing), Network Security Groups, routes, and Application Gateway.
- **Metrics:** Metrics are performance measurements and counters collected over a period of time on resources. Metrics can be used to trigger alerts based on thresholds. Currently metrics are available on Application Gateway.
- **Troubleshooting:** Troubleshooting information is accessible directly in the Azure portal. The information helps diagnose common problems with ExpressRoute, VPN Gateway, Application Gateway, Network Security Logs, Routes, DNS, Load Balancer, and Traffic Manager.
- **Role-based access control (RBAC):** Control who can create and manage networking resources with role-based access control (RBAC).
- **Packet capture:** The Azure Network Watcher service provides the ability to run a packet capture on a VM through an extension within the VM. This capability is available for Linux and Windows VMs.
- **Verify IP flows:** Network Watcher allows you to verify IP flows between an Azure VM and a remote resource to determine whether packets are allowed or denied. This capability provides administrators the ability to quickly diagnose connectivity issues.
- **Troubleshoot VPN connectivity:** The VPN troubleshooter capability of Network Watcher provides the ability to query a connection or gateway and verify the health of the resources.
- **View network topology:** View a graphical representation of the network resources in a VNet with Network Watcher.

Deployment & Configuration

- You can deploy and configure Azure networking resources with any of the following tools:
- **Azure portal: Azure PowerShell:**
- **Azure command-line interface (CLI):**
- **Azure Resource Manager templates:**
  - Templates are files (in JSON format) that define the infrastructure and configuration of an Azure solution. By using a template, we can repeatedly deploy our solution throughout its lifecycle and have confidence our resources are deployed in a consistent state.

- Some network resources are free, others are not

VPN Gateways Pricing

VPN GATEWAY TYPE	PRICE	BANDWIDTH	S2S TUNNELS	P2S TUNNELS
Basic	$0.04/hour	100 Mbps	Max 10 1-10: Included	Max 128 1-128: Included
VpnGw1	$0.19/hour	650 Mbps	Max 30 1-10: Included 11-30: $0.015/hour per tunnel	Max 128 1-128: Included
VpnGw2	$0.49/hour	1 Gbps	Max 30 1-10: Included 11-30: $0.015/hour per tunnel	Max 128 1-128: Included
VpnGw3	$1.25/hour	1.25 Gbps	Max 30 1-10: Included 11-30: $0.015/hour per tunnel	Max 128 1-128: Included

Application Gateway Pricing

APPLICATION GATEWAY TYPE	BASIC APPLICATION GATEWAY	WEB APPLICATION FIREWALL APPLICATION GATEWAY
Small	**$0.025** per gateway-hour (~$18.30/month)	Not available
Medium	**$0.07** per gateway-hour (~$51.24/month)	**$0.126** per gateway-hour (~$92.24/month)
Large	**$0.32** per gateway-hour (~$234.24/month)	**$0.448** per gateway-hour (~$327.94/month)

- Data processing charge is based on the amount of data processed by the application gateways

DATA PROCESSING	SMALL	MEDIUM	LARGE
First 10 TB/month	$0.008 per GB	Free	Free
Next 30 TB (10–40 TB)/month	$0.008 per GB	$0.007 per GB	Free
Over 40 TB/month	$0.008 per GB	$0.007 per GB	$0.0035 per GB

- Inbound data transfers (i.e. data going into Azure data centers)—**Free**
- Outbound data transfers (i.e., data going out of Azure data centers from application gateways will be charged at standard data transfer rates.)

DNS

- Azure DNS billing is based on the number of DNS zones hosted in Azure and the number of DNS queries received.

DNS	PUBLIC ZONES	PRIVATE ZONES (MANAGED PREVIEW)
First 25 hosted DNS zones	$0.50 per zone per month[1]	No Charge
Additional hosted DNS zones (over 25)	$0.10 per zone per month[1]	No Charge
First billion DNS queries/month	$0.40 per million[2]	No Charge
Additional DNS queries (over 1 billion)/month	$0.20 per million[2]	No Charge

Express Route

PORT SPEED	PRICE PER MONTH	PRICE PER MONTH WITH PREMIUM ADD-ON	INBOUND DATA TRANSFER INCLUDED	OUTBOUND DATA TRANSFER INCLUDED
50 Mbps	$55	$130	Unlimited	None
100 Mbps	$110	$200	Unlimited	None
200 Mbps	$145	$295	Unlimited	None
500 Mbps	$290	$690	Unlimited	None
1 Gbps	$436	$1,186	Unlimited	None
2 Gbps	$872	$2,372	Unlimited	None
5 Gbps	$2,180	$5,180	Unlimited	None
10 Gbps	$5,000	$8,000	Unlimited	None

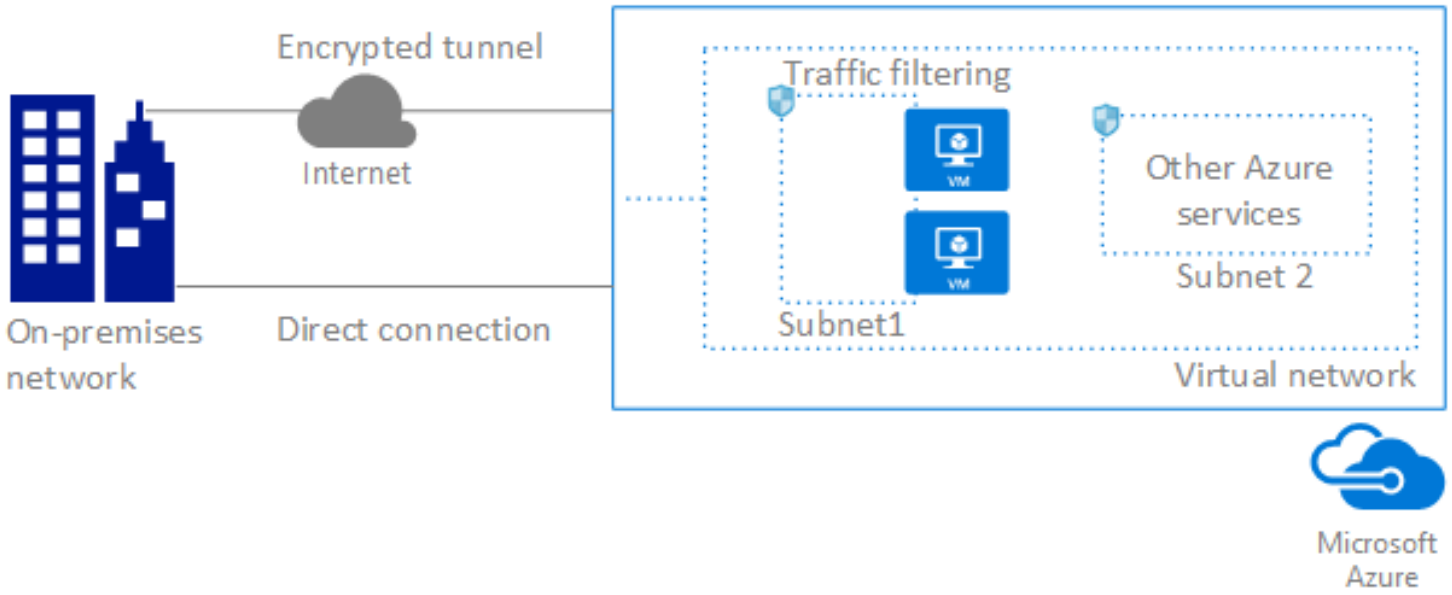
Traffic Manager

- Traffic Manager billing is based on the number of DNS queries received, with a discount for services receiving more than 1 billion monthly queries.
- Azure charges for each monitored endpoint (internal and external points differ).

	PRICE
First 1 billion DNS queries / month	$0.54 per million queries
Over 1 billion DNS queries / month	$0.375 per million queries
Basic Health Checks	
Basic Health Checks (Azure)	$0.36 per Azure endpoint / month
Fast Interval Health Checks Add-on (Azure)[1]	$1 per Azure endpoint / month
Basic Health Checks (External)	$0.54 per External endpoint / month
Fast Interval Health Checks Add-on (External)[1]	$2 per External endpoint / month

Azure Virtual Network

- Azure Virtual Networks enable Azure resources to securely communicate with your own resources in a virtual network.
- A virtual network could be an extension of your own network into the cloud.
- A virtual network is a logical isolation of the Azure cloud dedicated to your subscription.
- You can connect virtual networks to other virtual networks, or to your on-premises network.



Key Capabilities of Azure VNets

- **Isolation:** Virtual networks are isolated from one another. You can create separate virtual networks for development, testing, and production that use the same CIDR (10.0.0.0/0, for example) address blocks.
- Conversely, you can create multiple virtual networks that use different CIDR address blocks and connect the networks together. You can segment a virtual network into multiple subnets. Azure provides internal name resolution for resources deployed in a virtual network. If necessary, you can configure a virtual network to use your own DNS servers.
- **Internet communication:** Resources, such as virtual machines deployed in a virtual network, have access to the Internet. You can also enable inbound access to specific resources, as needed.
- **Azure resource communication:** Azure resources deployed in a virtual network can communicate with each other using private IP addresses, even if the resources are deployed in different subnets. Azure provides default routing between subnets, connected virtual networks. You can customize Azure's routing.
- **Virtual network connectivity:** Virtual networks can be connected to each other, enabling resources in any virtual network to communicate with resources in any other virtual network.
- **On-premises connectivity:** A virtual network can be connected to an on-premises network, enabling resources to communicate between each other.
- **Traffic filtering:** You can filter network traffic to and from resources in a virtual network by source IP address and port, destination IP address and port, and protocol.
- **Routing:** You can optionally override Azure's default routing by configuring your own routes, or by propagating BGP routes through a network gateway.

Network isolation, segmentation, Internet access

- You can implement multiple virtual networks within each Azure subscription and Azure region.
- Each virtual network is isolated from other virtual networks.
- For each virtual network you can:
  - Specify a custom private IP address space using public and private addresses.
  - Azure assigns resources in a virtual network a private IP address from the address space you assign.
  - Segment the virtual network into one or more subnets and allocate a portion of the virtual network's address space to each subnet.
  - Use Azure-provided name resolution, or specify your own DNS server, for use by resources in a virtual network.

- All resources in a virtual network can communicate outbound to the Internet.
- By default, the private IP address of the resource is source network address translated (SNAT) to a public IP address selected by the Azure infrastructure.
- To communicate inbound to Azure resources from the Internet, or to communicate outbound to the Internet without SNAT, a resource must be assigned a public IP address.

Filtering of Traffic Between Subnets

You can filter network traffic between subnets using either or both of the following options:

- **Network security groups:** A network security group can contain multiple inbound and outbound security rules that enable you to filter traffic by source and destination IP address, port, and protocol.
- You can apply a network security group to each network interface in a virtual machine.
- You can also apply a network security group to the subnet a network interface, or other Azure resource, is in.
- **Network virtual appliances:** A network virtual appliance is a virtual machine running software that performs a network function, such as a firewall. There are many available network virtual appliances in the Azure Marketplace.
- Network virtual appliances are also available that provide WAN optimization and other network traffic functions.
- Network virtual appliances are typically used with user-defined or BGP routes. You can also use a network virtual appliance to filter traffic between virtual networks.

Routing

Azure creates route tables that enable resources connected to any subnet in any virtual network to communicate with each other, and the Internet, by default.

You can implement either or both of the following options to override the default routes Azure creates:

- **User-defined routes:** You can create custom route tables with routes that control where traffic is routed to for each subnet.
- **BGP routes:** If you connect your virtual network to your on-premises network using an Azure VPN Gateway or ExpressRoute connection, you can propagate BGP routes to your virtual networks.

Pricing

- There is no charge for virtual networks, subnets, route tables, or network security groups.
- Outbound Internet bandwidth usage, public IP addresses, virtual network peering, VPN Gateways, and ExpressRoute each have their own pricing structures.

Limits on Azure Networks

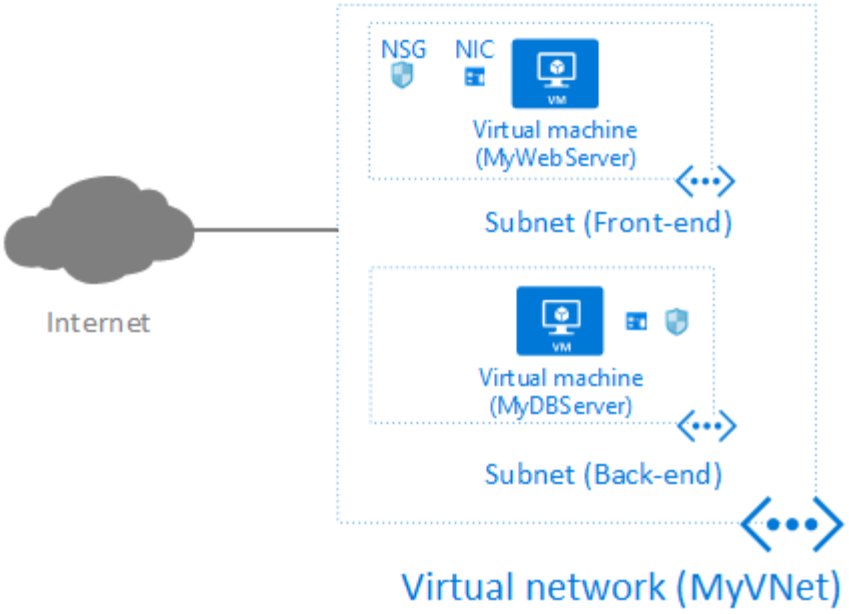
Resource Default	Resource Default limit	Maximum limit
Virtual networks per subscription	50	500
DNS servers per virtual network	9	25
Private IP addresses per virtual network	4,096	4,096
Concurrent TCP connections for a virtual machine or role instance	500k	500k
Network interfaces	300	10k
Network security groups	100	400
NSG rules per NSG	200	500
User defined route tables	100	400
User defined routes per route table	100	400
Public IP addresses (dynamic)	60	Limit can be increased by support
Public IP addresses (static)	20	Limit can be increased by support
Load balancers (internal and Internet facing)	100	Limit can be increased by support
Load balancer rules per load balancer	150	150
Public frontend IP per load balancer	5	Limit can be increased by support
Private frontend IP per load balancer	1	Limit can be increased by support
Application gateways	50	50

Virtual Networks

- Normally you're setting up a network like you do in your on-premises network.
- You create a network with an IP range such as 10.0.0.0/16 and split it up into different subnetworks. Every Azure VNet has at least a minimum of two subnets.
- The first subnet is the **gateway subnet**, which is basically a router network where every internal network router for the other Azure VNet subnetworks is in.
- The gateway subnet needs a minimum of /29 CIDR IPs. Some people recommend /24 CIDR.
- The second one is the network for your services or servers depending on your own design, normally it is /24 CIDR.

Network Diagrams

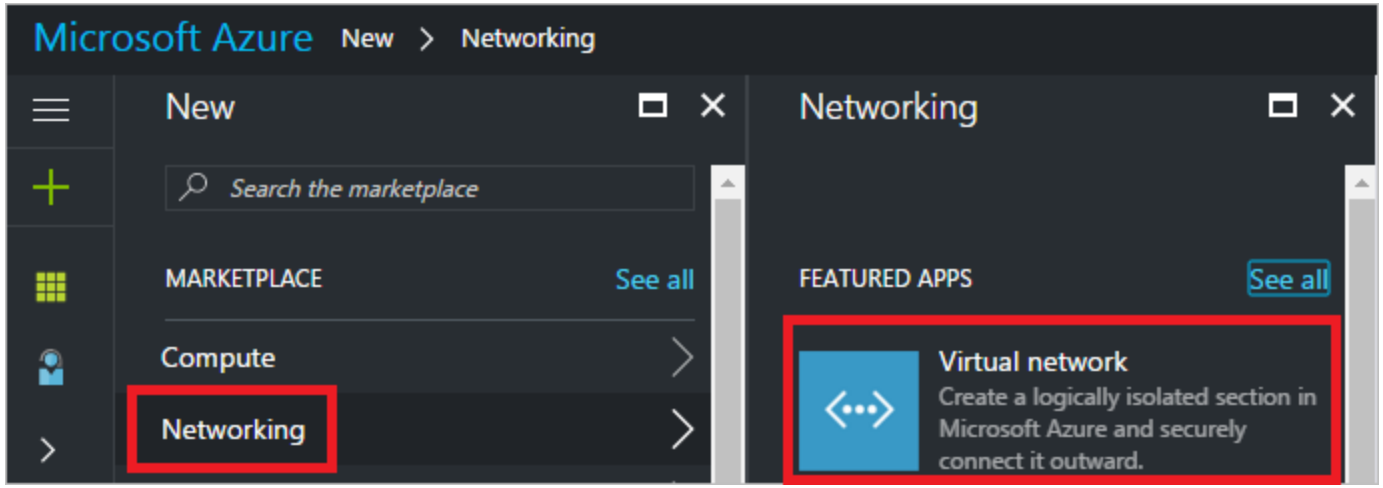
- We will create a virtual network (VNet) with two subnets, create two virtual machines (VM), and connect each VM to one of the subnets:



- An Azure virtual network (VNet) is a representation of your own network in the cloud. You can control your Azure network settings and define DHCP address blocks, DNS settings, security policies, and routing.
- Different subnets are typically used to control the flow of traffic between subnets.

Virtual Network with two Subnets

- To create a virtual network with two subnets do the following:
  1. Log in to the Azure portal.
  2. In the Favorites pane, of the portal, click New.
  3. In the New blade, click Networking. In the Networking blade, click Virtual network, as shown in the following picture:



  4. In the Virtual network blade, leave Resource Manager selected as the deployment model, and click Create

Create virtual network blade

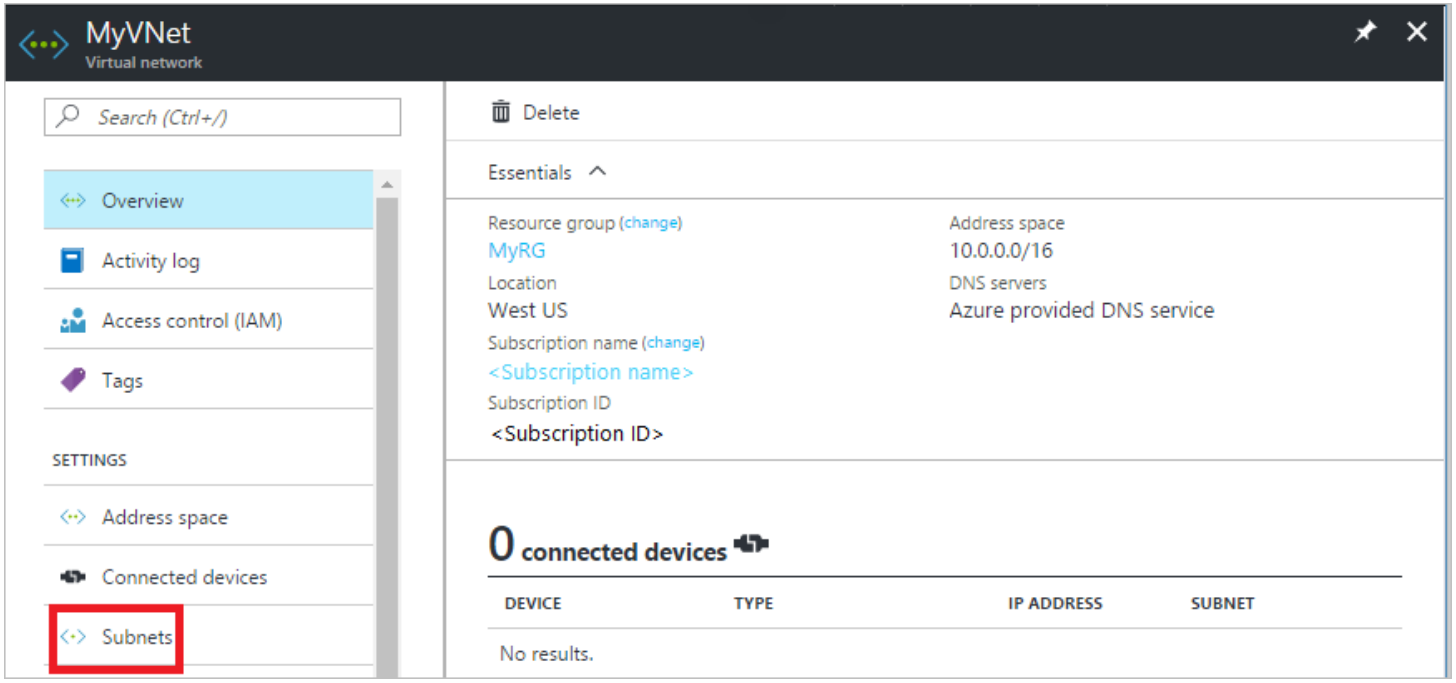
5. In the **Create virtual network blade** that appears, enter the following values, then click **Create**:

Setting	Value	Details
Name	*MyVNet*	The name must be unique within the resource group.
Address space	*10.0.0.0/16*	You can specify any address space you like in CIDR notation.
Subnet name	*Front-end*	The subnet name must be unique within the virtual network.
Subnet address range	*10.0.0.0/24*	The range you specify must exist within the address space you defined for the virtual network.
Subscription	*[Your subscription]*	Select a subscription to create the VNet in. A VNet exists within a single subscription.
Resource group	**Create new:***MyRG*	Create a resource group. The resource group name must be unique within the subscription you selected. To learn more about resource groups, read the Resource Manager overview article.
Location	*West US*	Typically the location that is closest to your physical location is selected.

The VNet takes a few seconds to create

All resources Blade

6. *With the virtual network created, in the Azure portal **Favorites** pane, click **All resources**. Click the **MyVNet** virtual network in the **All resources** blade. If the subscription you selected already has several resources in it, you can enter MyVNet in the **Filter by name…**box to easily access the VNet.*
7. The **MyVNet** blade opens and displays information about the Vnet:



8. Click **Subnets** to display a list of the subnets within the VNet. The only subnet that exists is **Front-end**, the subnet you created in step 5.

MyVNet Subnet blade

9. In the MyVNet - Subnets blade, click **+ Subnet** to create a subnet with the following information and click **OK** to create the subnet:

Setting	Value	Details
Name	*Back-end*	The name must be unique within the virtual network.
Address range	*10.0.1.0/24*	The range you specify must exist within the address space you defined for the virtual network.
Network security group and **Route table**	*None*(default)	Network security groups (NSG)s are covered later in this article

10. After the new subnet is added to the VNet, you can close the **MyVNet – Subnets** blade, then close the **All resources** blade.

Create Virtual Machine: Web Server VM

- With the VNet and subnets created, you can create the VMs. We will create both VMs with the Windows Server operating system.
- To create the web server VM, complete the following steps:

1. In the Azure portal favorites pane, click **New**, **Compute**, then **Windows Server 2016 Datacenter**.
2. In the **Windows Server 2016 Datacenter** blade, click **Create**.
3. In the **Basics** blade that appears, enter or select the following values and click **OK**:

Setting	Value	Details
Name	*MyWebServer*	This VM serves as a web server that Internet resources connect to.
VM disk type	*SSD*	
User name	*Your choice*	
Password	*Your choice*	
Subscription		The subscription must be the same subscription that you selected in step 5 of the Create a virtual network with two subnets step. The VNet you connect a VM to must exist in the same subscription as the VM.
Resource group	**Use existing:** Select *MyRG*	Though we're using the same resource group as we did for the VNet, the resources don't have to exist in the same resource group.
Location	*West US*	The location must be the same location that you specified in step 5 of the Create a virtual network with two subnets step

Choose a size blade

4. In the **Choose a size** blade, click *DS1_V2 Standard*, then click **Select**.
5. In the **Settings** blade, enter or select the following values and click **OK**:

Setting	Value	Details
Storage: Use managed disks	*Yes*	
Virtual network	Select *MyVNet*	You can select any VNet that exists in the same location as the VM you're creating.
Subnet	Select *Front-end*	You can select any subnet that exists within the VNet.
Public IP address	Accept the default	A public IP address enables you to connect to the VM from the Internet.
Network security group (firewall)	Accept the default	Click the (new) MyWebServer-nsg default NSG the portal created to view its settings. In the Create network security group blade that opens, notice that it has one inbound rule that allows TCP/3389 (RDP) traffic from any source IP address.
All other values	Accept the defaults	

6. In the **Summary** blade, review the settings and click **OK** to create the VM. A status tile is displayed on the portal dashboard as the VM creates. It may take a few minutes to create. You don't need to wait for it to complete. You can continue to the next step while the VM is created.

Create the database server VM

1. In the Favorites pane, click **New**, **Compute**, then **Windows Server 2016 Datacenter**.
2. In the **Windows Server 2016 Datacenter** blade, click **Create**.
3. In the **Basics blade**, enter or select the following values, then click **OK**:

Setting	Value	Details
Name	*MyDBServer*	This VM serves as a database server that the web server connects to, but that the Internet cannot connect to.
VM disk type	*SSD*	
User name	Your choice	
Password	Your choice	
Subscription		The subscription must be the same subscription that you selected in Step 5 of the Create a virtual network with two subnets step
Resource group	**Use existing:** Select *MyRG*	Though we're using the same resource group as we did for the VNet, the resources don't have to exist in the same resource group.
Location	*West US*	The location must be the same location that you specified in step 5 of the Create a virtual network with two subnets step

Choose a size blade

4. In the **Choose a size** blade, click *DS1_V2 Standard*, then click **Select**.
5. In the **Settings** blade, enter or select the following values and click **OK**:

Setting	Value	Details
Storage: Use managed disks	*Yes*	
Virtual network	Select *MyVNet*	You can select any VNet that exists in the same location as the VM you're creating.
Subnet	Select *Back-end*by clicking the **Subnet** box, then selecting **Back-end** from the **Choose a subnet** blade	You can select any subnet that exists within the VNet.
Public IP address	None – Click the default address, then click **None**from the **Choose public IP address** blade	Without a public IP address, you can only connect to the VM from another VM connected to the same VNet. You cannot connect to it directly from the Internet.
Network security group (firewall)	Accept the default	This NSG also has the same default inbound rule. You might add an additional inbound rule for TCP/1433 (MS SQL) for a database server. There is no rule for outbound traffic because by default, all outbound traffic is allowed.
All other values	Accept the defaults	

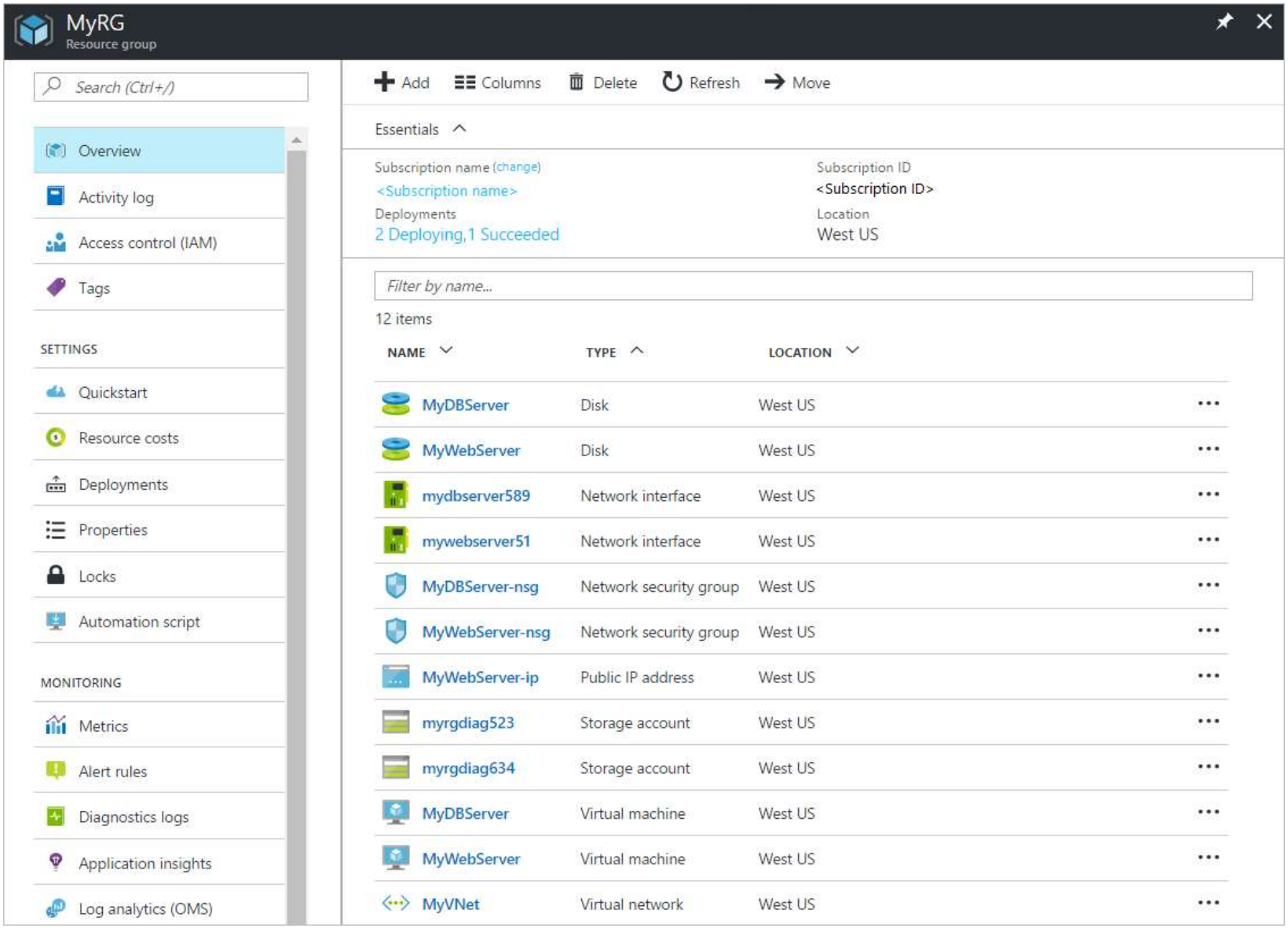
6. In the **Summary** blade, review the settings and click **OK** to create the VM.

Review Resources

- Though we created one VNet and two VMs, the Azure portal created several additional resources for you in the MyRG resource group.
- We can review the contents of the MyRG resource group by completing the following steps

1. In the **Favorites** pane, click **More services**.
2. In the **More services** pane, type *Resource groups* in the box that has the word *Filter* in it. Click **Resource groups** when you see it in the filtered list.
3. In the **Resource groups** pane, click the *MyRG* resource group. If you have many existing resource groups in your subscription, you can type *MyRG* in the box that contains the text *Filter by name…* to quickly access the MyRG resource group.
4. In the **MyRG** blade, you see that the resource group contains 12 resources, as shown in the picture on the following slide:

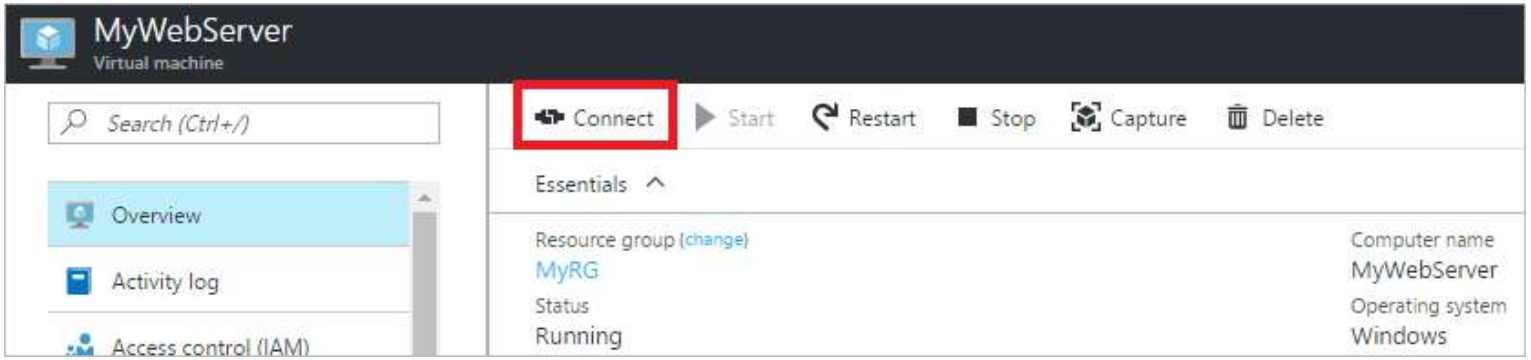
MyRG blade


MyRG
Resource group
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
SETTINGS
Quickstart
Resource costs
Deployments
Properties
Locks
Automation script
MONITORING
Metrics
Alert rules
Diagnostics logs
Application insights
Log analytics (OMS)
Add
Columns
Delete
Refresh
Move
Essentials
Subscription name (change)
<Subscription name>
Deployments
2 Deploying,1 Succeeded
Subscription ID
<Subscription ID>
Location
West US
Filter by name...
12 items
NAME
TYPE
LOCATION
MyDBServer Disk West US
MyWebServer Disk West US
mydbserver589 Network interface West US
mywebserver51 Network interface West US
MyDBServer-nsg Network security group West US
MyWebServer-nsg Network security group West US
MyWebServer-ip Public IP address West US
myrgdiag523 Storage account West US
myrgdiag634 Storage account West US
MyDBServer Virtual machine West US
MyWebServer Virtual machine West US
MyVNet Virtual network West US

Connect to the VMs

With your VNet and two VMs created, you can now connect to the VMs

1. In the portal, open the MyRG resource group by completing the steps in the Review resources step.
2. In the **MyRG** blade, click the **MyWebServer** VM.
3. In the **MyWebServer** blade, click **Connect**, as shown in the following picture:



4. Allow your browser to download the *MyWebServer.rdp* file, then open it.
5. If you receive a dialog box informing you that the publisher of the remote connection cannot be verified, click **Connect**.
6. When entering your credentials, ensure you login with the user name and password you specified in step 3 of the Create the web server VM section of this article. If the **Windows Security** box that appears doesn't list the correct credentials, you may need to click **More choices**, then **Use a different account**, so you can specify the correct user name and password). Click **OK** to connect to the VM.
7. If you receive a **Remote Desktop Connection** box informing you that the identity of the remote computer cannot be verified, click **Yes**.
8. You are now connected to the MyWebServer VM from the Internet. Leave the remote desktop connection open to complete the steps in the next section.

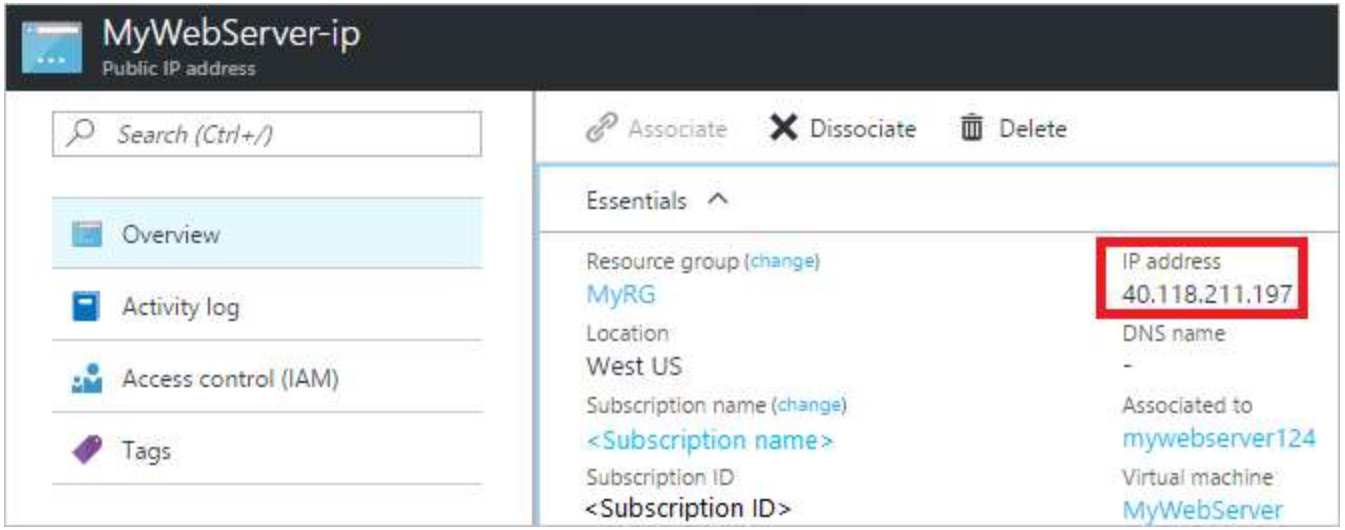
Connection to Web Server

- The remote connection is to the public IP address assigned to the public IP address resource the portal created in step 5 of the Create a virtual network with two subnets step.
- The connection is allowed because the default rule created in the **MyWebServer-nsg** NSG permitted TCP/3389 (RDP) inbound to the VM from any source IP address.
- If you try to connect to the VM over any other port, the connection fails, unless you add additional inbound rules to the NSG allowing the additional ports.
- If you add additional inbound rules to the NSG, ensure that the same ports are open on the Windows firewall, or the connection fails.

Connect Outbound to the Internet

1. If you don't already have a remote connection to the MyWebServerVM open, make a remote connection to the VM by completing the steps in the Connect to the web server VM from the Internet section of this article.
2. From the Windows desktop, open Internet Explorer. In the **Setup Internet Explorer 11** dialog box, click **Don't use recommended settings**, then click **OK**. It's recommended to accept the recommended settings for a production server.
3. In the Internet Explorer address bar, enter bing.com. If you receive an Internet Explorer dialog box, click **Add**, then **Add** in the **Trusted sites** dialog box and click **Close**. Repeat this process for any other Internet Explorer dialog boxes.
4. At the Bing search page, enter *whatsmyipaddress*, then click the magnifying glass button. Bing returns the public IP address assigned to the public IP address resource created by the portal when you created the VM. If you examine the settings for the **MyWebServer-ip**resource, you see the same IP address assigned to the public IP address resource, as shown in the picture on the following slide. The IP address assigned to your VM will be different however.
5. Leave the remote desktop connection.

Note: on reboot this IP address may change



Outbound Connections are open by Default

- You are able to connect to the Internet from the VM because all outbound connectivity from the VM is allowed by default. You can limit outbound connectivity by adding addition rules to the NSG applied to the NIC, to the subnet the NIC is connected to, or both.
- If the VM is put in the stopped (deallocated) state using the portal, the public IP address can change. If you require that the public IP address never change, you can use the static allocation method for the IP address, rather than the dynamic allocation method (which is the default).

Connect to the database server VM from the web server VM

To connect to the database server VM from the web server VM, complete the following steps:

1. If you don't already have a remote connection to the MyWebServer VM open, make a remote connection to the
2. Click the Start button in the lower-left corner of the Windows desktop, then start typing *remote desktop*. When the Start menu list displays **Remote Desktop Connection**, click it.
3. In the **Remote Desktop Connection** dialog box, enter *MyDBServer* for the computer name and click **Connect**.
4. Enter the user name and passwords you entered in step 3 of the Create the database server VM step, then click **OK**.
5. If you receive a dialog box informing you that the identity of the remote computer cannot be verified, click **Yes**.
6. Leave the remote desktop connection to both servers open.

Connection between servers

You are able to make the connection to the database server VM from the web server VM for the following reasons:

- TCP/3389 inbound connections are enabled for any source IP in the default NSG created in step 5 of the Create the database server VM step
- You initiated the connection from the web server VM, which is connected to the same VNet as the database server VM. To connect to a VM that doesn't have a public IP address assigned to it, you must connect from another VM connected to the same VNet, even if the VM is connected to a different subnet.
- Even though the VMs are connected to different subnets, Azure creates default routes that enable connectivity between subnets. You can override the default routes by creating your own however.
- If you try to initiate a remote connection to the database server VM from the Internet, you will see that the **Connect** option is grayed out.
- Connect is grayed out because there is no public IP address assigned to the VM, so inbound connections to it from the Internet are not possible.

Public IP Addresses

- Public IP addresses allow Internet resources to communicate inbound to Azure resources.
- Public IP addresses also enable Azure resources to communicate outbound to Internet and public-facing Azure services with an IP address assigned to the resource.
- The address is dedicated to the resource, until it is unassigned by you. If a public IP address is not assigned to a resource, the resource can still communicate outbound to the Internet, but Azure dynamically assigns an available IP address that is not dedicated to the resource.
- In Azure Resource Manager, a public IP address is a resource that has its own properties. Some of the resources you can associate a public IP address resource with are:
  - Virtual machine network interfaces
  - Internet-facing load balancers
  - VPN gateways
  - Application gateways

Features of IP Addresses

IP address version

- Public IP addresses are created with an IPv4 or IPv6 address. Public IPv6 addresses can only be assigned to Internet-facing load balancers.

SKU

- Public IP addresses are created with one of the following SKUs:
- Basic

All public IP addresses created before the introduction of SKUs are Basic SKU public IP addresses. With the introduction of SKUs, you have the option to specify which SKU you would like the public IP address to be. Basic SKU addresses are:

  - Assigned with the static or dynamic allocation method.
  - Assigned to any Azure resource that can be assigned a public IP address, such as network interfaces, VPN Gateways, Application Gateways, and Internet-facing load balancers.
  - Can be assigned to a specific zone.
  - Not zone redundant.

- Standard

Standard SKU public IP addresses are:

  - Assigned with the static allocation method only.
  - Assigned to network interfaces or Standard Internet-facing load balancers.
  - Zone redundant by default. Can be created zonal and guaranteed in a specific availability zone.

Allocation Method, dynamic vs. static

There are two methods in which an IP address is allocated to a public IP address resource - *dynamic* or *static*.

- The default allocation method is *dynamic*, where an IP address is **not** allocated at the time of its creation. Instead, the public IP address is allocated when you start (or create) the associated resource (like a VM or load balancer).
- The IP address is released when you stop (or delete) the resource. After being released from resource A, for example, the IP address can be assigned to a different resource. If the IP address is assigned to a different resource while resource A is stopped, when you restart resource A, a different IP address is assigned.
- To ensure the IP address for the associated resource remains the same, you can set the allocation method explicitly to *static*.
- A static IP address is assigned immediately. The address is released only when you delete the resource or change its allocation method to *dynamic*. Azure assigns the IP address from a pool of available IP addresses in the Azure location the resource is created in.
- Static public IP addresses are commonly used in the following scenarios:
- When you must update firewall rules to communicate with your Azure resources.
- DNS name resolution, where a change in IP address would require updating A records.
- Your Azure resources communicate with other apps or services that use an IP address-based security model.
- You use SSL certificates linked to an IP address.

DNS hostname resolution

- You can specify a DNS domain name label for a public IP resource, which creates a mapping for *domainnamelabel*.*location*.cloudapp.azure.com to the public IP address in the Azure-managed DNS servers.
- For instance, if you create a public IP resource with **catalina** as a *domainnamelabel* in the **West US** Azure *location*, the fully qualified domain name (FQDN) **catalina.westus.cloudapp.azure.com** resolves to the public IP address of the resource.
- You can use the FQDN to create a custom domain CNAME record pointing to the public IP address in Azure.
- Instead of, or in addition to, using the DNS name label with the default suffix, you can use the Azure DNS service to configure a DNS name with a custom suffix that resolves to the public IP address.

IP Address of Load Balancers, VPN Gateways

- You can associate a public IP address created with either SKU with an Azure Load Balancer, by assigning it to the load balancer **frontend** configuration.
- The public IP address serves as a load-balanced virtual IP address (VIP).
- You can assign either a dynamic or a static public IP address to a load balancer front-end.
- You can also assign multiple public IP addresses to a load balancer front-end, which enables multi-VIP scenarios like a multi-tenant environment with SSL-based websites.

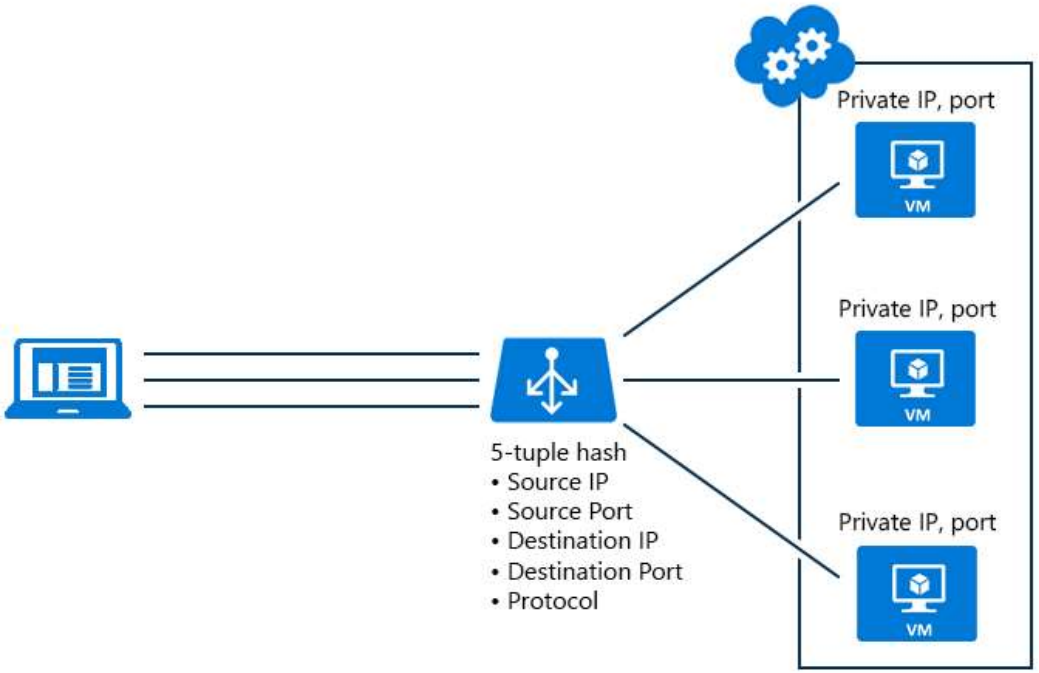
- An Azure VPN Gateway connects an Azure virtual network to other Azure virtual networks, or to an on-premises network. A public IP address is assigned to the VPN Gateway to enable it to communicate with the remote network. You can only assign a *dynamic* public IP address to a VPN gateway.
- You can associate a public IP address with an Azure Application Gateway by assigning it to the gateway's **frontend** configuration. This public IP address serves as a load-balanced VIP. You can only assign a *dynamic* public IP address to an application gateway frontend configuration.

Load Balancers

- Azure Load Balancer delivers high availability and network performance to your applications. It is a Layer 4 (TCP, UDP) load balancer that distributes incoming traffic among healthy instances of services defined in a load-balanced set.
- Azure Load Balancer can be configured to:
  - Load balance incoming Internet traffic to virtual machines. This configuration is known as Internet-facing load balancing.
  - Load balance traffic between virtual machines in a virtual network, between virtual machines in cloud services, or between on-premises computers and virtual machines in a cross-premises virtual network. This configuration is known as internal load balancing.
  - Forward external traffic to a specific virtual machine.

Load Balancers Features

- Azure Load Balancer uses a hash-based distribution algorithm. By default, it uses a 5-tuple hash composed of source IP, source port, destination IP, destination port, and protocol type to map traffic to available servers.
- Load Balancer provides stickiness only *within* a transport session. Packets in the same TCP or UDP session will be directed to the same instance behind the load-balanced endpoint.
- When the client closes and reopens the connection or starts a new session from the same source IP, the source port changes. This may cause the traffic to go to a different endpoint in a different datacenter.



Service Monitoring

Azure Load Balancer can probe the health of the various server instances. When a probe fails to respond, the load balancer stops sending new connections to the unhealthy instances. Existing connections are not impacted.

Three types of probes are supported:

- **Guest agent probe (on Platform as a Service Virtual Machines only):** The load balancer utilizes the guest agent inside the virtual machine. The guest agent listens and responds with an HTTP 200 OK response only when the instance is in the ready state (i.e. the instance is not in a state like busy, recycling, or stopping). If the agent fails to respond with an HTTP 200 OK, the load balancer marks the instance as unresponsive and stops sending traffic to that instance. The load balancer continues to ping the instance. If the guest agent responds with an HTTP 200, the load balancer will send traffic to that instance again.
- **HTTP custom probe:** This probe overrides the default (guest agent) probe. You can use it to create your own custom logic to determine the health of the role instance. The load balancer will regularly probe your endpoint (every 15 seconds, by default). The instance is considered to be in rotation if it responds with a TCP ACK or HTTP 200 within the timeout period (default of 31 seconds). This is useful for implementing your own logic to remove instances from the load balancer's rotation. For example, you can configure the instance to return a non-200 status if the instance is above 90% CPU.
- **TCP custom probe:** This probe relies on successful TCP session establishment to a defined probe port.

Options for Traffic Distribution

There are different options to distribute network traffic using Microsoft Azure. These options work differently from each other, having a different feature set and support different scenarios. They can each be used in isolation, or combining them.

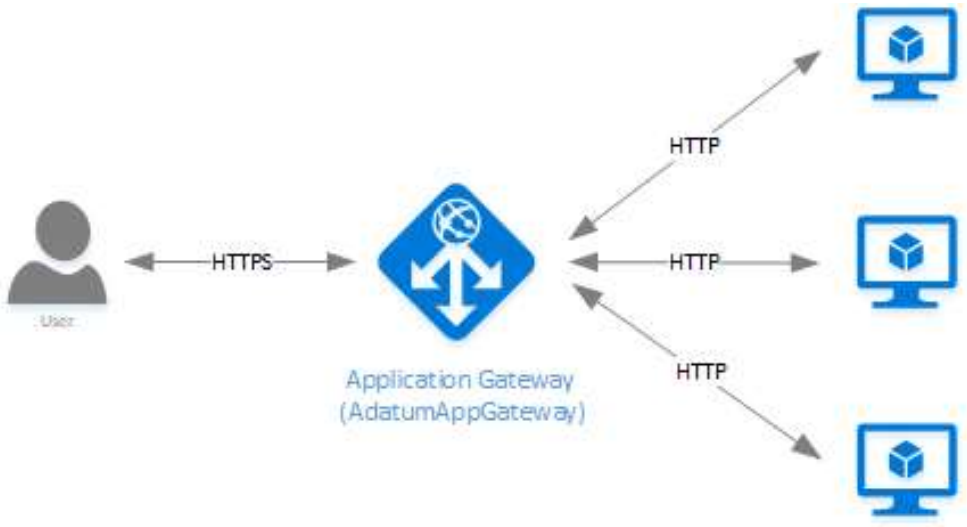
- **Azure Load Balancer** works at the transport layer (Layer 4 in the OSI network reference stack). It provides network-level distribution of traffic across instances of an application running in the same Azure data center.
- **Application Gateway** works at the application layer (Layer 7 in the OSI network reference stack). It acts as a reverse-proxy service, terminating the client connection and forwarding requests to back-end endpoints.
- **Traffic Manager** works at the DNS level. It uses DNS responses to direct end-user traffic to globally distributed endpoints. Clients then connect to those endpoints directly.
- Azure Load Balancer and Application Gateway route network traffic to endpoints but they have different usage scenarios to which traffic to handle. Table on the next slide presents the difference between the two load balancers.

Azure Load Balancer and Application Gateway

Type	Azure Load Balancer	Application Gateway
Protocols	UDP/TCP	HTTP, HTTPS, and WebSockets
IP reservation	Supported	Not supported
Load balancing mode	5-tuple(source IP, source port, destination IP, destination port, protocol type)	Round Robin Routing based on URL
Load balancing mode (source IP /sticky sessions)	2-tuple (source IP and destination IP), 3-tuple (source IP, destination IP, and port). Can scale up or down based on the number of virtual machines	Cookie-based affinity Routing based on URL
Health probes	Default: probe interval - 15 secs. Taken out of rotation: 2 Continuous failures. Supports user-defined probes	Idle probe interval 30 secs. Taken out after 5 consecutive live traffic failures or a single probe failure in idle mode. Supports user-defined probes
SSL offloading	Not supported	Supported
Url-based routing	Not supported	Supported
SSL Policy	Not supported	Supported

Application Gateways

- Microsoft Azure Application Gateway is a dedicated virtual appliance providing application delivery controller (ADC) as a service.
- Application Gateway offers various layer 7 load balancing capabilities for your application. It allows customers to optimize web farm productivity by offloading CPU intensive SSL termination to the application gateway. It also provides other layer 7 routing capabilities including round robin distribution of incoming traffic, cookie-based session affinity, URL path-based routing, and the ability to host multiple websites behind a single Application Gateway.
- A web application firewall (WAF) is also provided as part of the application gateway WAF SKU. It provides protection to web applications from common web vulnerabilities and exploits. Application Gateway can be configured as internet facing gateway, internal only gateway, or a combination of both.



Features of Application Gateways

Application Gateway currently provides the following capabilities:

- **Web application firewall** - The web application firewall (WAF) in Azure Application Gateway protects web applications from common web-based attacks like SQL injection, cross-site scripting attacks, and session hijacks.
- **HTTP load balancing** - Application Gateway provides round robin load balancing. Load balancing is done at Layer 7 and is used for HTTP(S) traffic only.
- **Cookie-based session affinity** - The cookie-based session affinity feature is useful when you want to keep a user session on the same back-end. By using gateway-managed cookies, the Application Gateway is able to direct subsequent traffic from a user session to the same back-end for processing. This feature is important in cases where session state is saved locally on the back-end server for a user session.
- **Secure Sockets Layer (SSL) offload** - This feature takes the costly task of decrypting HTTPS traffic off your web servers. By terminating the SSL connection at the Application Gateway and forwarding the request to the server unencrypted, the web server is unburdened by decryption. Application Gateway re-encrypts the response before sending it back to the client. This feature is useful in scenarios where the back-end is located in the same secured virtual network as the Application Gateway in Azure.
- **End to End SSL** - Application Gateway supports end to end encryption of traffic. Application Gateway does this by terminating the SSL connection at the application gateway. The gateway then applies the routing rules to the traffic, re-encrypts the packet, and forwards the packet to the appropriate backend based on the routing rules defined. Any response from the web server goes through the same process back to the end user.
- **URL-based content routing** - This feature provides the capability to use different back-end servers for different traffic. Traffic for a folder on the web server or for a CDN could be routed to a different back-end. This capability reduces unneeded load on backends that don't serve specific content.

Features of Application Gateways

- **URL-based content routing** - This feature provides the capability to use different back-end servers for different traffic. Traffic for a folder on the web server or for a CDN could be routed to a different back-end. This capability reduces unneeded load on backends that don't serve specific content.
- **Multi-site routing** - Application gateway allows for you to consolidate up to 20 websites on a single application gateway.
- **Websocket support** - Another great feature of Application Gateway is the native support for Websocket.
- **Health monitoring** - Application gateway provides default health monitoring of backend resources and custom probes to monitor for more specific scenarios.
- **SSL Policy and Ciphers** - This feature provides the ability to limit the SSL protocol versions and the ciphers suites that are supported and the order in which they are processed.
- **Request redirect** - This feature provides the capability to redirect HTTP requests to an HTTPS listener.
- **Multi-tenant back-end support** - Application gateway supports configuring multi-tenant back-end services like Azure Web Apps and API Gateway as back-end pool members.
- **Advanced diagnostics** - Application gateway provides full diagnostics and access logs. Firewall logs are available for application gateway resources that have WAF enabled.

Benefits

Application Gateway is useful for:

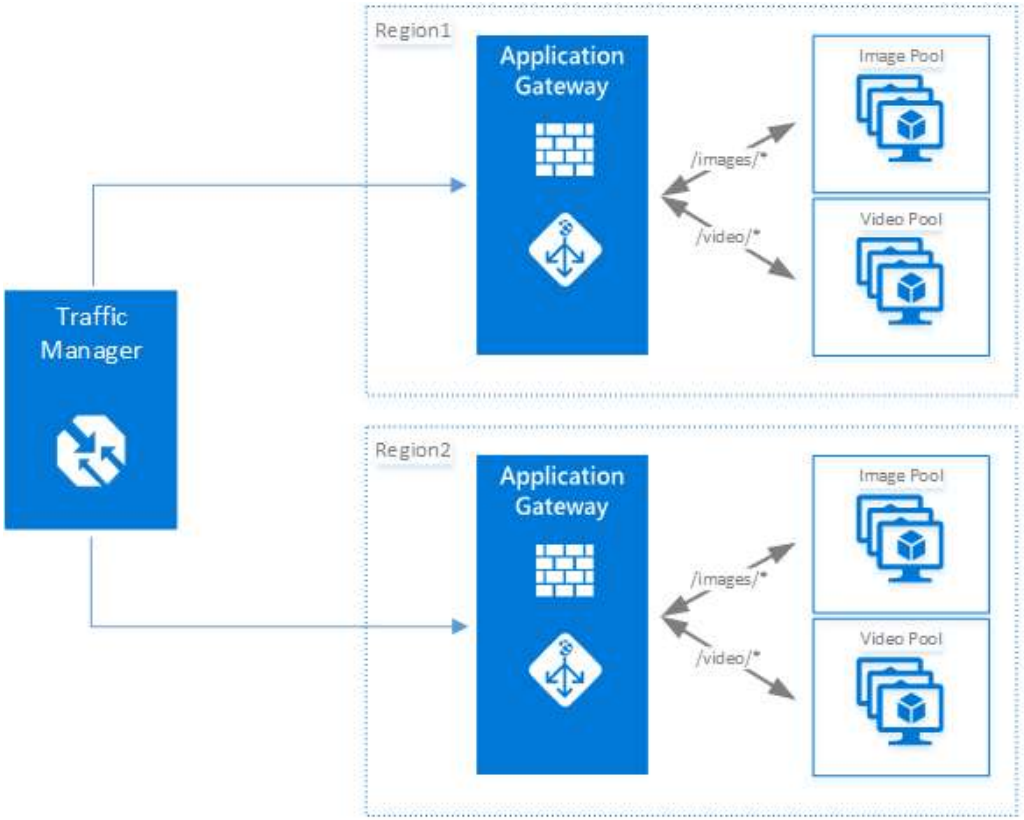
- Applications that require requests from the same user/client session to reach the same back-end virtual machine. Examples of these applications would be shopping cart applications and web mail servers.
- Removing SSL termination overhead for web server farms.
- Applications, such as a content delivery network, that requires multiple HTTP requests on the same long-running TCP connection to be routed or load balanced to different back-end servers.
- Applications that support websocket traffic
- Protecting web applications from common web-based attacks like SQL injection, cross-site scripting attacks, and session hijacks.
- Logical distribution of traffic based on different routing criteria such as, url path or domain headers.

Application Gateway is fully Azure managed, scalable and highly available.

- When you create an application gateway, an endpoint (public VIP or internal ILB IP) is associated and used for ingress network traffic. This VIP or ILB IP is provided by Azure Load Balancer working at the transport level (TCP/UDP) and having all incoming network traffic being load balanced to the application gateway worker instances.
- The application gateway then routes the HTTP/HTTPS traffic based on its configuration whether it's a virtual machine, cloud service, internal or an external IP address.

Interaction with Traffic Manager

- Traffic manager can be used together with Application gateways to complete the following scenario.
- Traffic Manager provides redirection and availability of traffic to multiple application gateway resources in different regions, while application gateway provides cross region layer 7 load balancing.



Load balancing VMs for Highly Available Application

- Load balancing provides a higher level of availability by spreading incoming requests across multiple virtual machines. To create a highly available system we need different components that distribute traffic
  - Virtual network with subnets
  - Azure load balancer
  - Load balancer health probe
  - Load balancer traffic rules
  - Virtual machines attached to a load balancer
  - An application running on virtual machines
- Creation of these resources could be accomplished using PowerShell, Azure Portal, Azure CLI and/or programmatic APIs.
- A typical process of creating highly available system includes steps described in following slides. We will give examples using mostly Azure CLI.

Create Load Balancer

- The following creates a resource group named *myResourceGroupLoadBalancer* in the *eastus* location:

```
az group create --name myResourceGroupLoadBalancer --location eastus
```

- To access your app on the Internet, you need a public IP address for the load balancer. Create a public IP address with `az network public-ip create`:

```
az network public-ip create \
    --resource-group myResourceGroupLoadBalancer \
    --name myPublicIP
```

- Create a load balancer with az network lb create. The following example creates a load balancer named *myLoadBalancer* and assigns the *myPublicIP* address to the front-end IP configuration:

```
az network lb create \
    --resource-group myResourceGroupLoadBalancer \
    --name myLoadBalancer \
    --frontend-ip-name myFrontEndPool \
    --backend-pool-name myBackEndPool \
    --public-ip-address myPublicIP
```

Create a health probe for the Load Balancer

- To allow the load balancer to monitor the status of your app, you use a health probe. The health probe dynamically adds or removes VMs from the load balancer rotation based on their response to health checks. By default, a VM is removed from the load balancer distribution after two consecutive failures at 15-second intervals. You create a health probe based on a protocol or a specific health check page for your app.+
- The following example creates a TCP probe. You can also create custom HTTP probes for more fine grained health checks. When using a custom HTTP probe, you must create the health check page, such as *healthcheck.js*. The probe must return an **HTTP 200 OK** response for the load balancer to keep the host in rotation.
- To create a TCP health probe, you use az network lb probe create. The following example creates a health probe named *myHealthProbe*:

```
az network lb probe create \
    --resource-group myResourceGroupLoadBalancer \
    --lb-name myLoadBalancer \
    --name myHealthProbe \
    --protocol tcp \
    --port 80
```

Crate a load balancer rule

- A load balancer rule is used to define how traffic is distributed to the VMs. You define the front-end IP configuration for the incoming traffic and the back-end IP pool to receive the traffic, along with the required source and destination port. To make sure only healthy VMs receive traffic, you also define the health probe to use.
- Create a load balancer rule with az network lb rule create. The following example creates a rule named *myLoadBalancerRule*, uses the *myHealthProbe* health probe, and balances traffic on port *80*:

```
az network lb rule create \
    --resource-group myResourceGroupLoadBalancer \
    --lb-name myLoadBalancer \
    --name myLoadBalancerRule \
    --protocol tcp \
    --frontend-port 80 \
    --backend-port 80 \
    --frontend-ip-name myFrontEndPool \
    --backend-pool-name myBackEndPool \
    --probe-name myHealthProbe
```

Configure virtual network, create network resources

- Before you deploy some VMs and can test your balancer, create the supporting virtual network resources.
- Create a virtual network with az network vnet create. The following example creates a virtual network named *myVnet* with a subnet named *mySubnet*:

```
az network vnet create \
    --resource-group myResourceGroupLoadBalancer \
    --name myVnet \
    --subnet-name mySubnet
```

- To add a network security group, you use az network nsg create. The following example creates a network security group named *myNetworkSecurityGroup*:

```
az network nsg create \
    --resource-group myResourceGroupLoadBalancer \
    --name myNetworkSecurityGroup
```

- Create a network security group rule with az network nsg rule create. The following example creates a network security group rule named *myNetworkSecurityGroupRule*:

```
az network nsg rule create \
    --resource-group myResourceGroupLoadBalancer \
    --nsg-name myNetworkSecurityGroup \
    --name myNetworkSecurityGroupRule \
    --priority 1001 \
    --protocol tcp \
    --destination-port-range 80
```

Create network resources

- Virtual NICs are created with az network nic create. The following example creates three virtual NICs. (One virtual NIC for each VM you create for your app in the following steps). You can create additional virtual NICs and VMs at any time and add them to the load balancer:

```
for i in `seq 1 3`; do
    az network nic create \
        --resource-group myResourceGroupLoadBalancer \
        --name myNic$i \
        --vnet-name myVnet \
        --subnet mySubnet \
        --network-security-group myNetworkSecurityGroup \
        --lb-name myLoadBalancer \
        --lb-address-pools myBackEndPool
done
```

Create virtual machines, cloud-init.txt config

- In your current shell, create a file named *cloud-init.txt* and paste the following configuration. For example, create the file in the Cloud Shell not on your local machine.

```
#cloud-config
package_upgrade: true
packages:
  - nginx
  - nodejs
  - npm
write_files:
  - owner: www-data:www-data
  - path: /etc/nginx/sites-available/default
    content: |
      server {
        listen 80;
        location / {
          proxy_pass http://localhost:3000;
          proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection keep-alive;
          proxy_set_header Host $host;
          proxy_cache_bypass $http_upgrade;
        }
      }
  - owner: azureuser:azureuser
  - path: /home/azureuser/myapp/index.js
    content: |
      var express = require('express')
      var app = express()
      var os = require('os');
      app.get('/', function (req, res) {
        res.send('Hello World from host ' + os.hostname() + '!')
      })
      app.listen(3000, function () {
        console.log('Hello world app listening on port 3000!')
      })
runcmd:
  - service nginx restart
  - cd "/home/azureuser/myapp"
  - npm init
  - npm install express -y
  - nodejs index.js
```

Create virtual machines

- To improve the high availability of your app, place your VMs in an availability set.
- Create an availability set with az vm availability-set create. The following example creates an availability set named *myAvailabilitySet*:

```
az vm availability-set create \
    --resource-group myResourceGroupLoadBalancer \
    --name myAvailabilitySet
```

- Now you can create the VMs with az vm create. The following example creates three VMs and generates SSH keys if they do not already exist:

```
for i in `seq 1 3`; do
    az vm create \
        --resource-group myResourceGroupLoadBalancer \
        --name myVM$i \
        --availability-set myAvailabilitySet \
        --nics myNic$i \
        --image UbuntuLTS \
        --admin-username azureuser \
        --generate-ssh-keys \
        --custom-data cloud-init.txt \
        --no-wait
done
```

Availability Sets

- Availability and reliability of your Virtual Machine solutions on Azure are assured using a capability called Availability Sets.
- Availability sets ensure that the VMs you deploy on Azure are distributed across multiple isolated hardware clusters. Doing this ensures that if a hardware or software failure within Azure happens, only a subset of your VMs are impacted and that your overall solution remains available and operational.
- An Availability Set is a logical grouping capability that you can use in Azure to ensure that the VM resources you place within it are isolated from each other when they are deployed within an Azure datacenter. Azure ensures that the VMs you place within an Availability Set run across multiple physical servers, compute racks, storage units, and network switches. If a hardware or Azure software failure occurs, only a subset of your VMs are impacted, and your overall application stays up and continues to be available to your customers. Availability Sets are an essential capability when you want to build reliable cloud solutions.

Test load balancer

- Obtain the public IP address of your load balancer with az network public-ip show. The following example obtains the IP address for *myPublicIP* created earlier:

```
az network public-ip show \
    --resource-group myResourceGroupLoadBalancer \
    --name myPublicIP \
    --query [ipAddress] \
    --output tsv
```

- You can then enter the public IP address in to a web browser. Remember - it takes a few minutes for the VMs to be ready before the load balancer starts to distribute traffic to them.
- The app is displayed, including the hostname of the VM that the load balancer distributed traffic to. When you recycle your call to the Load balancer IP address a different IP address will appear showing to you that the load balancer has directed the traffic to another machine in the cluster.

Add a VM to the load balancer

- After performing VM maintenance, or if you need to expand capacity, you can add a VM to the backend address pool with az network nic ip-config address-pool add. The following example adds the virtual NIC for **myVM2** to *myLoadBalancer*:

```
az network nic ip-config address-pool add \
    --resource-group myResourceGroupLoadBalancer \
    --nic-name myNic2 \
    --ip-config-name ipConfig1 \
    --lb-name myLoadBalancer \
    --address-pool myBackEndPool
```

Remove a VM from the Load Balancer

- You can remove a VM from the backend address pool with az network nic ip-config address-pool remove. The following example removes the virtual NIC for **myVM2** from *myLoadBalancer*:

```
az network nic ip-config address-pool remove \
    --resource-group myResourceGroupLoadBalancer \
    --nic-name myNic2 \
    --ip-config-name ipConfig1 \
    --lb-name myLoadBalancer \
    --address-pool myBackEndPool
```

- To see the load balancer distribute traffic across the remaining two VMs running your app you can force-refresh your web browser. You can now perform maintenance on the VM, such as installing OS updates or performing a VM reboot.+
- To view a list of VMs with virtual NICs connected to the load balancer, use az network lb address-pool show. Query and filter on the ID of the virtual NIC as follows:

```
az network lb address-pool show \
    --resource-group myResourceGroupLoadBalancer \
    --lb-name myLoadBalancer \
    --name myBackEndPool \
    --query backendIpConfigurations \
    --output tsv | cut -f4
```