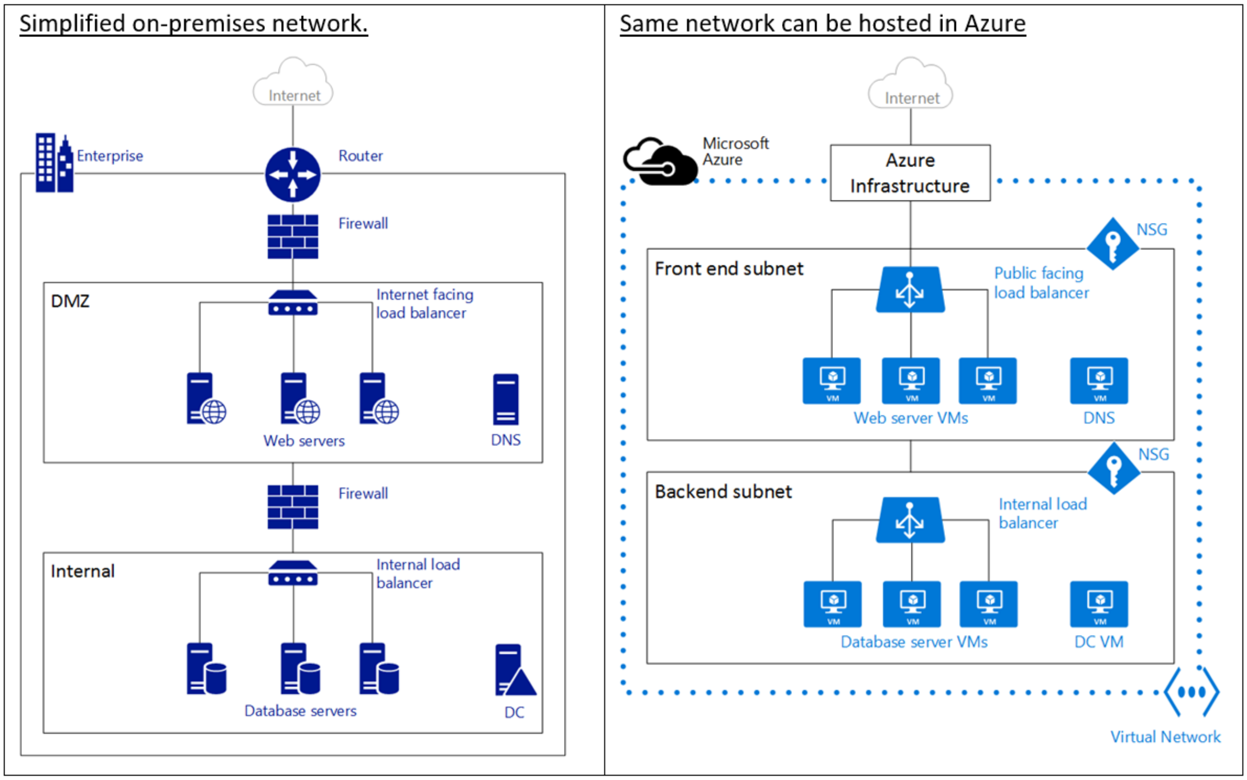
Configure and Manage Azure Virtual Networks

- Overview of Azure Networking
- Virtual Network Benefits
- Understanding Network Resources
- Implement and manage virtual networking
  - Create a VNet using Azure Portal
  - Create a Subnet
  - Configure private and public IP addresses
  - Create Network Interface Card with public, and private IP addresses
  - Create a Virtual Machine
- Setup Network Security Group
  - Create security rules
  - Associate NSG to a subnet or network interface
  - Identify required ports
  - Evaluate effective security rules
  - Application Security Groups
- Understanding Azure DNS
  - Configure Azure DNS
  - DNS Zones
  - DNS Records and Record Sets
  - DNS Resolution
- Network Routing Table
  - System Routes
  - User Defined Routes
  - Creating User Defined Route Table
  - Create and Associate Route
- Create connectivity between virtual networks
  - Overview
  - Create a Point to Site VPN
  - Create and configure VNET to VNET
  - Verify virtual network connectivity
  - Create and Configure VNET peering

Overview of Azure Networking

- An Azure virtual network (VNet) is a representation of your own network in the cloud.
- It is **a logical isolation** of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network.
- You can also further segment your VNet into **subnets** and launch Azure virtual machines (VMs).
- You can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.



Simplified on-premises network.

Same network can be hosted in Azure

*In computer **networks**, a **DMZ** (**demilitarized zone**) is a physical or logical **sub-network** that separates an internal local area **network** (LAN) from other untrusted **networks**, usually the Internet.

Notice how the Azure infrastructure takes on the role of the router, allowing access from your VNet to the public Internet without the need of any configuration. Firewalls can be substituted by Network Security Groups (NSGs) applied to each individual subnet. And physical load balancers are substituted by internet facing and internal load balancers in Azure.

Azure VNet Pricing:

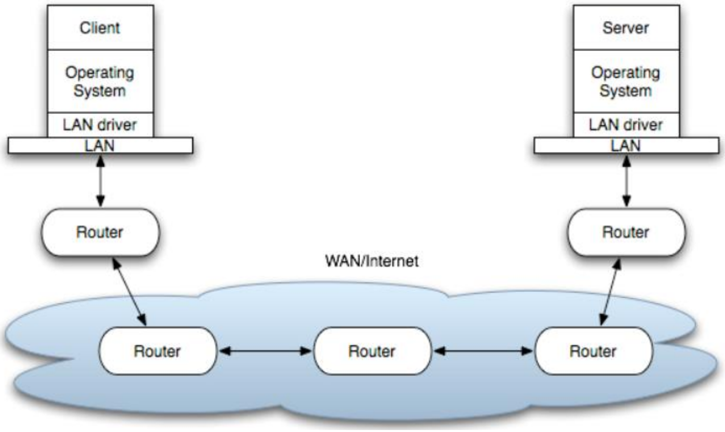
- There is **no extra cost** for using Virtual Networks in Azure.
- The compute instances launched within the Vnet will be charged the standard rates as described in Azure VM Pricing.
- The VPN Gateways and Public IP Addresses used in the VNet will also be charged standard rates.

Virtual Network Characteristics

- **Isolation**. VNets are completely isolated from one another. That allows you to create disjoint networks for **development, testing, and production** that use the same CIDR address blocks.
- **Connectivity**. VNets can be connected to each other, and even to your on-premises datacenter, by using a site-to-site VPN connection, or ExpressRoute connection.
- **Access to the public Internet**. All VMs in a VNet can access the public Internet by default. You can control access by using Network Security Groups (NSGs).
- **Security**. Traffic entering and exiting the virtual machines in a VNet can be controlled using Network Security groups and Azure Firewall.
- **Access to VMs within the VNet**. VMs can be launched in the same virtual network and they can connect to each other using private IP addresses even if they are in different subnets without the need to configure a gateway or use public IP addresses.
- **Name resolution**. Azure provides internal name resolution for IaaS VMs deployed in your VNet. You can also deploy your own DNS servers and configure the VNet to use them.

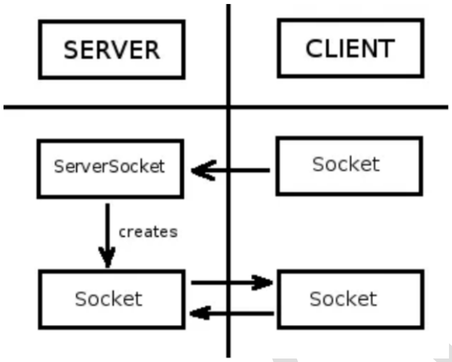
Note: The most important thing about Azure virtual networks is that you cannot add an existing virtual machine to a newly created virtual network. It is important that if you want to leverage virtual networking in Azure that you must create the virtual networks **BEFORE** creating your virtual machines! Don’t miss this important step. You’ll be disappointed if you’ve spent a lot of time setting up a virtual machine and later find that you can’t move it to a virtual network.

OSI Layers in Network Communication

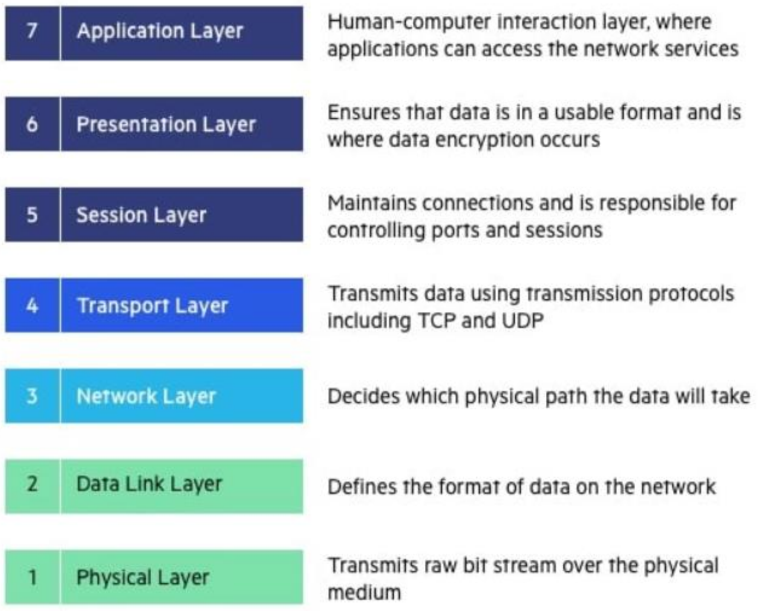


Socket = IP + Port No

HTTP = 80, HTTPS = 443, FTP=21…SQL Server DB = 1433, RDP=3389, SSH=22…



OSI Layers:



7	Application Layer	Human-computer interaction layer, where applications can access the network services
6	Presentation Layer	Ensures that data is in a usable format and is where data encryption occurs
5	Session Layer	Maintains connections and is responsible for controlling ports and sessions
4	Transport Layer	Transmits data using transmission protocols including TCP and UDP
3	Network Layer	Decides which physical path the data will take
2	Data Link Layer	Defines the format of data on the network
1	Physical Layer	Transmits raw bit stream over the physical medium

Understanding Network Resources

- **IP addresses**: There are two types of IP addresses assigned to resources in Azure: *public* and *private*.
  a. **Public IP Addresses** allow Azure resources to communicate with Internet and other Azure public-facing services like Azure Redis Cache.
  b. **Private IP Addresses** allows communication between resources in a virtual network, along with those connected through a VPN, without using an Internet-routable IP addresses.

Preferred IP Series for Intranets (Private IP):

Small Network1: 192.168.0.X – for 2^8 Systems – IP Address Range = 192.168.0.0/24 (Only last byte changes)

Small Network2: 192.168.1.X –for 2^8 Systems – IP Address Range = 192.168.1.0/24 (Only last byte changes)

Large Network: 172.16.X.X – for 2^{16} Systems - IP Address Range = 172.16.0.0/16 (last 2 bytes change)

Very Large Network: 10.X.X.X – for 2^{24} Systems – IP Address Range = 10.0.0.0/8 (last 3 bytes change)

Classless Inter-Domain Routing (CIDR) notation is a compact representation of an IP address and its associated routing prefix. The **notation** is constructed from an IP address, a slash ('/') character, and a decimal number. The number is the count of leading 1 bits in the routing mask, traditionally called the network mask.

Public IP Addresses

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

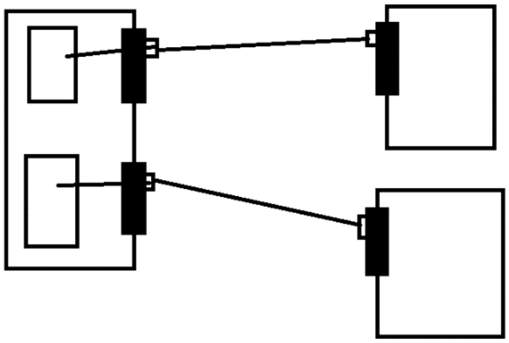
  - In the **dynamic** allocation method the IP address is **not** allocated at the time of its creation. Instead, the public IP address is allocated when you start (or create) the associated resource (like a VM or load balancer). The IP address is released when you stop (or delete) the resource. This means the IP address can change.
  - In the **static** allocation method the IP address for the associated resource does not change. In this case an IP address is assigned immediately. It is released only when you delete the resource or change its allocation method to *dynamic*.
- Public IP addresses allow Azure resources to communicate with Internet and Azure public-facing services such as Azure Redis Cache, Azure Event Hubs, SQL databases and Azure storage.
- In Azure Resource Manager, a public IP address is a resource that has its own properties. You can associate a public IP address resource with any of the following resources:
  - Internet-facing Virtual machines (VM)
  - Internet-facing load balancers
  - VPN gateways
  - Application gateways
- Public IP address is paid service.

Private IP Addresses

1. IP address is allocated from the address range of the subnet to which the resource is attached.
2. The default allocation method is dynamic, where the IP address is automatically allocated from the resource's subnet (using DHCP). This IP address can change when you stop and start the resource.
3. You can set the allocation method to static to ensure the IP address remains the same. In this case, you also need to provide a valid IP address that is part of the resource's subnet.
4. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address.
5. In the Azure Resource Manager deployment model, a private IP address is associated to the following types of Azure resources:
   - VMs
   - Internal load balancers (ILBs)
   - Application gateways

- **Subnets**: Subnet is a **range of IP addresses** in the VNet, you can divide a VNet into multiple subnets for organization and security. VMs deployed to subnets (same or different) within a VNet can communicate with each other without any extra configuration. You can also configure route tables and NSGs to a subnet.

- **Network Interface Card (NIC):** VMs communicate with other VMs and other resources on the network by using virtual network interface card (NIC). Virtual NICs configure VMs with private and optional public IP address. VMs can have more than one NIC for different network configurations.
  Note: VMs can have more than one NIC adapter that links the VM with the virtual network. The number of NICs you can attach to a VM depends on its size. For example, a VM that is based on a D2 size can have 2 NICs, and a D4-based VM can have a maximum of 16 NICs. Multiple NICs configuration is common for virtual appliances that provide additional control of traffic in virtual networks.



- **Network Security Group** (NSG): You can create NSGs to control **inbound and outbound** access to network interfaces (NICs), VMs, and subnets. Each NSG contains one or more rules specifying whether or not traffic is **allowed or denied** based on **protocol, source IP address, source port, destination IP address, and destination port.**

- **VPN Gateways:** Azure VPN Gateway is used to connect an Azure virtual network (VNet) to other Azure VNets or to an on-premises network. You need to assign a public IP address to its IP configuration to enable it to communicate with the remote network. Currently, you can only assign a ***dynamic*** **public IP** address to a VPN gateway.

- **Azure DNS:** The Domain Name System (DNS) enables clients to resolve user-friendly fully qualified domain names (FQDNs), such as www.adatum.com, to IP addresses**.** Azure Domain Name System (DNS) allows you to host your domains with your Azure apps. By hosting your domains in Azure, you can manage your DNS records by using your existing Azure subscription.

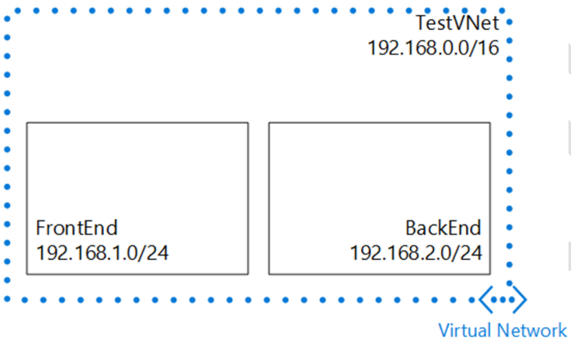
- . . .

Create a Virtual Network (VNet) using the Azure portal

In this scenario we will create a VNet named **TestVNet** with a reserved CIDR block of **192.168.0.0/16**.

Your VNet will contain the following **subnets**:

- **FrontEnd**, using **192.168.1.0/24** as its CIDR block.
- **BackEnd**, using **192.168.2.0/24** as its CIDR block.



1. Click Search → Virtual network → **+Add →**

2. Name=Demo-east-vnet, Address Space=192.162.0.0/16, Subnet name=”**Frontend-subnet**”, Subnet Address Range=192.168.1.0/24, Select Resource Group → Create

3. Wait for the VNet to be created, → **Virtual network** blade, click **All settings** → **Subnets** → **Add** a new Subnet. (Name=**Backend-subnet**, Address space=192.168.2.0/24, Leave NSG and Route table=None → OK.

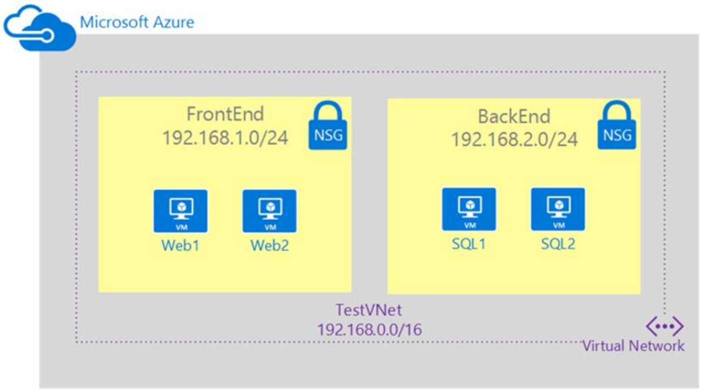
Network Security Group

NSGs are simple, stateful packet inspection devices that use the 5-tuple (the source IP, source port, destination IP, destination port, and layer 4 protocol) approach to create **allow/deny rules** for network traffic. You allow or deny traffic to and from a single IP address, to and from multiple IP addresses, or to and from entire subnets.

In this scenario you will create an NSG for each subnet in the **Demo-vnet** virtual network, as described below:

- **Frontend-nsg**. The front end NSG will be applied to the *FrontEnd* subnet, and contain two rules:
  - **rdp-allow**. This rule will allow RDP (3389) traffic to the *FrontEnd* subnet.
  - **web-allow**. This rule will allow HTTP (80) traffic to the *FrontEnd* subnet.
- **Backend-nsg**. The back end NSG will be applied to the *BackEnd* subnet, and contain two rules:

- **sql-allow**. This rule allows SQL (1433) traffic only from the *FrontEnd* subnet.
- **Rdb-allow**: This rule will allow RDP (80) traffic to the *BackEnd* subnet
- **web-deny**. This is Outbound rule **denies all internet bound** traffic **from** the *BackEnd* subnet.



4. **Create NSG for Frontend:** Browse → Network Security Groups → Add → Name=**Frontend-nsg** → Create
   a. Select Frontend-nsg → Settings →
      i. Inbound security rules → Add, Name=**AllowHTTP**, priority, Priority=1000, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**80**, Action=Allow → OK
      ii. Inbound security rules → Add, Name=**AllowRDP**, priority, Priority=1001, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**3389**, Action=Allow → OK
   b. Associate the NSG to the FrontEnd subnet
      i. Select Test-vnet → Settings → Subnets → Frontend-subnet → Network security group → Select Frontend-nsg → Save
5. **Create NSG for Backend:** Browse → Network Security Groups → Add → Name=Backend-nsg → Create
   a. Select Backend-nsg → Settings →
      i. Inbound security rules → Add, Name=**AllowSQL**, priority, Priority=1001, Source=**CIDR block**, **Source IP address range=192.168.1.0/24,** Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=**1433**, Action=Allow → OK
      ii. Inbound security rules → Add, Name=**AllowRDP**, priority, Priority=1002, Source=Any, Source port range=*, Protocol=**TCP**, Destination=Any, Destination port range=3389, Action=Allow → OK

iii. **Outbound** security rules → Add, Name=**DenyWeb**, priority, Priority=1000, Destination=**Tag**, destination Tag=**Internet**, Destination port range=80, Source=**Any**, Protocol=**Any**, Source port range=*, Action=**Deny** → OK

b. Associate the NSG to the BackEnd subnet

i. Select Test-vnet → Settings → Subnets → Backend-subnet → Network security group → Select Backend-nsg → Save

Summary:

Virtual Network (192.168.0.0/16)

Frontend-subnet (192.168.1.0/24)

Frontend-nsg

Allow RDP / HTTP (Inbound)

Backend-subnet (192.168.2.0/24)

Backend-nsg

Allow RDP / SQL (Inbound)

Deny Internet (Outbound)

Creating a Virtual Machine

6. Azure portal → On the Hub menu, click New → Compute → Windows Server 2022 Datacenter.

Note: To find additional images, click Marketplace and then search or filter for available items.

7. On the **Windows Server 2019 Datacenter** page, under Select a deployment model = Resource Manager → Create.

8. Create virtual machine blade →

a. Basics → provide values for Name, Username and Password, Resource Group → OK

b. Size → Select an appropriate virtual machine size for your needs. Note that Azure recommends certain sizes automatically depending on the image you choose.

c. Settings to see storage and networking settings for the new virtual machine.

i. NSG = None

d. Click Summary to review your configuration choices.

9. Click Create

Create the following two VM

Demo1-vm

DemoVM1-nic (name provided by Azure)

Demo1-vm-publicIP

RDP into the machines and install IIS Web Server in both.

Summary:

Demo-VNet

Frontend-subnet

Frontend-nsg

Allowed HTTP and RDP

Demo-vm

NO NSG (NIC Level)

Remote Login and installed IIS

edit wwwroot\iisstart.png - Added ONE

Web1-vm-ip

DNS Name

Backend-sub

Backend-nsg

Allowed RDP Inbound

Denied Internet: OutBound

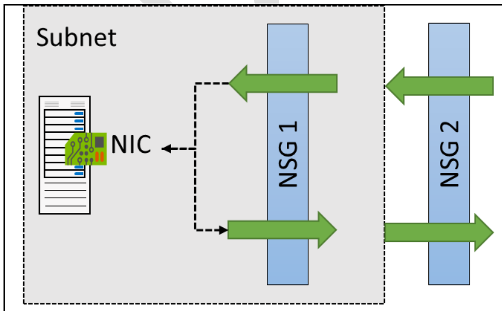
Accessed

Web1 http://<ip> or http://<dnsname>

Web2 http://<ip> or http://<dnsname>

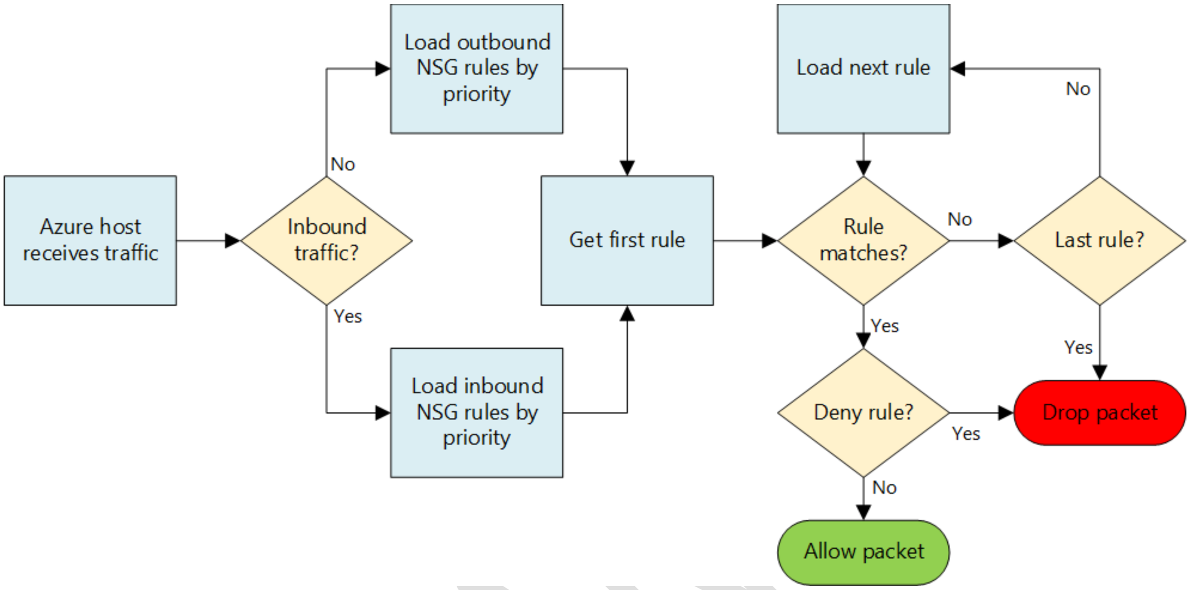
NSG: Evaluate effective security rules

Be very careful when you want to apply NSG to both VM (NIC) and subnet level at the same time. NSGs are evaluated independently, and an “allow” rule must exist at **both levels** otherwise traffic will not be admitted.



If there was incoming traffic on port 80, you would need to have the NSG at subnet level ALLOW port 80, and you would also need another NSG with ALLOW rule on port 80 at the NIC level. For incoming traffic, the NSG set at the subnet level is evaluated first, then the NSG set at the NIC level is evaluated. For outgoing traffic, it is the converse.

The picture below should even clarify this concept more: you can see how rules are evaluated for network packets, once again remember that you need to **evaluate this diagram two times**: once for subnet level NSG rules, and once for NIC level NSG rules.



To see the Effective Rules:

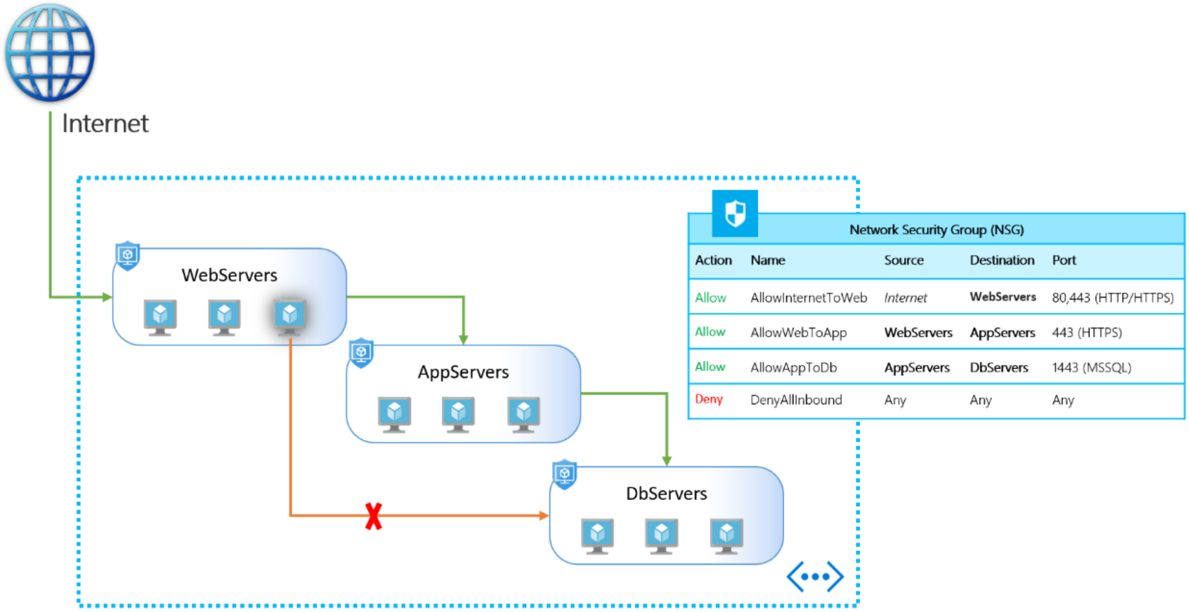
Select the VM → Settings → Networking → Click on **Effective security rules**

Now you get an overview which NSGs are associated with the VM's NIC and which rules are applied to it.

For an offline analysis there is a download option, that generates a CSV file of the output.

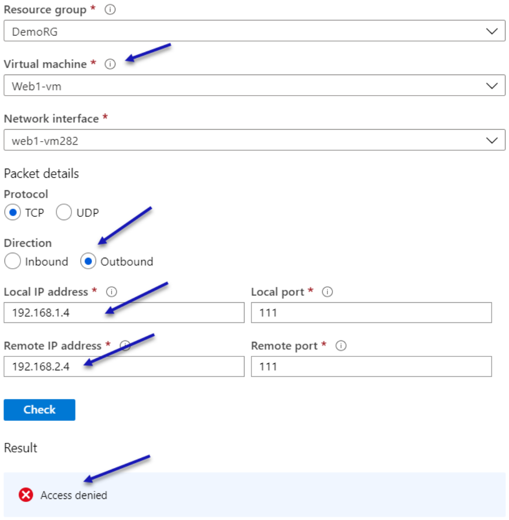
Application Security Groups

Application security groups enable you to configure network security as a natural extension of an application's structure, allowing you to group virtual machines and define network security policies based on those groups. This feature allows you to reuse your security policy at scale without manual maintenance of explicit IP addresses. The platform handles the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic.



Network Security Group (NSG)				
Action	Name	Source	Destination	Port
Allow	AllowInternetToWeb	*Internet*	WebServers	80,443 (HTTP/HTTPS)
Allow	AllowWebToApp	WebServers	AppServers	443 (HTTPS)
Allow	AllowAppToDb	AppServers	DbServers	1443 (MSSQL)
Deny	DenyAllInbound	Any	Any	Any

1. Create two new Application Secuirty Groups
   - WebServers-asg
   - DbServers-asg
2. Attach them to respective VM: VM → Networking → Application Securty Group tab
3. Use **Network Watcher** and note that **IP Flow verify** is **success** from Web1-vm to Db1-vm
4. Create an NSG **outbound rule** to **deny** traffic from WebServers to DbServers
5. Wait for couple of minutes.
6. Use **Network Watcher** and note that **IP Flow verify** is **failed** from Web1-vm to Db1-vm



Resource group * ⓘ
DemoRG

Virtual machine * ⓘ
Web1-vm

Network interface *
web1-vm282

Packet details

Protocol
● TCP ○ UDP

Direction
○ Inbound ● Outbound

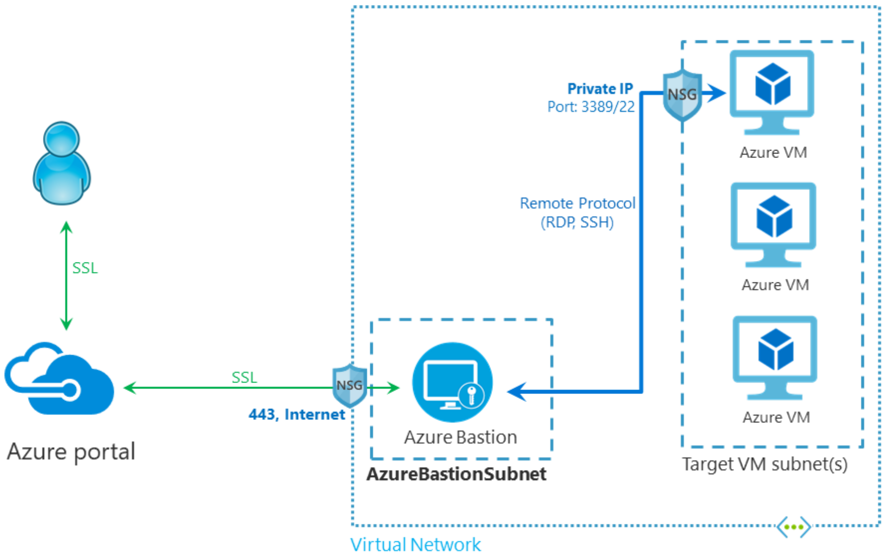
Local IP address * ⓘ	Local port * ⓘ
192.168.1.4	111
Remote IP address * ⓘ	Remote port * ⓘ
192.168.2.4	111

Result

⊗ Access denied

Azure Bastion

Azure Bastion is a fully managed PaaS service that provides secure and seamless RDP and SSH access to your virtual machines directly through the Azure Portal. Azure Bastion is provisioned directly in your Virtual Network (VNet) and supports all VMs in your Virtual Network (VNet) using SSL without any exposure through public IP addresses.



In this diagram:

- The Bastion host is deployed in the virtual network.
- The user connects to the Azure portal using any HTML5 browser.
- The user selects the virtual machine to **connect** to.
- With a single click, the RDP/SSH session opens in the browser.
- No public IP is required on the Azure VM.

The following features are available:

- **RDP and SSH directly in Azure portal:** You can directly get to the RDP and SSH session directly in the Azure portal using a single click seamless experience.
- **Remote Session over SSL and firewall traversal for RDP/SSH:** Azure Bastion uses an HTML5 based web client that is automatically streamed to your local device, so that you get your RDP/SSH session over SSL on port 443 enabling you to traverse corporate firewalls securely.
- **No Public IP required on the Azure VM:** Azure Bastion opens the RDP/SSH connection to your Azure virtual machine using private IP on your VM. You don't need a public IP on your virtual machine.

- **No hassle of managing NSGs:** Azure Bastion is a fully managed platform PaaS service from Azure that is hardened internally to provide you secure RDP/SSH connectivity. You don't need to apply any NSGs on Azure Bastion subnet. Because Azure Bastion connects to your virtual machines over private IP, you can configure your NSGs to allow RDP/SSH from Azure Bastion only. This removes the hassle of managing NSGs each time you need to securely connect to your virtual machines.
- Azure Bastion can support up to **25 concurrent RDP**, this is still dependent on the Azure Virtual Machines. Azure Virtual Machine doesn’t support more than 2 concurrent RDP connections and these must be from two different user accounts.

Create a bastion host

1. Azure Vnet → Subnet → Create a **New Subnet** by name **AzureBastionSubnet** (You must use a subnet of at least /27 or larger eg: /26, /25 and …)
2. Azure Portal → + New → Bastion

Connect to VM

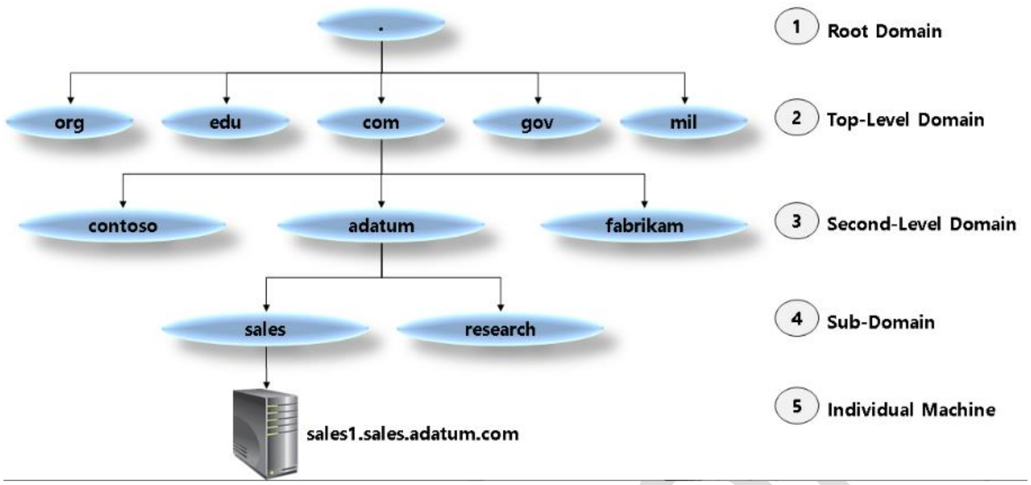
3. Azure VM → **Connect** → **Bastion** → Provide the RDP Username and Password → Connect

Azure DNS

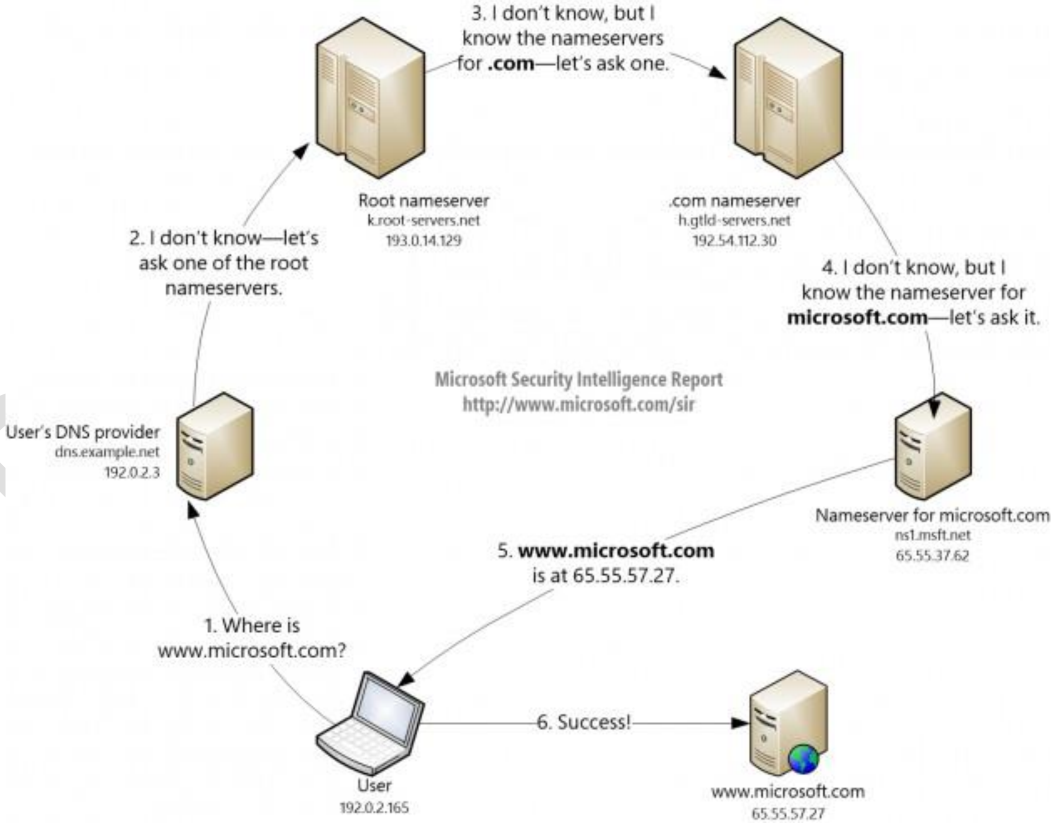
- The Domain Name Service, or DNS, is responsible for translating (or resolving) a website or service name to its IP address.
- Azure DNS is a hosting service for DNS domains, providing name resolution using Microsoft Azure infrastructure.
- Applications requiring automatic DNS management can integrate with the service via the REST API and SDKs.
- When you add a new DNS record, the Azure DNS name servers are updated in a few seconds so you don’t have to wait long before that DNS record can be used.
- Azure DNS does not currently support purchasing of domain names.

DNS Domains:

The DNS is a hierarchy of domains. The hierarchy starts from the ‘root’ domain, whose name is simply '.'. Below this come top-level domains, such as ‘com’, 'net', ‘org’, 'uk' or ‘jp’. Below these are second-level domains, such as 'org.uk' or ‘co.jp’. The domains in the DNS hierarchy are globally distributed, hosted by DNS name servers around the world.



DNS Resolution: To answer queries, it uses aspecial type of DNS record called a Name Server (NS)record. For example, the root zone contains NS records for ‘com’and shows the name servers for the ‘com’ zone. In turn,the ‘com’ zone contains NS records for ‘contoso.com’, whichshows the name servers for the ‘contoso.com’ zone. Settingup the NS records is called delegating the domain.


3. I don't know, but I know the nameservers for .com—let's ask one.
Root nameserver
k.root-servers.net
193.0.14.129
.com nameserver
h.gtld-servers.net
192.54.112.30
2. I don't know—let's ask one of the root nameservers.
4. I don't know, but I know the nameserver for microsoft.com—let's ask it.
Microsoft Security Intelligence Report
http://www.microsoft.com/sir
User's DNS provider
dns.example.net
192.0.2.3
Nameserver for microsoft.com
ns1.msft.net
65.55.37.62
5. www.microsoft.com is at 65.55.57.27.
1. Where is www.microsoft.com?
6. Success!
User
192.0.2.165
www.microsoft.com
65.55.57.27

How DNS Server Works

In browser http://www.bestazuretraining.com

1. Browser will send request to DNS Server as configured in your machine for finding IP of www.bestazuretraining.com
2. DNS if has IP - It immediately returns
3. DNS does'nt have IP - It will send the request to ROOT Name Server
4. Root Name Server will query -> .com Name Server
5. .com Name Server will sent the request bestazuretraining.com name server
6. In bestazuretraining.com Name Server it will search for the required Recordset and return the value...
7. If IP is returned browser will directly send the request to target machine...
8. If Alias (CName) is returned then it again starts from Step 2...

DNS Zone:

A DNS zone is used to host the DNS records for a particular domain. In order to start hosting your domain, you need to create a DNS zone. Any DNS record created for a particular domain will be inside a DNS zone for the domain.

For example, the domain "contoso.com" may contain a number of DNS records, such as "mail.contoso.com" (for a mail server) and "www.contoso.com" (for a web site).

About DNS Zone names

2. The name of the zone must be unique within the resource group, and the zone must not exist already. Otherwise, the operation will fail.
3. The same zone name can be re-used in a different resource group or a different Azure subscription.
4. Where multiple zones share the same name, each instance will be assigned different name server addresses.
5. **Only one set of addresses can be configured with the domain name registrar.**

Steps to Create a DNS Zone and Map Name to IP Address:

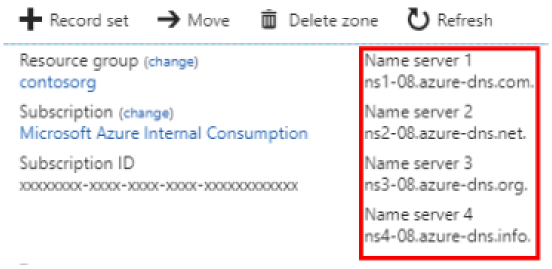
1. Buy a Domain Name from a Registrar (eg: godaddy.com is registrar)

2. Azure Portal → New → Networking → **DNS zone**

3. Name = deccansoft.net, Provide other details → Create

4. Goto Registrar Website → Login

5. DNS Delegation: Map domain Name Server to NS records for the DNS Zone created



Resource group (change)
contosorg

Subscription (change)
Microsoft Azure Internal Consumption

Subscription ID
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Name server 1
ns1-08.azure-dns.com.

Name server 2
ns2-08.azure-dns.net.

Name server 3
ns3-08.azure-dns.org.

Name server 4
ns4-08.azure-dns.info.

6. Select the DNS Zone → **+ Record set** → Name=www, Type=”A”, TTL=1, IP Address=<Public IP of VM Created> → OK

DNS Record Type:

Record Type	Full Name	Function
A (IPv4) AAAA (IPv6)	Address	Maps a host name such as mail.adatum.com to an IP address, such as 131.107.10.10.
CNAME	Canonical name	Points one host record, such as adatum.ftp.adatum.com, to another host record, such as mail.lucernepublishing.com, or even another host record in another domain, such as www.contoso.com.
MX	Mail exchange	Points to the host that will receive mail for that domain. MX records must point to an A record, not to a CNAME record.
NS	Name server	Delegates a DNS zone to the specified authoritative name server.
SOA	Start of Authority	Defines the authoritative record for the zone.
SRV	Service	Locates hosts that are providing specific services, such as the Session Initiation Protocol (SIP) endpoint.
TXT	Text	Records a human-readable text field in DNS.

To Test the name resolution

- **Ipconfig** /all
- **ping** <host name>
- **nslookup** <host name> <name server name>
- **nslookup** www.bestazuretraining.com ns1-01.azure-dns.com

Private DNS Zones

https://docs.microsoft.com/en-us/azure/dns/private-dns-getstarted-cli

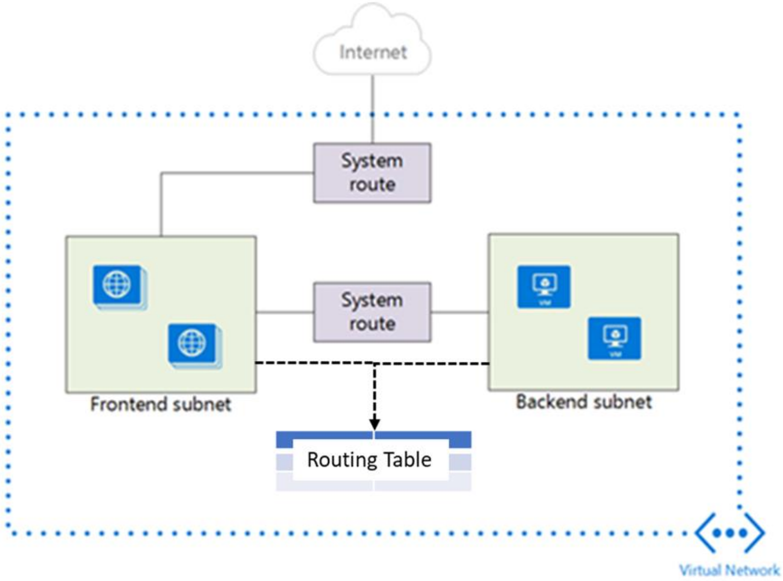
- A Private DNS Zone can be connected to multiple Virtual Networks (in any ADTenant/Subscriptions/Region).
- Private DNS Zone is a Global Services (Independent of Region)
- Here Domain Names are mapped Private IP Address.

Network Route Table

- When you add virtual machines (VMs) to a virtual network (VNet) in Azure, you will notice that the VMs are able to communicate with each other over the network, automatically. You do not need to specify a gateway, even though the VMs are in different subnets. The same is true for communication from the VMs to the public Internet, and even to your on-premises network when a hybrid connection from Azure to your own datacenter is present.
- This flow of communication is possible because Azure uses a series of **system routes** to define how IP traffic flows.

System routes control the flow of communication in the following scenarios:

- From within the same subnet.
- From a subnet to another within a VNet.
- From VMs to the Internet.
- From a VNet to another VNet through a VPN gateway.
- From a VNet to another VNet through VNet Peering (Service Chaining).
- From a VNet to your on-premises network through a VPN gateway.



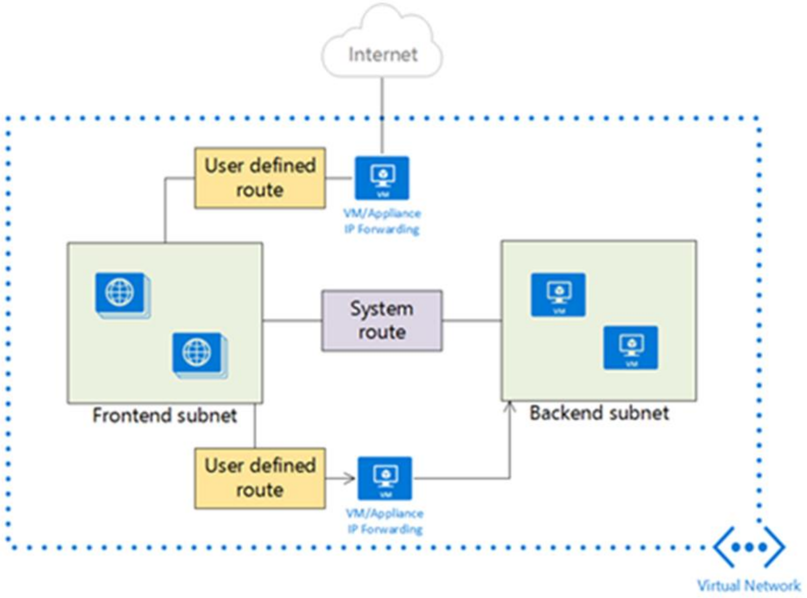
Information about the **system routes** is recorded in a **route table**. A route table contains a set of **rules**, called **routes**, that specifies how packets should be routed in a virtual network. Route tables are **associated to subnet**s, and each packet leaving a subnet is handled based on the associated route table. Packets are matched to routes using the destination. The destination can be an **IP address, a virtual network gateway, a virtual appliance, or the internet**. If a matching route can't be found, then the packet is **dropped.**

User Defined Routes

For most environments you will only need the system routes already defined by Azure.

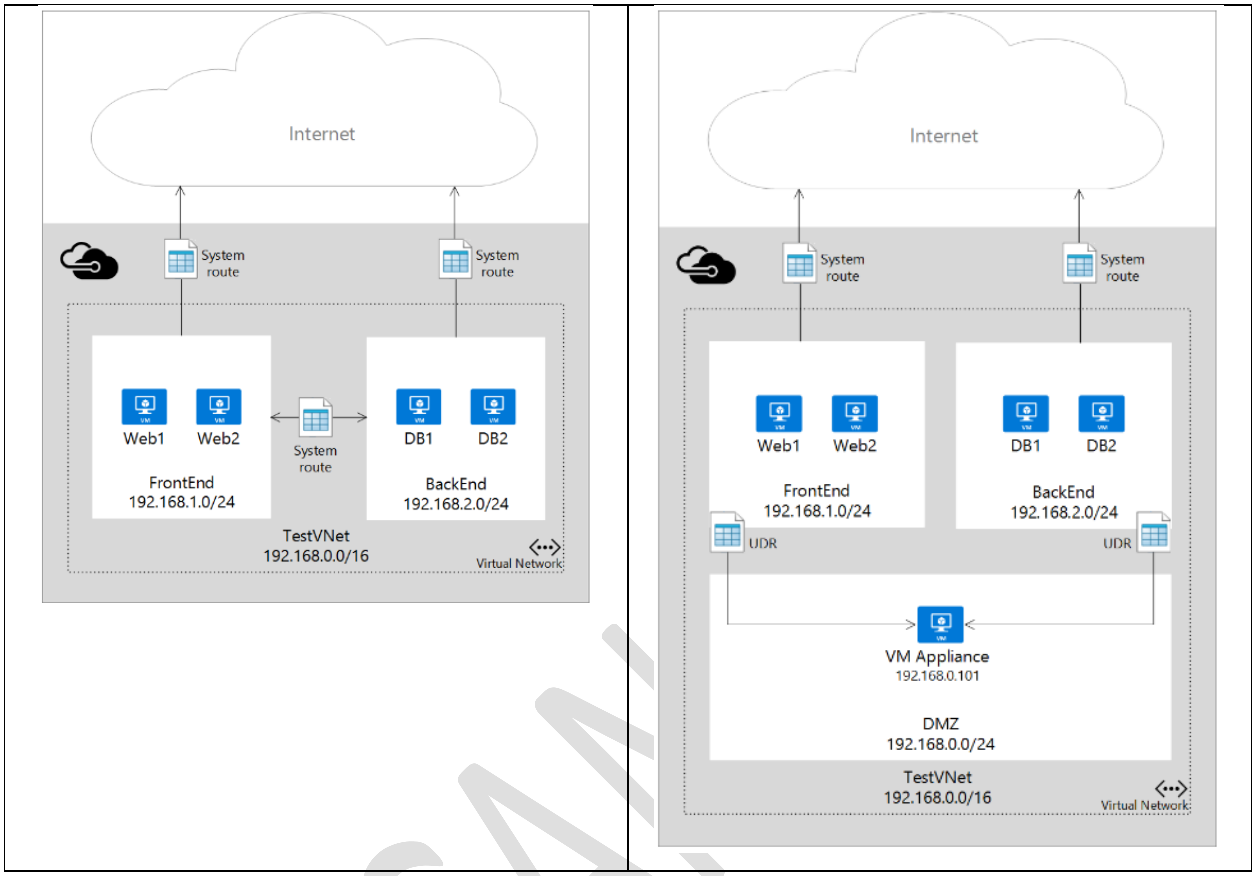
However, you may need to create a route table and add one or more routes in specific cases, such as:

- Use of virtual appliances in your Azure environment.
- Force tunneling to the Internet via your on-premises network.



Each route table can be associated to multiple subnets, but a subnet can only be associated to a single route table.

There are no additional charges for creating route tables in Microsoft Azure.



- User defined routes are only applied to **traffic leaving a subnet**. You cannot create routes to specify how traffic comes into a subnet from the Internet, for instance. Also, the appliance you are forwarding traffic to cannot be in the same subnet where the traffic originates. **Always create a separate subnet for your appliances**.
- NVAs are VMs that help with network functions like routing and firewall optimization. Some of the cases where virtual appliances can be used include:
  - Monitoring traffic with an intrusion detection system (IDS).
  - Controlling traffic with a firewall.
- This **virtual appliance VM** must be able to receive incoming traffic that is not addressed to itself. To allow a VM to receive traffic addressed to other destinations, you must **enable IP Forwarding** for the VM. This is an Azure setting, not a setting in the guest operating system.
- You can have multiple route tables, and the same route table can be associated to one or more subnets. And each subnet can only be associated to a single route table.

NOTE: An **intrusion detection system** (IDS) is a device or software application that monitors a network or systems for **malicious activity or policy violations**. Any malicious activity or violation is typically reported either to an administrator or collected centrally using a security information and event management (SIEM) system. A SIEM system combines outputs from multiple sources, and uses alarm filtering techniques to distinguish malicious activity from false alarms.

The most common classifications are **network intrusion detection systems** (**NIDS**) and **host-based intrusion detection systems** (HIDS).

Network security capabilities of virtual network security appliances include:

- Firewalling
- Intrusion detection/intrusion prevention
- Vulnerability management
- Application control
- Network-based anomaly detection
- Web filtering
- Antivirus
- Botnet protection

To find available Azure virtual network security appliances, go to the Azure Marketplace and search for "security" and "network security."

Example of Virtual Appliance: Palo Alto Networks VM-Series.

Create User Defined Routes (UDR) :

1. Create a **New Subnet (name=VirtualAppliance-subnet)** with Address Prefix 192.168.4.0/24.
2. Create a **new VM** (Demo-va) to be used for virtual appliance with private IP address **192.168.4.4** (preferably static ip)

UDR for Frontend Subnet when target is any VM in backend subnet

3. Create UDR: Search Bar → **Route table** → + Add
4. Set Name=**Frontend-udr**. . . → Create
5. Virtual network gateway route propagation = Enabled (default)

Border Gateway Protocol (**BGP**): An on-premises network gateway can exchange routes with an Azure virtual network gateway using the BGP. Routes are automatically added to the route table of all subnets with BGP propagation enabled.

6. Select Route table → Routes → + Add
7. Set Name=**Frontend-to-Backend-Subnet-route**,
   [Destination] Address prefix=192.168.2.0/24 (Range of Backend Subnet),
   Next hop type=**Virtual appliance**,
   Next hop address = **192.168.4.4** (Private IP of VM Appliance - Demo-va)

Routing Algorithms:

a) If multiple routes contain the **same address prefix**, Azure selects the route type, based on the following priority:

1. User-defined route
2. BGP route
3. System route

b) Longest prefix match algorithm

For example, if the destination address is 10.0.0.5 and there are two routes: One route specifies the 10.0.0.0/24 address prefix, while the other route specifies the 10.0.0.0/16 address prefix. In this case, Azure selects a route using the longest prefix match algorithm, which is the 10.0.0.0/24 route.

c) A route with the **0.0.0.0/0** address prefix instructs Azure how to route traffic destined for an IP address that is not within the address prefix of any other route in a subnet's route table.

8. Select Frontend-udr-table → Subnets ➔ **+Associate** → Select **Frontend-subnet**

For the VM in New Subnet (used for Virtual Appliance):

9. In Portal = Enable IP Forwarding for NIC of **Demo-va** VM.
   1. Goto Virtual Appliance machine → **Networking** → Click on **Network Interface Card** (eg: **web3-vm126**)
   2. IP Configuration → **IP Forwarding = Enable**

10. Turn on **IP forwarding** within **Virtual Appliance VM** Operating System.
    1. RDP to Virtual Appliance VM → PowerShell
    2. Execute the following command

```
Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name
IpEnableRouter -Value 1
```

3. **Restart the Virtual Appliance VM**

11. In target VM (with PrivateIP 192.168.2.4), Enable Internet Control Message Protocol (ICPM) which the Windows Firewall denies by default.
    1. RDP to VM (In Backend subnet) → PowerShell
    2. Execute the command on VM's

       ```
       New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
       ```

12. Test the routing of network traffic
    1. RDP to Source VM (Frontend subnet) → PowerShell
    2. Execute the following command

       tracert <Target VM Name from Backend-subnet>
    3. Note that the first hop is Virtual Appliance VM and send hop to the target VM

Rules Explained:

https://docs.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview

Azure Firewall

Controlling **outbound network access** is an important part of an overall network security plan. For example, you may want to **limit access to web sites**. Or, you may want to limit the outbound IP addresses and ports that can be accessed.

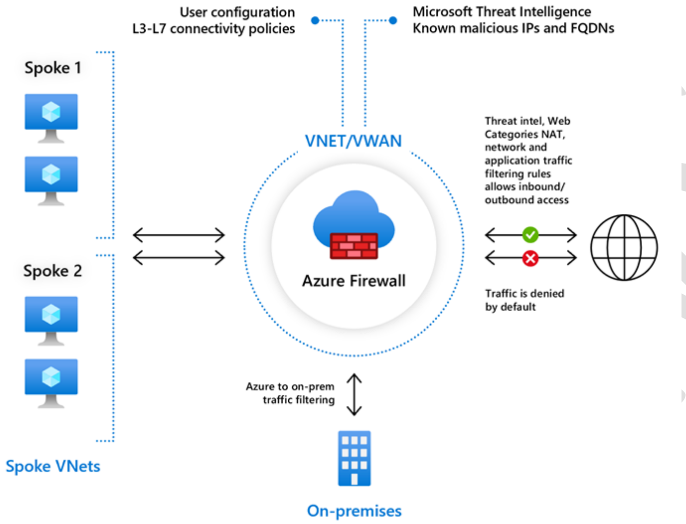
With Azure Firewall, you can configure:

- **Application rules** that define fully qualified domain names (FQDNs) that can be accessed from a subnet (having a route table)
- **Network rules** that define source address, protocol, destination port, and destination address (same NSG)
- **NAT rules**

Key Features of Azure Firewall:

- Azure Firewall is a managed, cloud-based network security service that protects your Azure Virtual Network resources.
- It's a **fully stateful** firewall as a service with built-in **high availability and unrestricted cloud scalability**.

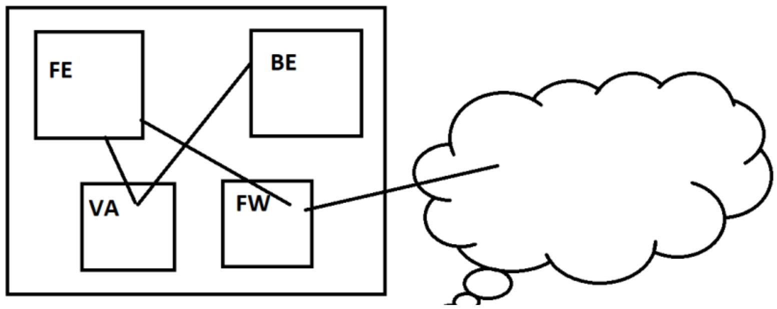
- Azure Firewall **can scale up** as much as you need to accommodate changing network traffic flows, so you don't need to budget for your peak traffic.
- Application FQDN filtering rules. You can limit outbound HTTP and HTTPS traffic to a specified list of FQDNs, including wildcards. This feature doesn’t require SSL termination.
- Threat intelligence-based filtering can alert and deny traffic from/to known malicious IP addresses and domains which are updated in real time to protect against new and emerging attacks.
- Multiple public IP addresses.
- Integrated with Azure Monitor. All events are integrated with Azure Monitor, allowing you to archive logs to a storage account, stream events to your event hub, or send events to Azure Monitor logs.


User configuration
L3-L7 connectivity policies
Microsoft Threat Intelligence
Known malicious IPs and FQDNs
Spoke 1
Spoke 2
Spoke VNets
VNET/VWAN
Azure Firewall
Threat intel, Web Categories NAT, network and application traffic filtering rules allows inbound/ outbound access
Traffic is denied by default
Azure to on-prem traffic filtering
On-premises

Note: Azure Firewall must be in the same resource group as Azure VNet.

Pricing: https://azure.microsoft.com/en-in/pricing/details/azure-firewall/

Requirement:



Configure an application rule in Firewall Policy:

This is the application rule that allows outbound access to www.google.com.

1. In Vnet Create a Subnet by name=**"AzureFirewallSubnet" (DON'T change the name)**
2. **Create a Firewall** resource (Demo-firewall) in existing VNet (Demo-VNet) and with New IP Address (Demo-firewall-ip)

Note: Resource Group of Firewall should be same as that of Virtual Netowrk

3. Create a Route with prefix 0.0.0.0/0 and for Next hop select Virtual Appliance and provide the Private IP Address of Firewall (192.168.5.4).
4. Goto to any and browse www.google.com.

Note that the request is **blocked** by the firewall

5. **Configure an Application Rule**: Source Address = <IP Range of Frontend-subnet> (192.168.1.0/24), Protocol=http, https; Target FQDNS=www.google.com

Note that the request is **not blocked** by the firewall

Configure a network rule

This is the network rule that allows outbound access to two IP addresses at port 53 (DNS).

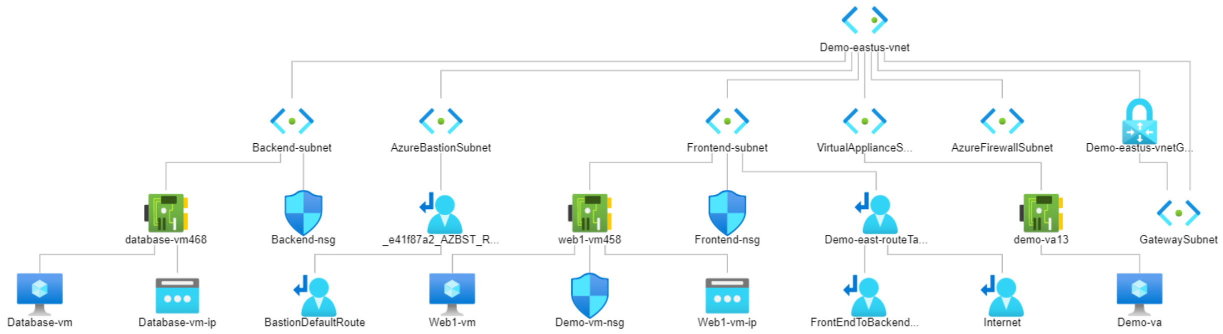
1. **Configure Network Rule**: Allow-DNS, Protocol=**UDP**, Source Address=<IP Range of Frontendsubnet>, Destination Address="209.244.0.3, 209.244.0.4", Destination Port=**53**
2. Change the DNS Settings of Virtual Network : Demo-vnet → DNS Servers → Add 209.244.0.3 & 209.244.0.4 as Servers
3. Restart the VM's

Configure a DNAT rule

This rule allows you to **connect a remote desktop** to the virtual machine through the firewall.

1. Select the NAT rule collection tab → Add NAT rule collection.
2. Name=rdp, Priority=200,
3. Rules Name=rdp-nat, Protocol=TCP, Source type=IP address, Source=*, Destination address=<**firewall public IP address**>, Destination Ports=3389, Translated address=<**private IP address of VM**>, Translated port=3389.
4. Select Add.

Assignment



1. Demo-vnet (192.168.0.0/16)

2. Frontend-subnet (192.168.1.0/24)

3. Backend-subnet (192.168.2.0/24)

4. Web1-vm (192.168.1.4)

5. DbSrv-vm (192.168.2.4)

6. Frontend-nsg Associated to Frontend-subnet

    a) Allow Inbound RDP(3389) and HTTP(80)

7. Backend-nsg Associated to Backend-subnet

    b) Allow Inbound RDP(3389) & SQL(1433)

    c) Deny Outbound Internet

8. AzureBastionSubnet (Subnet)

    Bastion Service to connect to all VM

9. VirtualAppliance-subnet (192.168.4.0/27)

10. Demo-va (192.168.4.4)

11. Frontend-udr and attach to Frontend-subnet

    Dest is 192.168.2.0/24 => Next Hop = VA

12. AzureFirewallSubnet (192.168.5.0/24)

13. Azure Firewall Service (192.168.5.4)

    Application Rule

      Allow *.google.com / *.deccansoft.com

    Network Rule

      Allow 8.8.8.8:53

14. New Rule Frontend-udr

Dest is 0.0.0.0/0 => Next Hop = 192.168.5.4 (Firewall)

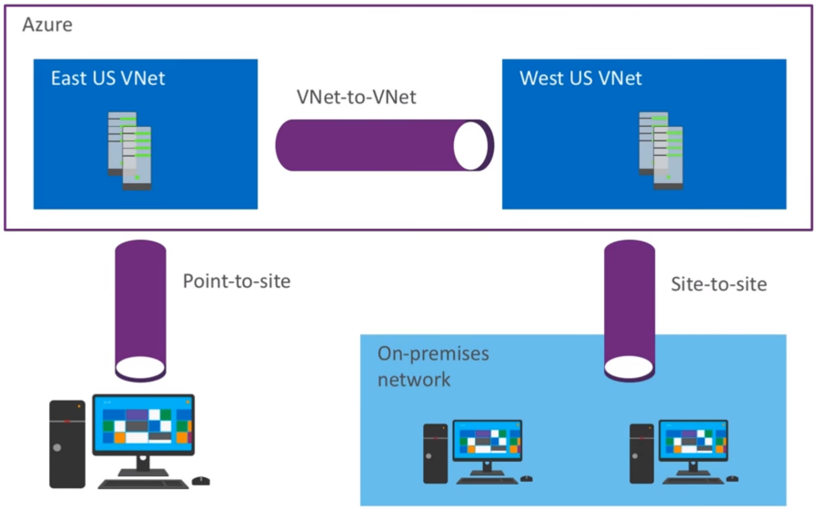
15. Jump-subnet (192.168.0.0/24)

Jump-vm (192.168.0.4)

With NSG and allowed RDP.

Create connectivity between virtual networks

There are multiple ways to connect VNets. The sections below describe different ways to connect virtual networks.



Cloud-Only Virtual Networks:

You can choose not to make any kind of virtual private network (VPN) connection to a VNet. Instead, when you create a VM or cloud service, you can specify endpoints that external clients can connect to. An endpoint is a VIP and a port number. Therefore an endpoint can be used only for a specific protocol, such as connecting a Remote Desktop Protocol (RDP) client or browsing a website. These VNets are known as cloud-only virtual networks. A dynamic routing gateway is not required in the VNet. Endpoints are published to the Internet, so they can be used by anyone with an Internet connection, including your on-premises computers.

Point-to-Site VPNs

A simple way to connect a VPN to an Azure VNet is to use a Point-to-Site VPN. In these VPNs, you configure the connection on individual on-premises computers. No extra hardware is required but you must complete the configuration procedure on every computer that you want to connect to the VNet. Point-to-site VPNs can be used by the client computer to connect to a VNet from any location with an Internet connection. Once the VPN is connected, the client computer can access all VMs and cloud services in the VNet as if they were running on the local network.

Site-to-Site VPNs

To connect **all the computers** in a physical site to an Azure VNet, you can create a Site-to-Site VPN. In this configuration, you do not need to configure individual computers to connect to the VNet, **instead you configure a VPN device**, which acts as a gateway to the VNet.

When you use the Site-to-Site IPsec steps, you create and configure the local network gateways manually. The local network gateway for each VNet treats the other VNet as a local site. This lets you specify additional address space for the local network gateway in order to route traffic. If the address space for a VNet changes, you need to update the corresponding local network gateway to reflect that. It does not automatically update.

VNet-to-VNet

Connecting a virtual network to another virtual network (VNet-to-VNet) is similar to connecting a virtual network to an on-premises site location. Both connectivity types use a VPN gateway to provide a secure tunnel using IPsec/IKE. The VNets you connect can be in **different subscriptions** and **different regions**.

The difference between the S2S AND V2V connection types is the way the local network gateway is configured. When you create a VNet-to-VNet connection, you do not see the local network gateway address space. It is automatically created and populated. If you update the address space for one VNet, the other VNet automatically knows to route to the updated address space. Creating a VNet-to-VNet connection is typically faster and easier than creating a Site-to-Site connection between VNets.

You can combine VNet to VNet communication with multi-site configurations. This lets you establish network topologies that combine cross-premises connectivity with inter-virtual network connectivity.

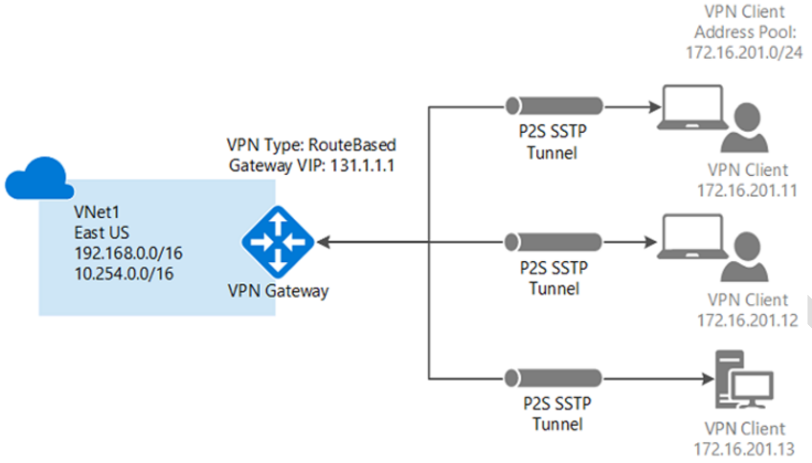
VNet peering

You may want to consider connecting your VNets using VNet Peering. VNet peering does not use a VPN gateway and has different constraints. Additionally, VNet peering pricing is calculated differently than VNet-to-VNet VPN Gateway pricing

ExpressRoute

ExpressRoute is a service that enables Azure customers to create a dedicated connection to Azure, which does not connect through the public Internet. This contrasts with VPNs, which use encryption to tunnel securely through the public Internet. Because ExpressRoute connections are dedicated, they can offer faster speeds, higher security, lower latencies, and higher reliability than VPNs.

Create a Point-to-Site VPN



Generate Certificates – Self signed root certificate for P2S connection

1. Open the Powershell command window and execute the following (Do not close the window)
2. **Generate server and client certificate: Execute the following command in the same PowerShell window opened earlier.**

```
$cert = New-SelfSignedCertificate -Type Custom -KeySpec Signature `
 -Subject "CN=P2SRootCert" -KeyExportPolicy Exportable `
 -HashAlgorithm sha256 -KeyLength 2048 `
 -CertStoreLocation "Cert:\CurrentUser\My" -KeyUsageProperty Sign -KeyUsage CertSign


New-SelfSignedCertificate -Type Custom -KeySpec Signature `
-Subject "CN=P2SChildCert" -KeyExportPolicy Exportable `
-HashAlgorithm sha256 -KeyLength 2048 `
-CertStoreLocation "Cert:\CurrentUser\My" `
-Signer $cert -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.2")
```

3. **To obtain the public key (.cer file) of Root Certificate**
    1. Search → **Manage User Certificates** → Personal → Certificates

       or
    2. **MMC → File → Add/Remove Snap-In → Certificates → Add → OK**

       open **certmgr.msc**., typically in **'Certificates - Current User\Personal\Certificates'**.
    3. Locate the self-signed root certificate **(P2SRootCert)** → Right Click → **All Tasks**, and then click **Export**. This opens the **Certificate Export Wizard**.
    4. In the Wizard, click **Next**. Select **No, do not export the private key**, and then click **Next**.

5. On the **Export File Format** page, select **Base-64 encoded X.509 (.CER).**, and then click **Next**.
6. **File to Export**, **Browse** = d:\P2SRootCert.cer → **Next**.
7. Click **Finish** to export the certificate. You will see **The export was successful**. Click **OK** to close the wizard.

A client certificate that is present on the device is used to authenticate the connecting user. Client certificates are generated from a trusted root certificate and then installed on each client computer. You can use a root certificate that was generated using an Enterprise solution, or you can generate a self-signed certificate.

The validation of the client certificate is performed by the VPN gateway and happens during establishment of the P2S VPN connection. The root certificate is required for the validation and must be uploaded to Azure

More about Certficates:

https://docs.microsoft.com/en-us/azure/vpn-gateway/vpn-gateway-certificates-point-to-site

Optional: Ensure that a Gateway Subnet is already created in a VNet

4. Select VNet → Subnets → **+ Gateway subnet** → OK

Create a Virtual Network Gateway

5. All Services → **Virtual Network Gateway** → +Add
   a) Name=TestVNetGateway, . . . ,
   b) Gateway type = VPN
   c) VPN type = Route-based
   d) SKU = Basic (table below)
   e) Choose a virtual network,
   f) Create a New IP
   g) Create

Note: Provisioning a virtual network gateway may take up to 20 minutes.

About VPN Types:

1. **RouteBased**: RouteBased VPNs use "**routes**" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The tunnel interfaces then encrypt or decrypt the packets in and out of the tunnels.

2. **PolicyBased**: Policy-based VPNs encrypt and direct packets through IPsec tunnels based on the **IPsec policies** configured with the combinations of address prefixes between your on-premises network and the Azure VNet.
    1. ~~PolicyBased VPNs can **only** be used on the **Basic** gateway SKU.~~
    2. You can have only **1 tunnel** when using a PolicyBased VPN.
    3. You can only use PolicyBased VPNs for **S2S connections**.

Which Gateway SKUs Support P2S VPN?

SKU	P2S Connections	S2S/VNet-to-VNet Tunnels	Aggregate Throughput Benchmark
Basic	128	Max. 30	100 Mbps
VpnGw1	128	Max. 30	650 Mbps
VpnGw2	128	Max. 30	1Gbps
VpnGw3	128	Max. 10	1.25 Gbps

About BGP with Azure VPN Gateway

BGP (Border Gateway Protocol) 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 your on-premises VPN devices, called BGP peers or neighbors, to exchange "routes" that will inform both gateways on the availability and reachability for those prefixes to go through the gateways or routers involved.

Upload the root certificate .cer file

6. Open the Root certificate (not child) with a text editor, such as Notepad. Copy the content between and **excluding** -----BEGIN CERTIFICATE------ and -----END CERTIFICATE------
7. Select VNetGateway created earlier → **Point-to-site configuration**,
    1. **Address Pool**=172.16.201.0/24 (is the pool of IP addresses from which clients that connect will receive an IP address.)
    2. Tunnel type = **IKEv2 and SSTP (SSL)**
    3. **Root Certificates**: Name=RootCert1, Public Certficate Data <Value copied in step 6>

Note: You can add up to **20 trusted root certificates**.

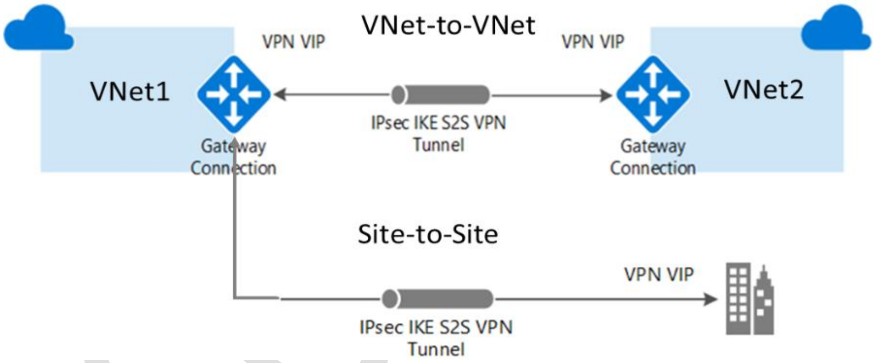
Download and Install VPN client

8. **Point-to-site configuration** → Download VPN client
9. Select X64 → Download
10. Execute the downloaded EXE file
11. Client Computer → Network Settings → VPN
12. Click on TestVNet and connect to VNet.
13. To verify that your VPN connection is active, open an elevated command prompt, and **run *ipconfig/all***.
14. You can also use the Private IP of any VM in the VNet and open it in Web Browser to the response of the page.
15. **You can RDP to one of the VM and browse websites of other VM in the other Vnet using Private IP.**

Create and configure VNET to VNET using VPN Gateway

You can connect your VNets with a VNet-to-VNet VPN connection.

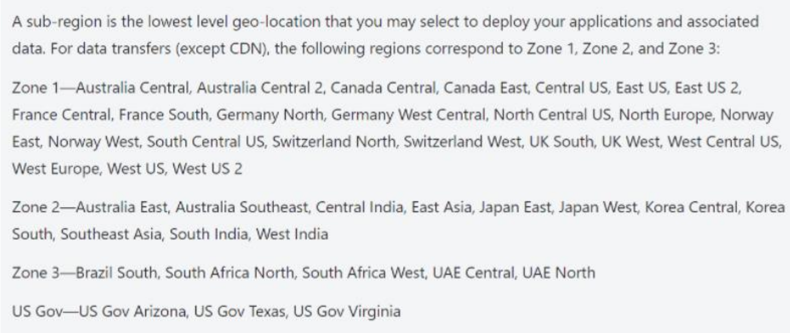
Uses an Azure VPN gateway to provide a secure tunnel using IPSec/IKE. Though the traffic is secured in VPN, it leaves Azure and travels over public internet for transport.



With a VNet-to-VNet connection your VNets can be:

- in the same or different regions.
- in the same or different subscriptions.
- in the same or different deployment models.

Note that VNet-to-VNet traffic within the **same region is free** for both directions when using a VPN gateway connection. Cross region VNet-to-VNet egress traffic is charged with the outbound inter-VNet data transfer rates based on the source regions. Visit https://azure.microsoft.com/en-us/pricing/details/vpn-gateway/ for Pricing details.



A sub-region is the lowest level geo-location that you may select to deploy your applications and associated data. For data transfers (except CDN), the following regions correspond to Zone 1, Zone 2, and Zone 3:

Zone 1—Australia Central, Australia Central 2, Canada Central, Canada East, Central US, East US, East US 2, France Central, France South, Germany North, Germany West Central, North Central US, North Europe, Norway East, Norway West, South Central US, Switzerland North, Switzerland West, UK South, UK West, West Central US, West Europe, West US, West US 2

Zone 2—Australia East, Australia Southeast, Central India, East Asia, Japan East, Japan West, Korea Central, Korea South, Southeast Asia, South India, West India

Zone 3—Brazil South, South Africa North, South Africa West, UAE Central, UAE North

US Gov—US Gov Arizona, US Gov Texas, US Gov Virginia

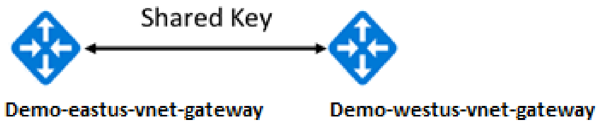
In the example, the virtual networks are in the same subscription, but in different resource groups. If your VNets are in different subscriptions, you can't create the connection in the portal. You can use PowerShell or CLI.

1. Create and configure the first VNet: **Demo-eastus-vnet**
2. Create a Gateway Subnet
3. Create a virtual network gateway – **Demo-eastus-vnet-gateway**

4. Create and configure the second VNet **Demo-westus-vnet**
5. Create a Gateway Subnet
6. Create a virtual network gateway – **Demo-westus-vnet-gateway**

Configuring and Connect VPN Gateways

Once your VPN gateways are created, you can create the connection between them. If your VNets are in the same subscription, you can use the portal.



7. Configure the **Demo-eastus-vnet-gateway** connection
   - Select **Demo-eastus-vnet-gateway** → Connections → +Add
   - Name = EastToWestConnection
   - Connection type = VNet-to-VNet
   - Second virtual network gateway = **Demo-westus-vnet-gateway**
   - **Shared key = "abc123"**
8. Configure the **Demo-westus-vnet-gateway** connection

   Follow the steps from the previous section, replacing the values to create a connection from **Demo-eastus-vnet** to **Demo-westus-vnet**. Make sure that you use the **same shared key**.

9. Verify your connections
   - Select Virtual Network Gateway → Connections
   - Ensure that Status value change to **Succeeded and Connected**.
10. **You can also RDP (using Public IP) to one of the VM and RDP or browse websites (using Private IP) of other VM in the other Vnet using Private IP.**

Create and Configure VNet Peering

Virtual network peering enables you to seamlessly connect two Azure virtual networks. Once peered, the virtual networks appear as one, for connectivity purposes.

The traffic between virtual machines in the peered virtual networks is routed through the **Microsoft backbone infrastructure**, much like traffic is routed between virtual machines in the same virtual network, through *private* IP addresses only.

Azure supports:

- **VNet peering** - connecting VNets within the **same Azure region.**
- **Global VNet peering** - connecting VNets across different Azure regions.

Benefits

1. Its best alternative to VPN for vNets because all network traffic between peered virtual networks is **private and routed over Azure internal networks** instead of public internet.

2. A low-latency, high-bandwidth connection between resources in different virtual networks.

3. The ability to transfer data across Azure subscriptions, deployment models, and across Azure regions.

Pros and Cons over VPN Gateway

Pros

1. Faster and easier to setup than VPN
2. No Public IP required.

Cons

11. Peering relationships are not transitive.

    If you create peerings between:

    - VirtualNetwork1 & VirtualNetwork2
    - VirtualNetwork2 & VirtualNetwork3

    There is no peering between VirtualNetwork1 and VirtualNetwork3 through VirtualNetwork2.

12. You can't add address ranges to, or delete address ranges from a virtual network's address space once a virtual network is peered with another virtual network (Even if flow is disabled)
13. Cannot use overlapping address spaces.

Pricing:

https://azure.microsoft.com/en-us/pricing/details/virtual-network/

Configuring a Peering

1. Select the Vnet → Settings → **Peerings**

2. Select + Add

3. Enter Name . . . and other details

- **I know my resource ID**

  If you have read access to the virtual network you want to peer with, leave this checkbox unchecked.

  If you don't have read access to the virtual network or subscription you want to peer with, check this box.

  *ARM Template Function: resourceId(<subscriptionID>,<ResourceGroupName>,***<Resource-Type>***,* **<ResourceName1>***,<ResourceName2>)*

  "id": "/subscriptions/3c062fc8-2da3-4704-9dbb-8f91ffb43902/resourceGroups/Demo-rg/providers/Microsoft.Network/virtualNetworks/VnetName",

- **Allow virtual network access:**

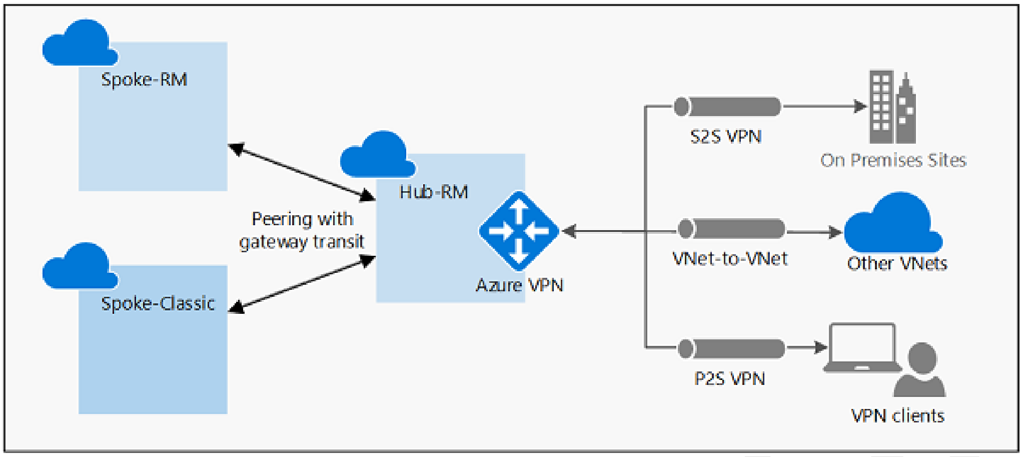
  Select **Enabled** (default) if you want to enable communication between the two virtual networks. You might select **Disabled** if you've peered a virtual network with another virtual network, but occasionally want to disable traffic flow between the two virtual networks.

- **Allow forwarded traffic:** Check this box to allow traffic *forwarded* by a network virtual appliance in a virtual network (that didn't originate from the virtual network) to flow to this virtual network through a peering. You don't need to check this setting if traffic is forwarded between virtual networks through an Azure VPN Gateway.

- **Allow gateway transit:** Check this box if you have a virtual network gateway attached to this virtual network and want to allow traffic from the peered virtual network to flow through the gateway.

- **Use remote gateways:** Check this box to allow traffic from this virtual network to flow through a virtual network gateway attached to the virtual network you're peering with.

Configure VPN gateway transit for virtual network peering



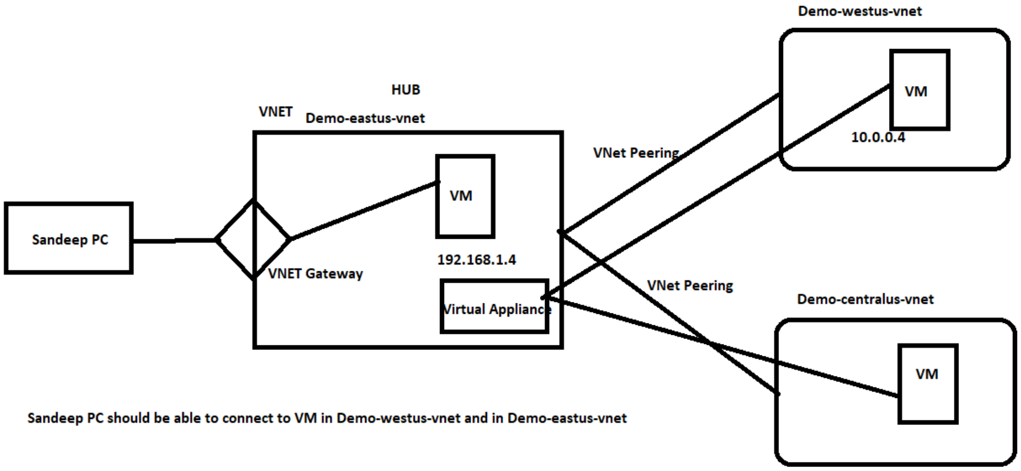
In the diagram, gateway transit allows the peered virtual networks to use the Azure VPN gateway in Hub-RM. Connectivity available on the VPN gateway, including S2S, P2S, and VNet-to-VNet connections, applies to all three virtual networks.

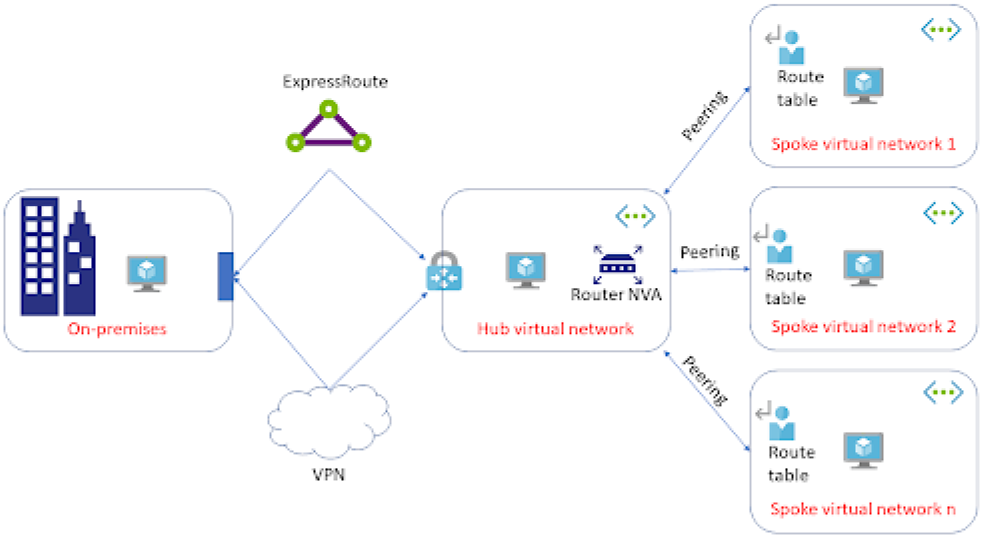
In HubToSpoke VNet Peering – Select **Use this virtual network's gateway**

In SpokeToHub VNet Peering - Select **Use the remote virtual network's gateway**

Now, we can connect from our Local Machine (Provided P2S Connection is established to Hub) to a VM in Hub as well as Spoke VNet.

Note: Delete the existing VPN Client. Download new VPN Client and setup again.





Communication between spokes: https://blog.ine.com/azure-practical-peer-to-peer-transitive-routing

Assignment:

Create Visio Diagram

=================

Demo-eastus-vnet = 10.1.0.0/16

Frontend-subnet = 10.1.1.0/24

Frontend-nsg

Inbound Rules

AllowRDP = 3389

AllowHTTP = 80

Web1-vm (IP=10.1.1.4)

Subnet=Frontend-subnet

No NSG (Subnet NSG will be used)

Public IP = Web1-vm-ip (Basic/Static)

Web2-vm (IP=10.1.1.5)

Subnet=Frontend-subnet

No NSG (Subnet NSG will be used)

Public IP = Web2-vm-ip (Basic/Static)

Backend-subnet = 10.1.2.0/24

Backend-nsg

Inbound Rules

AllowRDP = 3389

AllowSQL = 1433

Outbound Rule

DenyInternet = ServiceTag=Internet

Database-vm (IP=10.1.2.4)

VirtualAppliance-subnet = 10.1.4.0/24

VM: EastUS-va (10.1.4.4) - No Public IP

No NSG

Networking -> NIC -> IP Configuration -> IP Forwarding = Enabled.

RDP: PS Command: Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRouter -Value 1

AzureBastionSubnet* - 10.1.3.0/24

Bastion Service with Public IP

Demo-westus-vnet = 10.2.0.0/16

Default = 10.2.1.0/24

WestUS-VM (10.2.1.4)

No NSG

Demo-hub-vnet = 10.0.0.0/16 (Central US)

Default = 10.0.0.0/24

VM: Hub-va (10.0.0.4) - No Public IP

No NSG

Networking -> NIC -> IP Configuration -> IP Forwarding = Enabled.

RDP: PS Command: Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRouter -Value 1

GatewaySubnet* = 10.0.1.0/24

Create Virtual Network Gateway

Point to Site Configuration: Address pool: 172.16.0.0/24, Tunnel Type=IKEv2 and SSTP, Authentication Type=Azure certificate, Root Certificate=Content of Certificate file

AzureFirewallSubnet* = 10.0.2.0/24

Download VPN Client

RouteTable: EastUS-routetable

Subnets = Frontend-subnet, Backend-subnet

Routes = FromFrontEndSubnet-ToBackendSubnet, Address Prefix=10.1.2.0/24, Next hop type=Virtual Appliance, IP=10.1.4.4 (IP of EastUS-va)

Routes = FromEast-ToWest, Address Prefix=10.2.0.0/16, Next hop type=Virtual Appliance, IP=10.0.0.4(IP of Hub-va)

RouteTable: WestUS-routetable

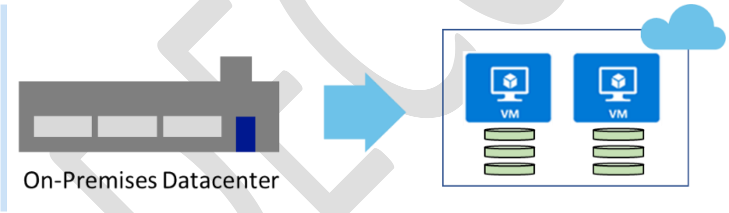
Subnets = Default

Routes = FromWest-ToEast, Address Prefix=10.1.0.0/16, Next hop type=Virtual Appliance, IP=10.0.0.4(IP of Hub-va)

Site-to-Site VPN

Site-to-Site Scenarios

There are many scenarios where Site-to-Site connections can be useful. Here are a few.

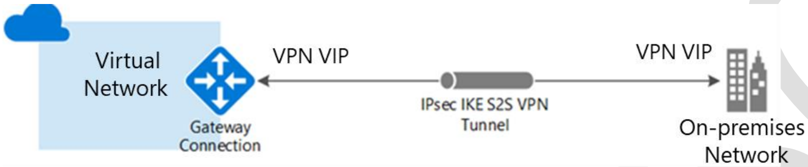


- **Capacity On-Demand:** Azure provides capacity on demand. By creating a connection to Azure, more storage or compute resources can easily be brought online. You can extend your on-premises datacenter without purchasing and installing equipment in the datacenter. This scenario includes spawning remote offices.
- **Strategic Migration:** There are many strategic reasons for moving to Azure. Organizations whose core purpose is not related to managing complex datacenter deployments, may want to shed competing interests and focus on improving their core business. They may also want to reduce costs by moving to a pay as you go model.

Migrating services is usually faster than responding in-house, especially when you trying to project a global presence.

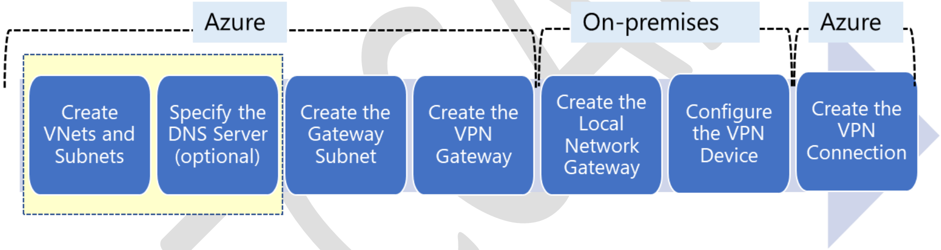
- **Disaster Recovery:** The cloud offers an efficient, cost effective choice for data backup and recovery. Most cloud platforms let you run third-party software for backup and disaster recovery, but with Microsoft these services are fully integrated and easy to turn on, which means you don't have to install and manage a separate product in the cloud.

A Site-to-Site (S2S) connection is a connection over IPsec/IKE (IKEv1 or IKEv2)VPN tunnel. S2S connections can be used for cross-premises and hybrid configurations. This type of connection requires a VPN device located on-premises that has a public IP address assigned to it.

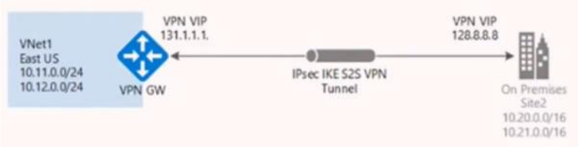


Implementing Site-to-Site VPN

To configure Site-to-Site you must configure both sides of the infrastructure. For example, Azure and on-premises. There are a lot of steps, but we will go through each one. As we do, try to keep the architecture diagrams in mind.


Azure
On-premises
Azure
Create VNets and Subnets
Specify the DNS Server (optional)
Create the Gateway Subnet
Create the VPN Gateway
Create the Local Network Gateway
Configure the VPN Device
Create the VPN Connection

✔ **Take time to carefully plan your network configuration. If a duplicate IP address range exists on both sides of the VPN connection, traffic will not route the way you may expect it to.**



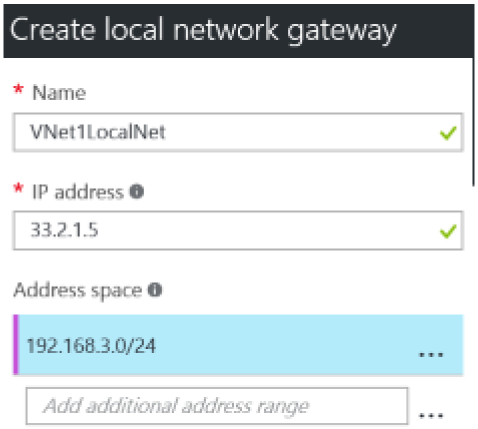
Step 1: Create a VNet, Gateway Subnet, VPN Gateway as described in example of **Point to Site.**

Step 2: Setup Local Network Gateway:

The local network gateway typically refers to the on-premises location.

4. You give the site a **name** by which Azure can refer to it,

5. Specify the **IP address (eg: 33.2.1.5)** of the on-premises VPN device for the connection.
6. Specify **Address Space**. One or more IP address ranges (in CIDR notation) that define your local network's address space. For example: 192.168.3.0/24 and 10.21.0.0/16.



Create local network gateway

* Name: VNet1LocalNet

* IP address: 33.2.1.5

Address space: 192.168.3.0/24

Step 3: Configure the VPN Device

Microsoft has validated a list of standard VPN devices that should work well with the VPN gateway. This list was created in partnership with device manufacturers like Cisco, Juniper, Ubiquiti, and Barracuda Networks. If you don't see your device listed in the validated VPN devices table (reference link), your device may still work with a Site-to-Site connection. Contact your device manufacturer...

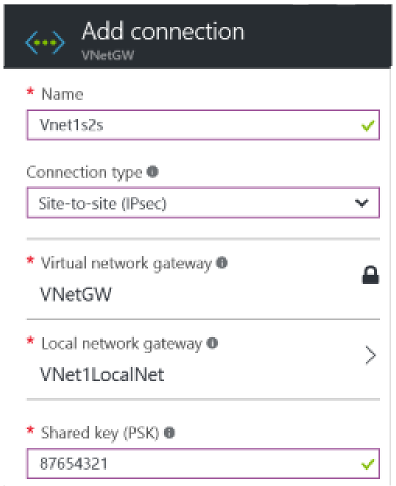
To configure your VPN device, you need the following:

- **A shared key**. This is the same shared key that you will specify when creating the VPN connection (next step).
- The **public IP address** of your VPN gateway.

✔ Depending on the VPN device that you have, you may be able to download a VPN device configuration script.

Step 4: Configure the VPN Connection

In this step you will configure the connection between the Azure VPN gatewayand the local network gateway.



Add connection
VNetGW

* Name
Vnet1s2s

Connection type
Site-to-site (IPsec)

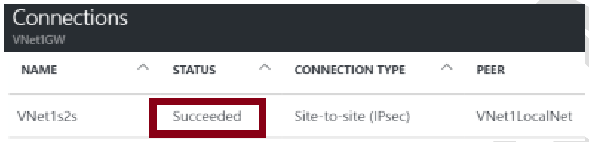
* Virtual network gateway
VNetGW

* Local network gateway
VNet1LocalNet

* Shared key (PSK)
87654321

For **Shared Key**, the value here must match the value that you are using for your local VPN device. If your VPN device on your local network doesn't provide a shared key, you can make one up and input it here and on your local device. The important thing is that the shared keys match.

When the connection is complete, you'll see it appear in the Connections bladefor your Gateway.



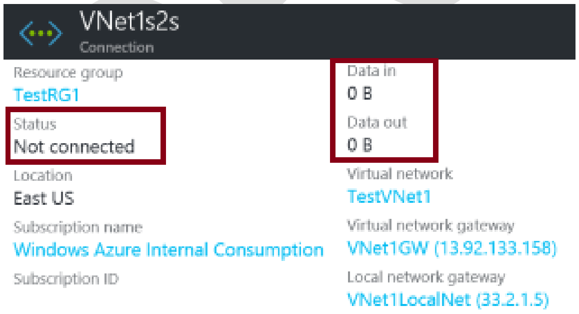
Connections
VNet1GW

NAME	STATUS	CONNECTION TYPE	PEER
VNet1s2s	Succeeded	Site-to-site (IPsec)	VNet1LocalNet

Step 4: Verify the VPN Connection

After you have configured all the Site-to-Site components it is time to verifythat everything is working. You can verify the connections either in the portal,or by using PowerShell.

Portal: When you view your connection in the Azure portal the Status should beSucceeded or Connected. Also, you should have data flowing in the Data in andData out information.



VNet1s2s
Connection

Resource group
TestRG1

Status
Not connected

Location
East US

Subscription name
Windows Azure Internal Consumption

Subscription ID

Data in
0 B

Data out
0 B

Virtual network
TestVNet1

Virtual network gateway
VNet1GW (13.92.133.158)

Local network gateway
VNet1LocalNet (33.2.1.5)

PowerShell: To verify your connection with PowerShell

```
Get-AzureRmVirtualNetworkGatewayConnection -NameMyGWConnection -ResourceGroupName MyRG
```

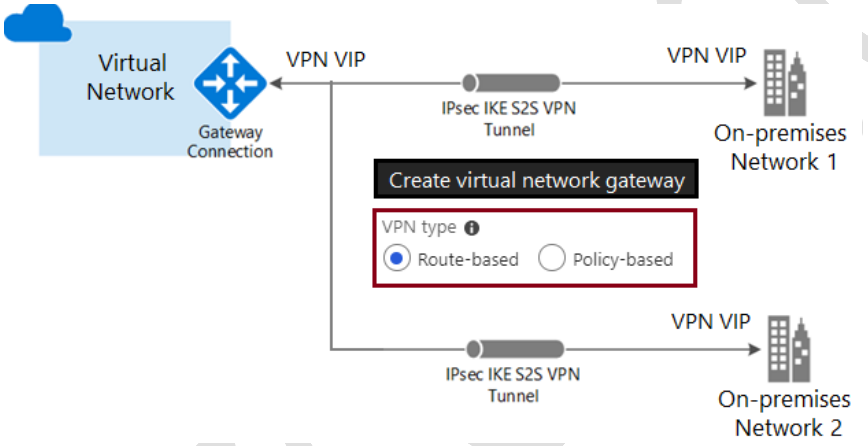
After the cmdlet has finished, view the values. The connection status shouldshow ‘Connected’ and you can see ingress and egress bytes.

```
"connectionStatus": "Connected",
"ingressBytesTransferred": 33509044,
"egressBytesTransferred": 4142431
```

YouTube Video: **https://youtu.be/5VTdah3VwYU**

Multiple Sites

A Multi-site connection is a variation of the Site-to-Site connection. You create more than one VPN connection from your virtual network gateway, typically connecting to multiple on-premises sites. When working with multiple connections, you must use a Route-based VPN. Because each virtual network can **only have one VPN gateway**, all connections through the gateway share the available bandwidth.

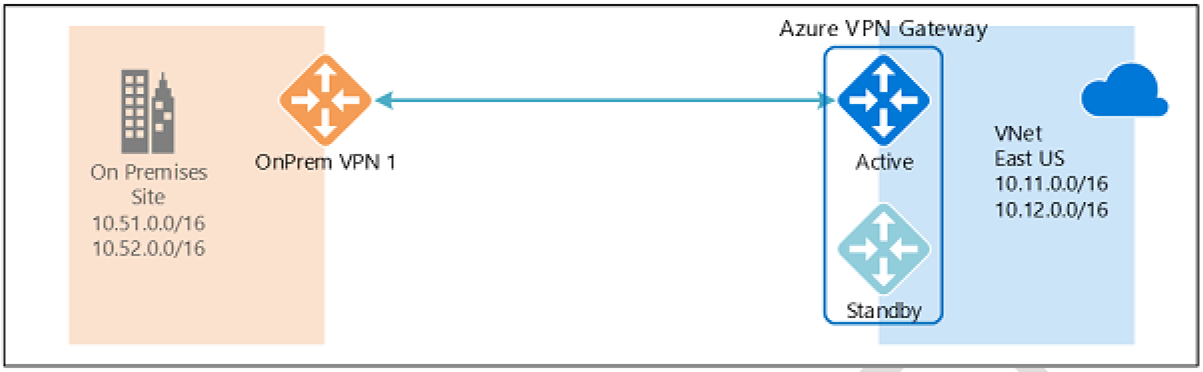


VPN type

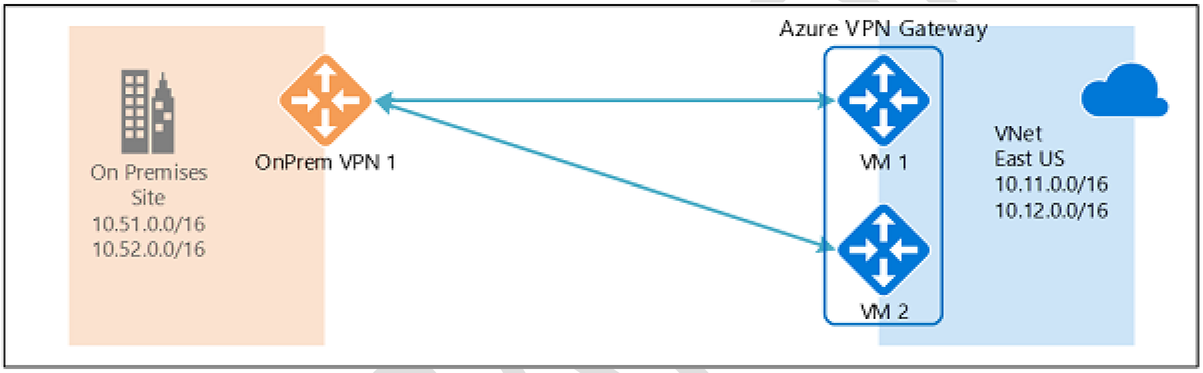
● Route-based ○ Policy-based

About active-active mode for Availability:

Every Azure VPN gateway consists of **two instances** in an **active-standby** configuration. For any planned maintenance or unplanned disruption that happens to the active instance, the standby instance would take over (failover) **automatically**, and resume the S2S VPN or VNet-to-VNet connections. **The switch over will cause a brief interruption**. For planned maintenance, the connectivity should be restored **within 10 to 15 seconds**. For unplanned issues, the connection recovery will be longer, about 1 minute to 1 and a half minutes in the worst case. For P2S VPN client connections to the gateway, the P2S connections will be disconnected and the users will need to reconnect from the client machines.



You can now create an Azure VPN gateway in an **active-active configuration**, where both instances of the gateway VMs will establish S2S VPN tunnels to your on-premises VPN device, as shown the following diagram:

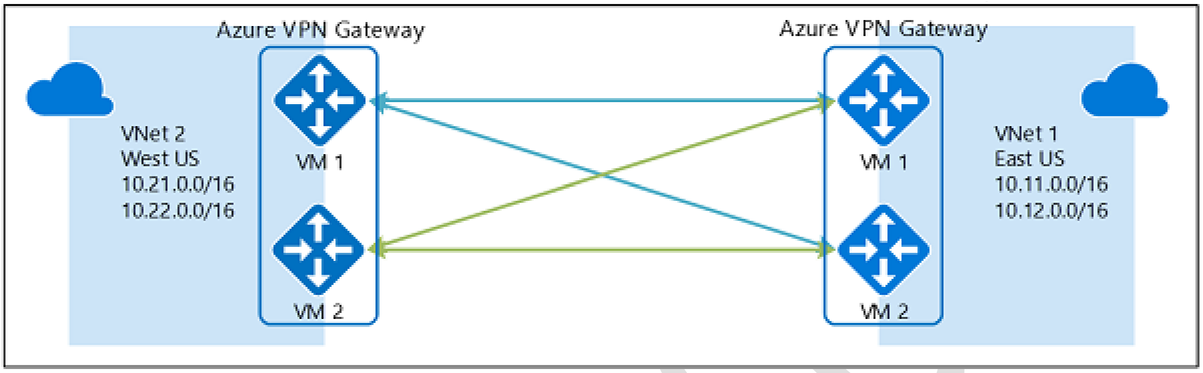


In this configuration, each Azure gateway instance will have a unique public IP address, and each will establish an IPsec/IKE S2S VPN tunnel to your on-premises VPN device specified in your local network gateway and connection. Note that both VPN tunnels are actually part of the same connection. You will still need to configure your on-premises VPN device to accept or establish two S2S VPN tunnels to those two Azure VPN gateway public IP addresses.

Because the Azure gateway instances are in active-active configuration, the traffic from your Azure virtual network to your on-premises network will be routed **through both tunnels simultaneously**, even if your on-premises VPN device may favor one tunnel over the other. Note though the same TCP or UDP flow will always traverse the same tunnel or path, unless a maintenance event happens on one of the instances.

When a planned maintenance or unplanned event happens to one gateway instance, the IPsec tunnel from that instance to your on-premises VPN device will be disconnected. The corresponding routes on your VPN devices should be removed or withdrawn automatically so that the traffic will be switched over to the other active IPsec tunnel. On the Azure side, the switch over will happen automatically from the affected instance to the active instance.

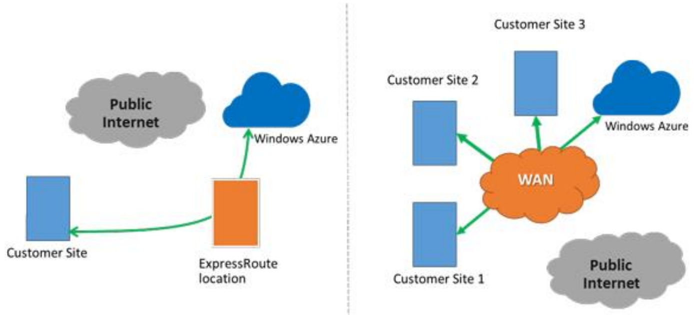
You can create active-active VPN gateways for both virtual networks, and connect them together to form the same full mesh connectivity of 4 tunnels between the two VNets, as shown in the diagram below:



This ensures there are always a pair of tunnels between the two virtual networks for any planned maintenance events, providing even better availability.

Azure Express Route

- Microsoft Azure ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a dedicated private connection facilitated by a connectivity provider.
- ExpressRoute offers **private, reliable and low-latency** connections between customer's datacenters and Azure.
- Microsoft is positioning the technology as a way for customers to extend their network in Windows Azure as part of a **hybrid IT approach**.

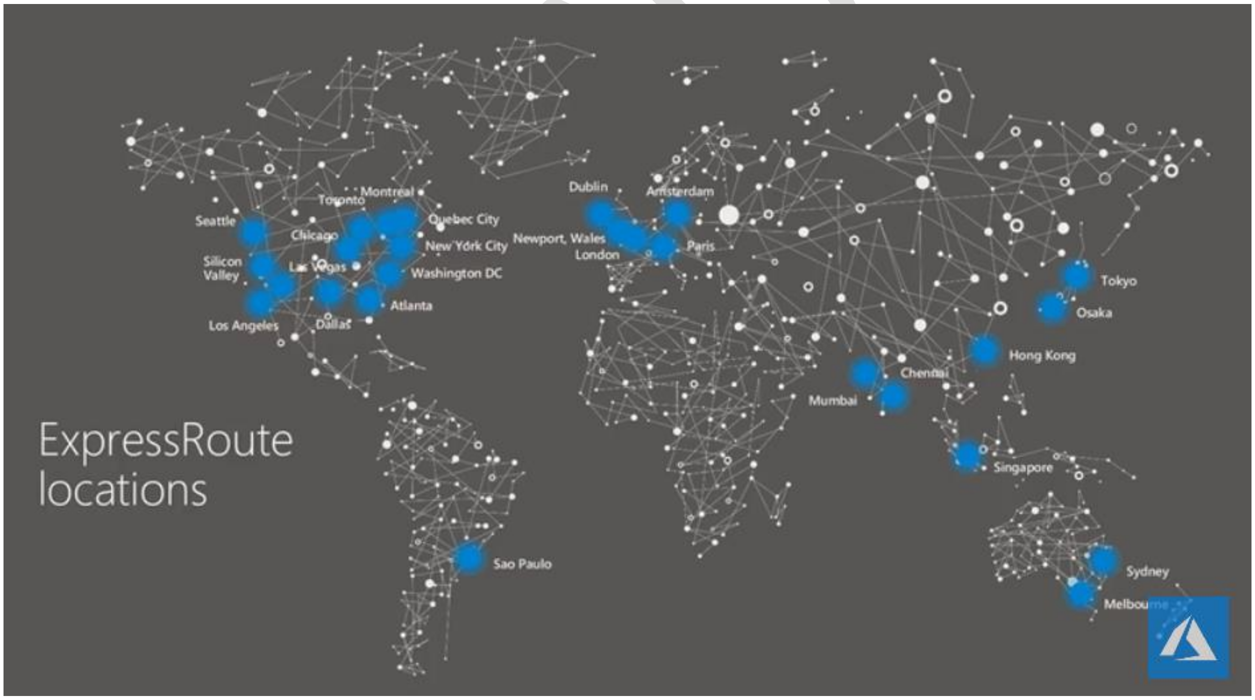


Connecting at ExpressRoute location (Exchange Provider facility)

Connecting from a WAN (e.g. MPLS VPN) provided by network service providers

Express Route Partners to deliver Express Route connectivity

alaska
altice
aarnet
AIRGATE
airtel
ARTERIA
Aryaka
Ascenty
AT&T
Bell
BT
CENTRO
cogeco
cologix
COMCAST BUSINESS
Console
CORESITE
COX
e-shelter
epsilon
euroRiber
GÉANT
GLOBAL CLOUD XCHANGE
gtt
IIJ
InterCloud
interxion
Jisc
KINX
kpn
Level(3)
LGA
MACROVIEW
MASERGY
Megaport
MTN
Neutrona
NexGen NETWORKS
NGD
NEXTDC
orange
PCCW
QSC
ROGERS
SEJONG
sify
SINET5
Singtel
SoftBank
T5
Tamares
TATA COMMUNICATIONS
TelecityGroup
Telefonica
KDDI TELEHOUSE
Telstra
TELUS
T-Systems
UIH
UOL DIVEO
verizon
vodafone
webair
windstream
zayo
zertia
ZIRRO



https://youtu.be/HZH6F_gLCQQ

https://youtu.be/wystDyqyRXc

Creating Express Circuits: https://youtu.be/_8S3tOwWgc8

Using PowerShell Commands

1. Create a new resource group

```
New-AzResourceGroup -Name TestRG -Location centralus
```

2. Create a new VNet named *TestVNet*

```
New-AzVirtualNetwork -ResourceGroupName TestRG -Name TestVNet -AddressPrefix 192.168.0.0/16 -Location centralus
```

3. Store the virtual network object in a variable

```
$vnet = Get-AzVirtualNetwork -ResourceGroupName TestRG -Name TestVNet
```

4. Add a subnet to the new VNet variable

```
Add-AzVirtualNetworkSubnetConfig -Name FrontEndSubnet -VirtualNetwork $vnet -AddressPrefix 192.168.1.0/24
```

5. Repeat above step for each subnet you want to create

```
Add-AzVirtualNetworkSubnetConfig -Name BackEndSubnet -VirtualNetwork $vnet -AddressPrefix 192.168.2.0/24
```

6. Although you create subnets, they currently only exist in the local variable used to retrieve the VNet you create in step 4 above.

```
Set-AzVirtualNetwork -VirtualNetwork $vnet
```

7. **Create a Public IP address** (PIP) resource named PublicIP, to be used by a front-end IP pool:

```
$publicIP = New-AzPublicIpAddress -Name PublicIp -ResourceGroupName TestRG -Location centralus –AllocationMethod Static -DomainNameLabel DssWebWM1
```

8. **Create the NIC** attached to a subnet, with a public facing IP, and a static private IP

```
$NIC = New-AzNetworkInterface -Name TestNic -ResourceGroupName TestRG –Location centralus -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $publicIP.Id -PrivateIpAddress "10.0.1.4"
```