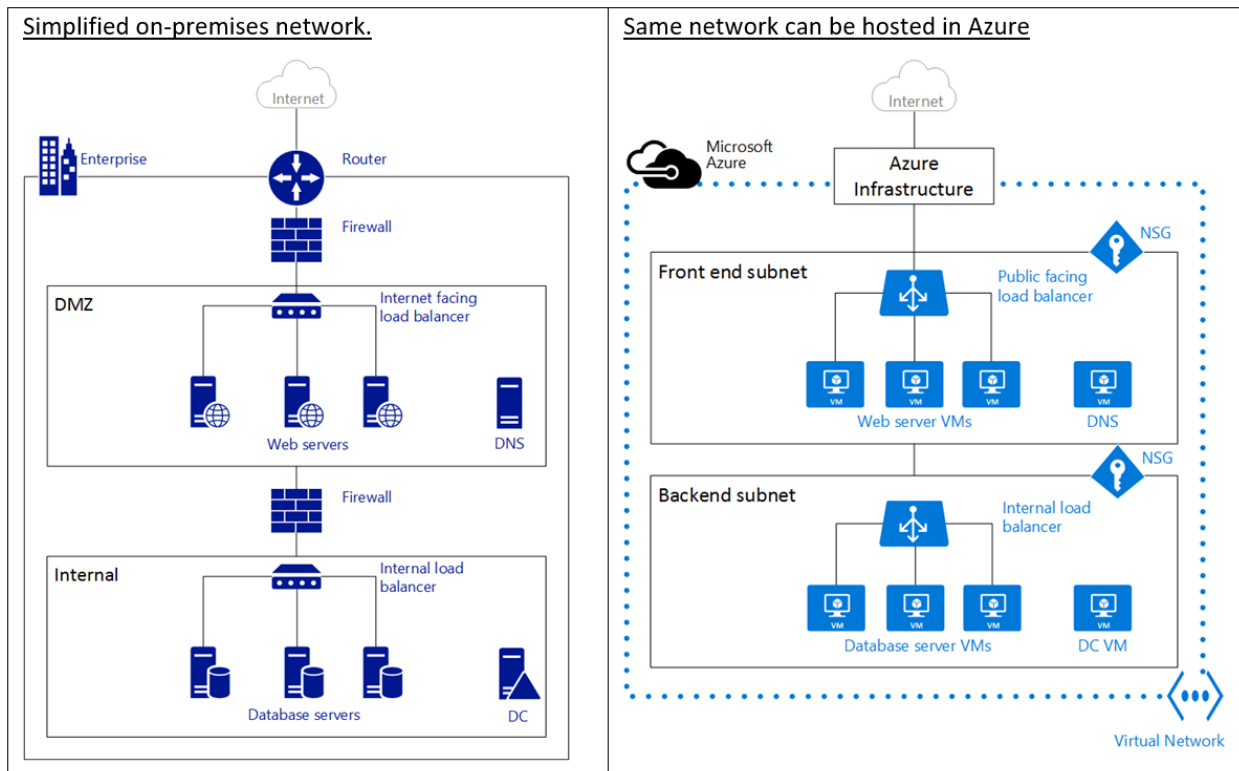


## Configure and Manage Azure Virtual Networks

- Overview of Azure Networking
- Virtual Network Benefits
- Understanding Network Resources
- Implement and manage virtual networking
- Network Security Group
- Application Security Groups

### 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.



\*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.

### 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.

### 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.

### Understanding IP Address Space

- **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<sup>8</sup> Systems – IP Address Range = 192.168.0.0/24 (Only last byte changes)

Small Network2: 192.168.1.X –for 2<sup>8</sup> 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.

202.123.56.123 to 202.123.56.123 = 202.123.56.123/32

0.0.0.0 to 255.255.255.255 = 0.0.0.0/0

192.168.0.0 to 192.168.0.255 = 192.168.0.0/24 = 256

192.168.170.0 to 192.168.170.255 = 192.168.170.0/24 = 256

172.16.0.0 to 172.16.255.255 = 172.16.0.0/16 = 256 \* 256

202.34.0.0/16 = 202.34.0.0 to 202.34.255.255 = 256 \* 256

10.0.0.0/16 = 10.0.0.0 to 10.255.255.255 = 256\*256\*256

10.0.0.0/28

10.0.0.4

10.0.0.5

10.0.0.6

10.0.1.5

10.0.1.0/24

10.0.1.4

10.0.1.5

10.0.1.6

10.0.2.0/24

10.0.3.0/24

10.0.4.0/24

10.1.0.0/16

10.1.0.0/24

10.1.0.4

10.1.0.5

10.1.0.6

10.1.1.5
10.1.1.0/24
10.1.1.4
10.1.1.5
10.1.1.6
10.1.2.0/24
10.1.3.0/24
10.1.4.0/24
10.2.0.0/16

### 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.

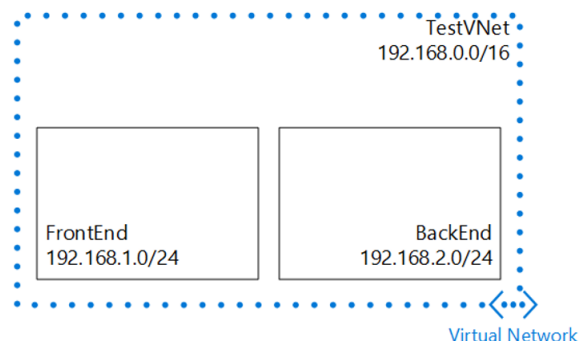
2. The default allocation method is dynamic, where the IP address is automatically allocated from the resource's subnet (using DHCP). This IP address can change when you stop and start the resource.
3. You can set the allocation method to static to ensure the IP address remains the same. In this case, 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

### Create a Virtual Network (VNet) using the Azure portal

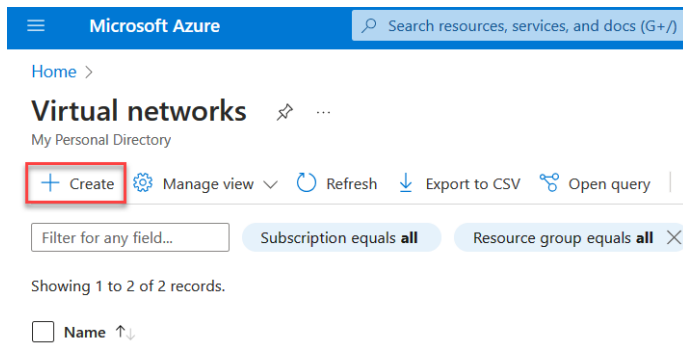
In this scenario we will create a VNet named **TestVNet** with a reserved CIDR block of **192.168.0.0/16**.

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 networks → **+Create**



Microsoft Azure

Search resources, services, and docs (G+J)

Home >

## Virtual networks

My Personal Directory

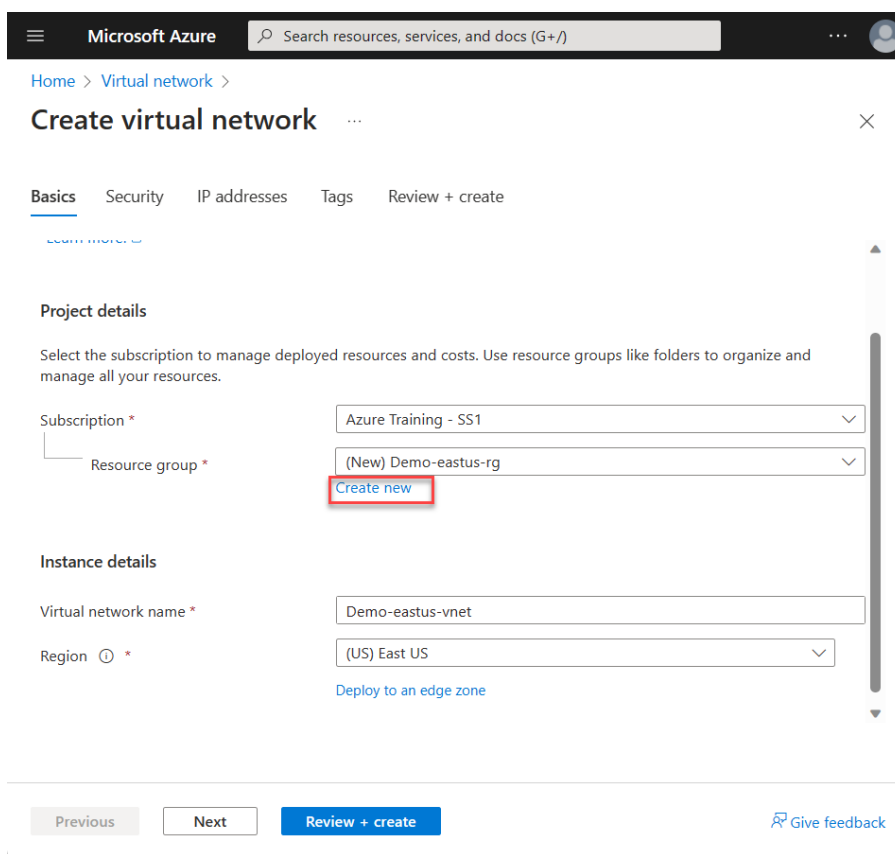
**+ Create** Manage view Refresh Export to CSV Open query

Filter for any field... Subscription equals all Resource group equals all

Showing 1 to 2 of 2 records.

☐ Name ↑↓

2. Select your Subscription, Resource Group → Create new (Demo-eastus-rg), Virtual network name=Demo-eastus-vnet, Region = East US



Microsoft Azure

Search resources, services, and docs (G+J)

Home > Virtual network >

## Create virtual network

Basics Security IP addresses Tags Review + create

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

Subscription \* Azure Training - SS1

Resource group \* (New) Demo-eastus-rg

Create new

Instance details

Virtual network name \* Demo-eastus-vnet

Region \* (US) East US

Deploy to an edge zone

Previous Next Review + create Give feedback

3. Go to IP address tab → Change CIDR block to 192.168.0.0/16, delete existing Subnet and click + Add a subnet

192.168.0.0/16 Delete address space

192.168.0.0/16 /16 (65,536 addresses)

192.168.0.0 - 192.168.255.255 (65536 addresses)

[+ Add a subnet](#)

Subnets	IP address range	Size	NAT gateway
default	10.0.0.0 - 10.0.0.255	/24 (256 addresses)	-

4. IP Address Space=192.168.0.0/16, Subnet name="Frontend-subnet", Subnet Address Range=192.168.1.0/24, click Add button

**Add a subnet**

Select an address space and configure your subnet. You can customize a default subnet select services later. [Learn more](#)

IP address space

192.168.0.0 - 192.168.255.255 (65536 addresses)

**Subnet details**

Subnet template

Name \*

Starting address \*

Subnet size

IP address space

**Security**

Simplify internet access for virtual machines by using a network address translation gate group. [Learn more](#)

NAT gateway  [Create new](#)

Network security group  [Create new](#)

[Add](#) [Cancel](#)

5. Click Review + create button, then Create button, wait for the VNet to be created. Click Go to resource button

**Add IPv4 address space**

192.168.0.0/16 Delete address space

192.168.0.0/16 /16 (65,536 addresses)

192.168.0.0 - 192.168.255.255 (65536 addresses)

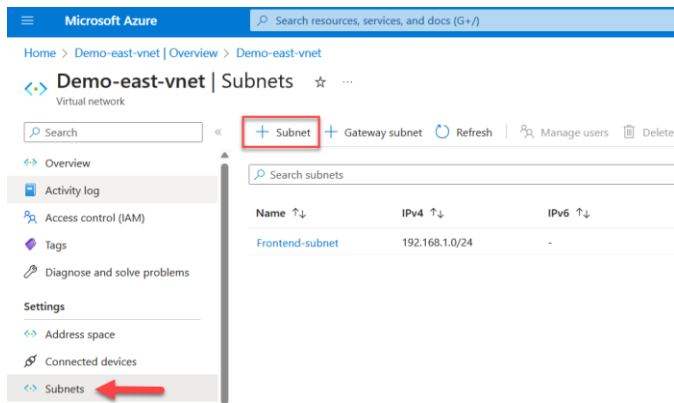
[+ Add a subnet](#)

Subnets	IP address range	Size	NAT gateway
Frontend-subnet	192.168.1.0 - 192.168.1.255	/24 (256 addresses)	-

A NAT gateway is recommended for outbound internet access from subnets. Edit the subnet to add a NAT gateway. [Learn more](#)

[Previous](#) [Next](#) [Review + create](#)

6. In the **Virtual network** blade, Settings section, click **Subnets** → + Subnet



7. Name=**Backend-subnet**, Subnet address range=192.168.2.0/24, Leave NAT Gateway, Network security group and Route table as None → Save

#### Add subnet

Name \*

Backend-subnet

Subnet address range \*

192.168.2.0/24

192.168.2.0 - 192.168.2.255 (251 + 5 Azu

☐ Add IPv6 address space

NAT gateway

None

Network security group

None

Route table

None

**SERVICE ENDPOINTS**

Create service endpoint policies to allow traffic to specific azure resources fr over service endpoints. [Learn more](#)

Services

0 selected

**SUBNET DELEGATION**

Save Cancel

8. View VNET Diagram

### Network Security Group

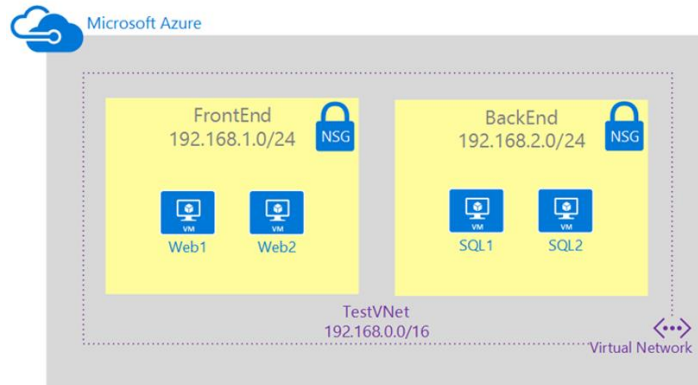
NSGs are simple, stateful packet inspection devices that use the 5-tuple (the source IP, source port, destination IP, destination port, and layer 4 protocol) approach to create **allow/deny rules** for network traffic. You allow or deny traffic to and from a single IP address, to and from multiple IP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

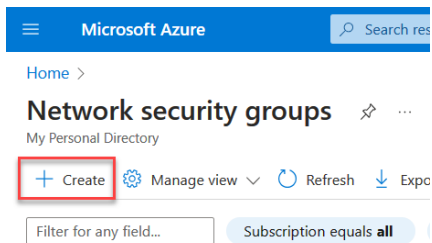


- **rdp-allow**. This rule will allow RDP (3389) traffic to the *FrontEnd* subnet.
- **web-allow**. This rule will allow HTTP (80) traffic to the *FrontEnd* subnet.
- **Backend-nsg**. The back end NSG will be applied to the *BackEnd* subnet, and contain two rules:
  - **sql-allow**. This rule allows SQL (1433) traffic only from the *FrontEnd* subnet.
  - **rdp-allow**: This rule will allow RDP (3389) traffic to the *BackEnd* subnet
  - **web-deny**. This is Outbound rule **denies** all internet bound traffic from the *BackEnd* subnet.



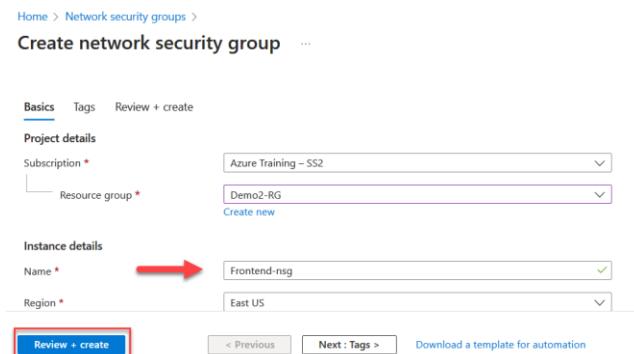
### Create NSG for Frontend-subnet:

#### 9. Search → Network Security Groups → + Create



#### 10. Name=Frontend-nsg → Review + Create button

Go to resource after creation



## Create HTTP and RDP Rules

### 11. Select Frontend-nsg → Settings →

- a. 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

- b. 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

## Associate the NSG to the FrontEnd subnet

### 12. Select **Demo-east-vnet** → Subnets → **Frontend-subnet** → Network security group → Select Frontend-nsg →

Save

### 13. Create NSG for Backend: Browse → Network Security Groups → Add → Name=Backend-nsg → Create

#### Configuring rules for Backend-subnet

#### 14. Select Backend-nsg →

- 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
- 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
- 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

**Add outbound security rule**  
Backend-nsg

Source ⓘ  
Any

Source port ranges \* ⓘ  
\*

Destination ⓘ  
Service Tag

Destination service tag ⓘ  
Internet

Service ⓘ  
HTTP

Destination port ranges ⓘ  
80

Protocol  
☐ Any  
☒ TCP  
☐ UDP  
☐ ICMP

Action  
☐ Allow  
☒ Deny

Priority \* ⓘ  
1021

Name \*  
DenyWeb

#### Associate the NSG to the BackEnd subnet

- Select Demo-eastus-vnet → 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

16. Azure portal → Search → Compute → Virtual machines

17. Click on + Create → Azure virtual machine

#### Basics tab →

- Select same Subscription, Resource group, Region as that of Virtual network created above.
- Virtual machine name = Web1-vm,
- Image = "Windows Server 2019 Datacenter x64 Gen 2"
- Username = dssadmin,
- Password = Password@123,
- Public Inbound Port = None.

## Create a virtual machine ...

The screenshot shows the 'Create a virtual machine' form in Microsoft Azure. Red arrows point to the following fields:

- Virtual machine name:** Windows-vm
- Image:** Windows Server 2019 Datacenter - x64 Gen2
- Size:** Standard\_D2s\_v3 - 2 vcpus, 8 GiB memory (₹5,503.70/month)
- Username:** dssadmin
- Public inbound ports:** Allow selected ports
- Select inbound ports:** RDP (3389)

Other visible fields include Region (US) East US, Availability options (No infrastructure redundancy required), Security type (Trusted launch virtual machines), VM architecture (x64), Run with Azure Spot discount (unchecked), Password, and Confirm password.

**Disks tab** → All Defaults

**Networking tab** →

Virtual Network = Select the earlier created VNET,

Subnet=Frontend-subnet

NSG = None

Basics Disks **Networking** Management Monitoring Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

**Network interface**

When creating a virtual machine, a network interface will be created for you.

Virtual network \*  [Create new](#)

Subnet \*  [Manage subnet configuration](#)

Public IP  [Create new](#)

NIC network security group ☒ None ☐ Basic ☐ Advanced

**Load balancing**

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

Place this virtual machine behind an existing load balancing solution? ☐

Delete public IP and NIC when VM is deleted ☐

Enable accelerated networking ☒

[Review + create](#) [< Previous](#) [Next : Management >](#)

Leave the other tabs as is.

Click **Review + Create** → Review your configuration and Click **Create**

18. Create another VM (All steps same as Web1-vm Except)

- a. Name=Database1-vm
- b. Subnet = Backend-subnet

19. Connect to Web1-vm → Server Manager → Local Server → Dashboard → Add roles and features → Next → Next → Select **Web server** → Next → Next → Install.

20. Azure Portal → Find the IP Address of VM and visit in browser: <http://<IP-of-VM>>

21. RDP to Database1-vm → Open Browser → Visit: <https://www.google.com>.

- a. Note that you will get error page.

22. From Frontend-nsg → Delete HTTP Rule

23. Azure Portal → Find the IP Address of VM and visit in browser: <http://<IP-of-VM>> (Refresh page using Ctrl+F5).

- a. Note that we will get error

#### 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

##### Database-vm

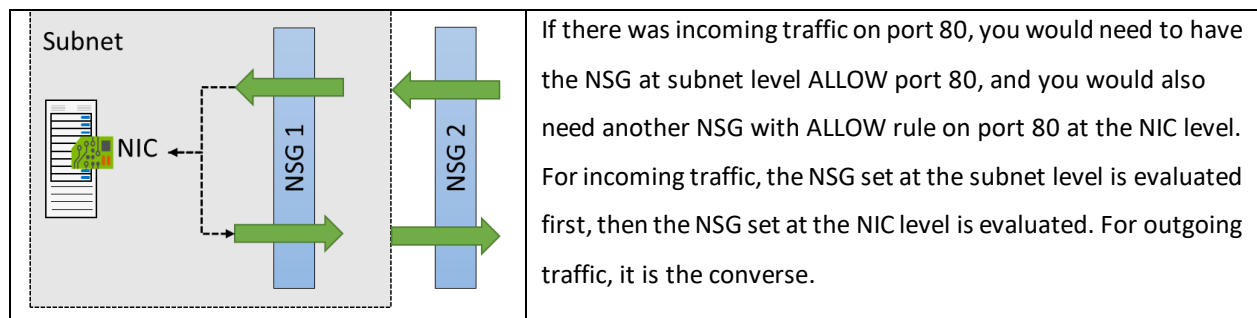
NO NSG (NIC Level)

Accessed

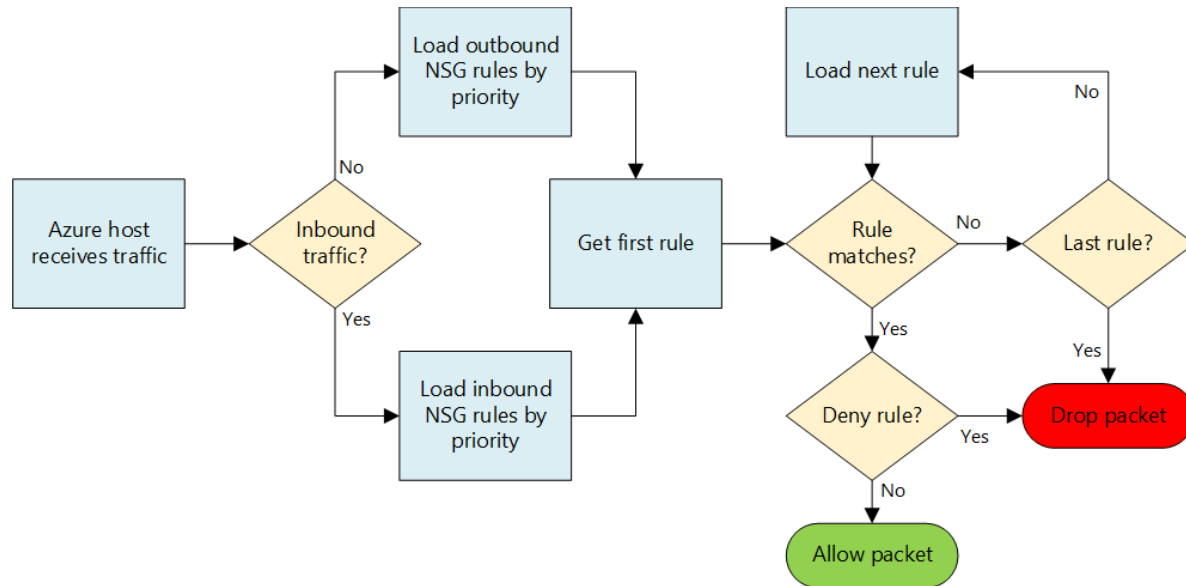
Web1 <http://<ip>> or <http://<dnsname>>

#### 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.



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**

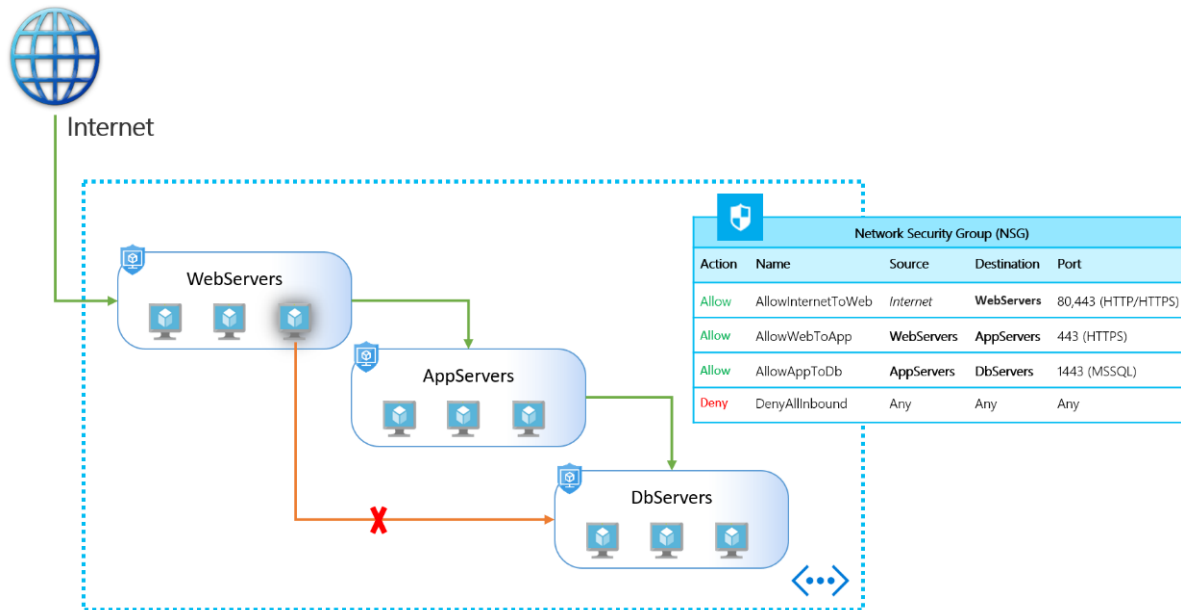
Now you get an overview which NSGs are associated with the VM's NIC and which rules are applied to it.

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.





1. Create two new Application Security Groups
  - WebServers-asg
  - DbServers-asg
2. Attach them to respective VM: VM → Networking → Application Security 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 \*

Virtual machine \*

Network interface \*

Packet details

Protocol ☒ TCP ☐ UDP

Direction ☐ Inbound ☒ Outbound

Local IP address \*  Local port \*

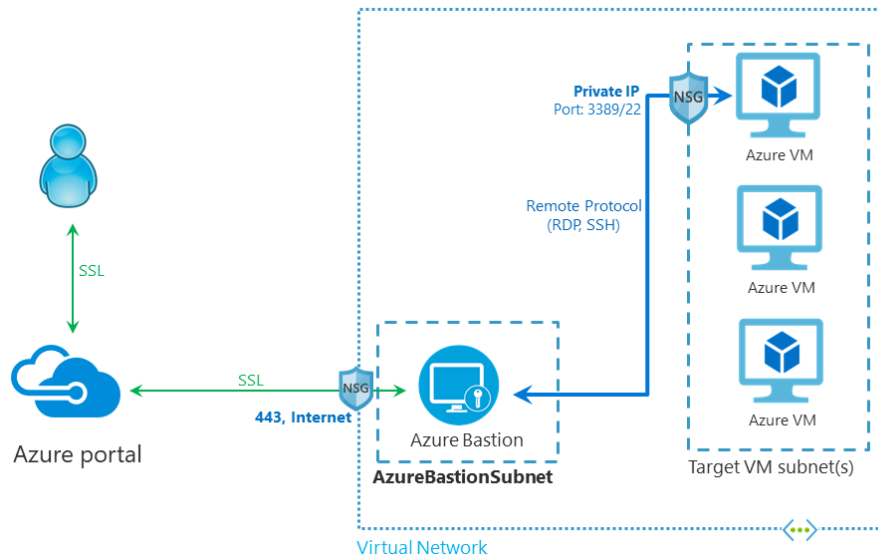
Remote IP address \*  Remote port \*

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 and in subnet **AzureBastionSubnet**.
- 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 /26 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

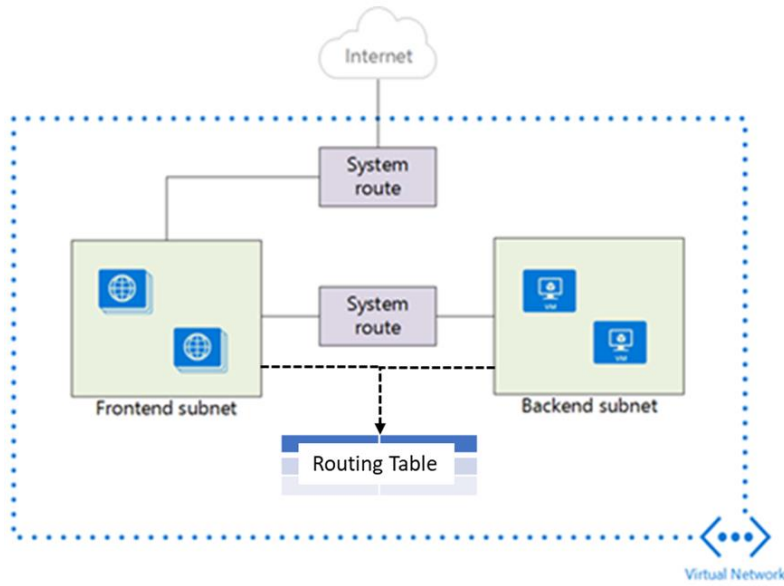
### Network Route Table

- 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.



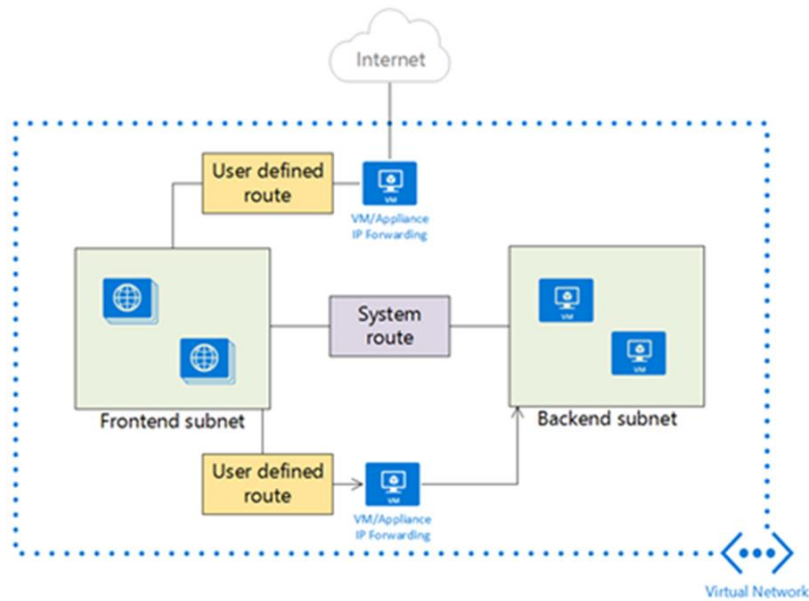
Information about the **system routes** is recorded in a **route table**. A route table contains a set of **rules**, called **routes**, that specifies how packets should be routed in a virtual network. Route tables are **associated to subnets**, and each packet leaving a subnet is handled based on the 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

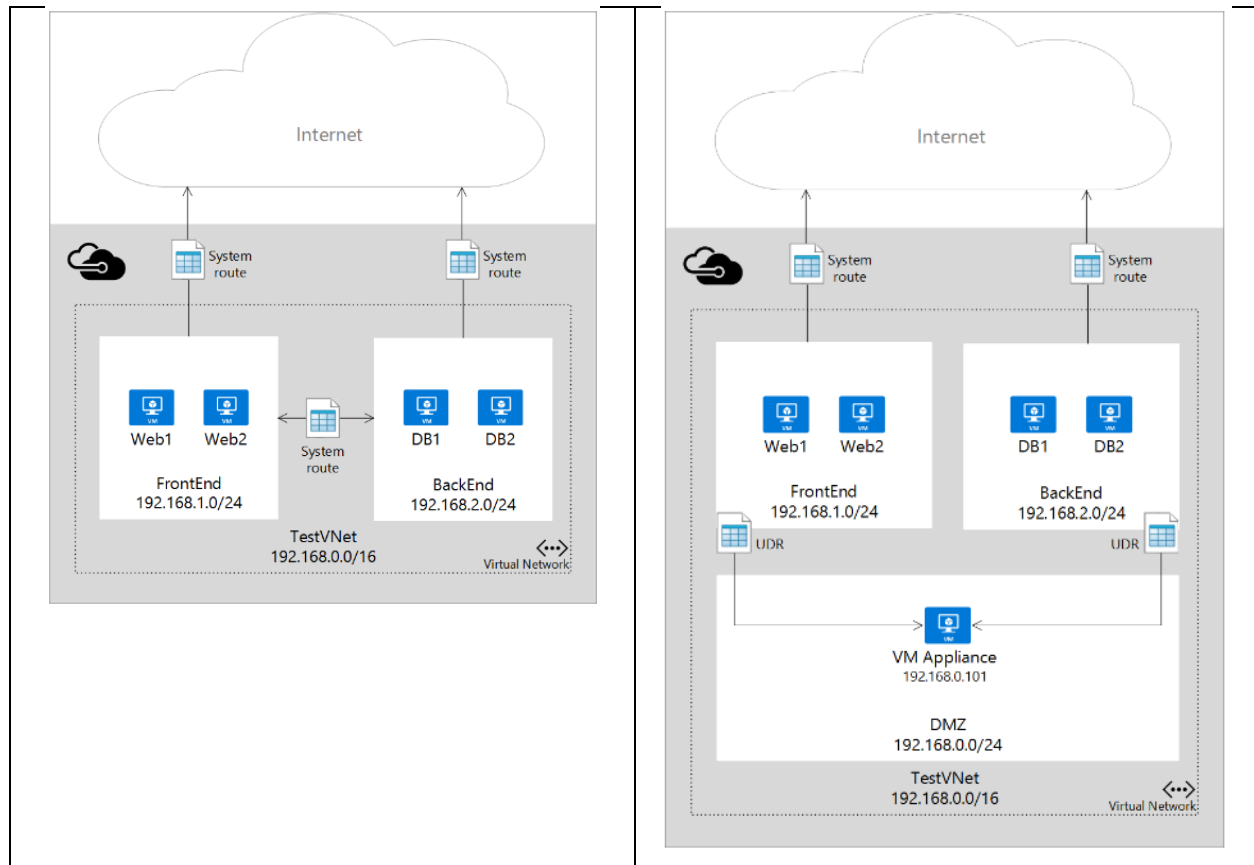
For most environments you will only need the system routes already defined by Azure.

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 (ICMP) 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

OR

**Network Watcher | Next hop** ...

Microsoft

Search

Overview

Get started

**Monitoring**

Topology

Connection monitor (classic)

Connection monitor

Network Performance Monitor

**Network diagnostic tools**

IP flow verify

NS diagnostics

**Next hop**

Effective security rules

VPN troubleshoot

Packet capture

Connection troubleshoot

**Metrics**

Usage + quotas

Subscription \* ⓘ  
Azure Training – SS2

Resource group \* ⓘ  
Demo-eastus-rg

Virtual machine \* ⓘ  
WebServer1-vm

Network interface \*  
webserver1-vm237

Source IP address \* ⓘ  
192.168.1.4

Destination IP address \* ⓘ  
192.168.2.4

**Next hop**

Result

Next hop type  
**VirtualAppliance**

IP address  
**192.168.4.4**

Route table ID  
/subscriptions/51081bf2-da0d-4...

**Rules Explained:**

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

## Azure Firewall

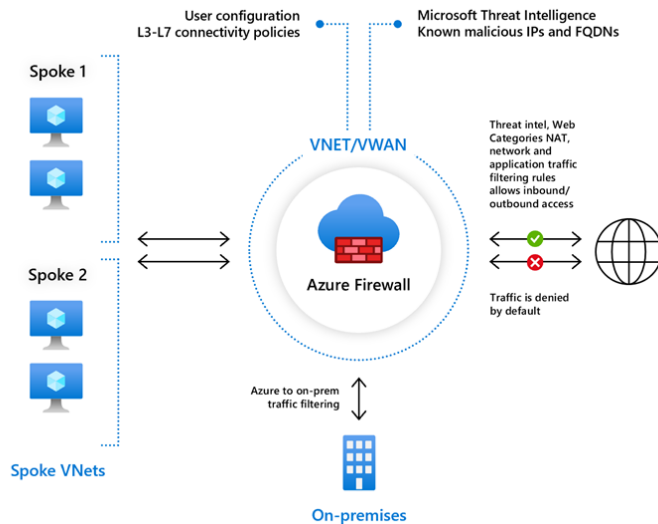
Controlling **outbound network access** is an important part of an overall network security plan. For example, you may want to **limit access to web sites**. Or, you may want to limit the outbound IP addresses and ports that can be accessed.

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** for forwarding the request on particular port of firewall to another IP and Port.

### Key Features of Azure Firewall:

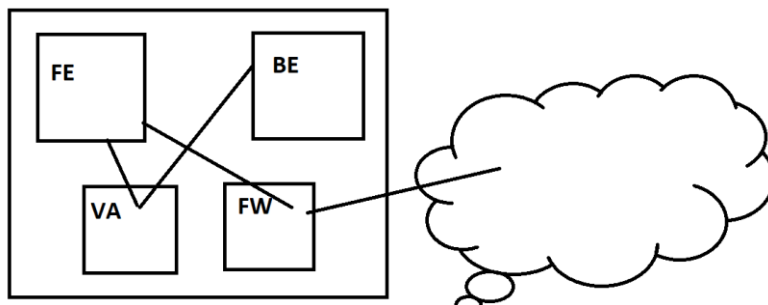
- Azure Firewall **is a managed**, cloud-based network security service that protects your Azure Virtual Network resources.
- It's a **fully stateful** firewall as a service with built-in **high availability and unrestricted cloud scalability**.
- Azure Firewall **can scale up** as much as you need to 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/TLS 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.
- **Firewall policy can be associated with Firewalls across regions regardless of where they are stored.**
- Parent policy must be in the same region as child policy.



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](http://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](http://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. Change the DNS Settings of Virtual Network : Demo-vnet → DNS Servers → Add 8.8.8.8 & 8.8.4.4 as Servers
2. **Configure Network Rule:** Allow-DNS, Protocol=**UDP**, Source Address=<IP Range of Frontendsubnet>, Destination Address="8.8.8.8, 8.8.4.4", Destination Port=**53**
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.