# Azure Networking

**Classless Inter-Domain Routing**

CIDR is a method for allocating and managing IP addresses efficiently.

CIDR Notation

IP address / Subnet mask

192.168.1.10/24 192.168.1.0-192.168.1.255

11000000.1

**Subnet Mask**

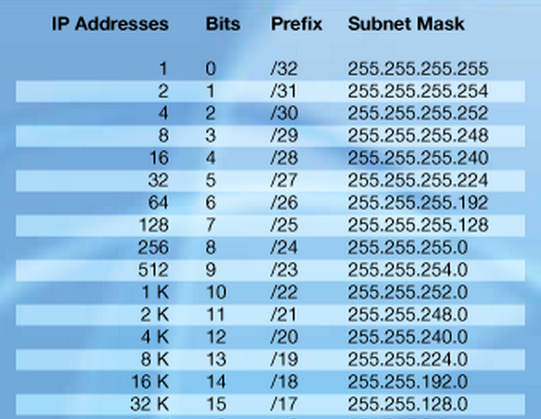
The process of dividing a network into smaller network sections is called **subnetting**.

Each address space is divided into a network portion and a host portion.

A subnet mask determines which part of an IP address is network and which part is host.

**CIDR /24 means the first 24 bits are for the network, and the last 8 bits are for hosts.**

**For the address 192.168.0.15, the 192.168.0 portion describes the network and the 15 describes the host.**

**Number of IP Addresses = 2 ^(Number of Host Bits)**

**Public and Private IP**

Public networks like the Internet communicate by using public IP addresses.

Private networks like your Azure Virtual Network use private IP addresses, which aren't routable on public networks

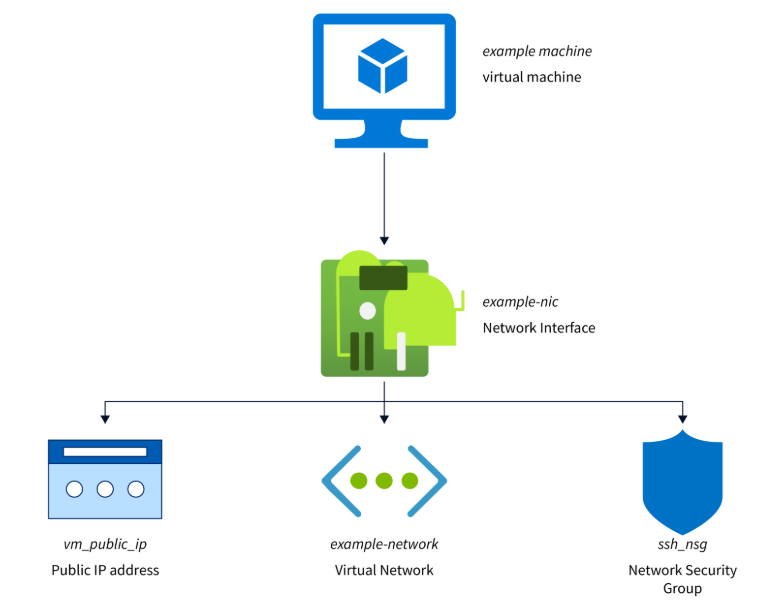
**Dynamic and Static**

Dynamic - An assigned address that can change over the lifespan of the Azure resource

Static - An assigned address that doesn't change over the lifespan of the Azure resource

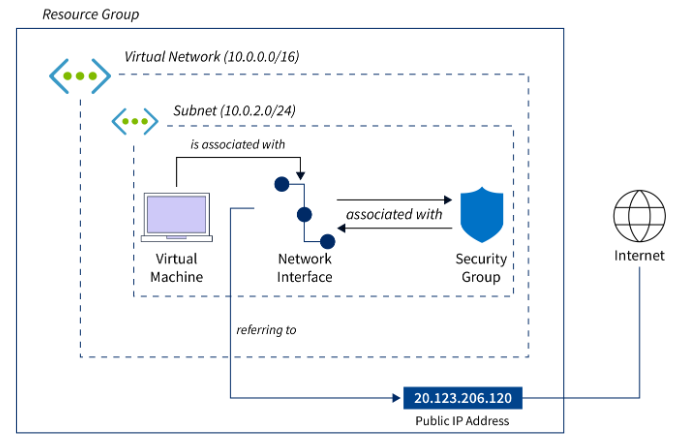
**Networking Components**

* Network Interface
* Network Security Group
* IP Address
* DNS Servers



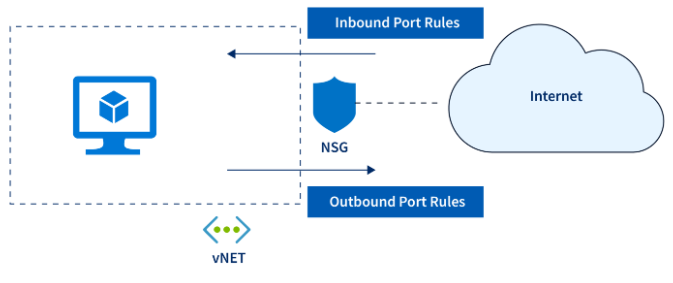
Network Interface (NIC)

* Connects a VM to a VNet.
* A gateway that allows a VM to communicate with other resources in a network.
* Manages the flow of data to and from a VM
* Each Network Interface is assigned one or more IP configurations which include ***private IP addresses, associated public IP addresses (if required), and network security group (NSG) settings.***



Network Security Group (NSG)

Acts as virtual firewalls, controlling inbound and outbound traffic to a Network Interface.



* Define inbound and outbound traffic rules.
* Filter traffic based on the source, destination, and protocol.
* Work with route tables to ensure that traffic is directed as intended.

Domain Name System (DNS) Servers

* Essential for translating human-readable domain names into IP addresses.
* Configuring DNS servers for a Network Interface ensures that it can seamlessly communicate with resources both within and outside the Azure environment.

**Demo**

**Understand Azure Network Security Groups (NSGs) and networking components.**

**AzCLI**

Create virtual network

*az network vnet create --resource-group myRGNetwork --name myVNet --address-prefix 10.0.0.0/16 --subnet-name myFrontendSubnet --subnet-prefix 10.0.1.0/24*

Create sub network

*az network vnet subnet create --resource-group myRGNetwork --vnet-name myVNet --name myBackendSubnet --address-prefix 10.0.2.0/24*

Create NSG

*az network nsg create --resource-group myRGNetwork --name myBackendNSG*

Configure NSG Rules

✅ Add Inbound Rule to Allow SSH (22)

1. Open MyNSG → Click Inbound Security Rules
2. Click Add Rule:
   * Name: Allow-SSH
   * Priority: 100
   * Source: Any
   * Destination: Any
   * Service: SSH
   * Action: Allow
3. Click Add

✅ Add Outbound Rule to Deny HTTPS (80)

1. Go to Outbound Security Rules
2. Click Add Rule:
   * Name: Deny-HTTPS-Out
   * Priority: 100
   * Protocol: HTPS
   * Action: Deny
3. Click Add

Associate NSG with a Subnet

1. Navigate to the NSG you just created.
2. Under "Settings," select "Subnets."
3. Click on "Associate" and select the virtual network and subnet you want to associate with the NSG.
4. Click "OK" to complete the association.

Configure NSG Rules

1. Navigate to the NSG you just created.
2. Under "Settings," select "Inbound security rules" or "Outbound security rules."
3. Click on "+ Add" to create a new rule.
4. Fill in the required details, such as source, destination, port, and protocol.
5. Click "Add" to create the rule.

Test Inbound

ssh azureuser@<VM\_Public\_IP> PASS

Delete Rule and test again

Test Outbound

curl <https://example.com> – FAIL

Delete Rule and try – Default is Allow All so PASS

Test SSH connection over TCP/UDP

nc -zvu 4.247.166.57 22

Connection to 4.247.166.57 22 port [udp/\*] succeeded!

# Private Endpoint

A private endpoint is a network interface that uses a private IP address from your virtual network. This network interface connects you privately and securely to a service that's powered by Azure Private Link. By enabling a private endpoint, you're bringing the service into your virtual network.

The service could be an Azure service such as:

* Azure Storage
* Azure Cosmos DB
* Azure SQL Database
* Your own service, using [Private Link service](https://learn.microsoft.com/en-us/azure/private-link/private-link-service-overview).

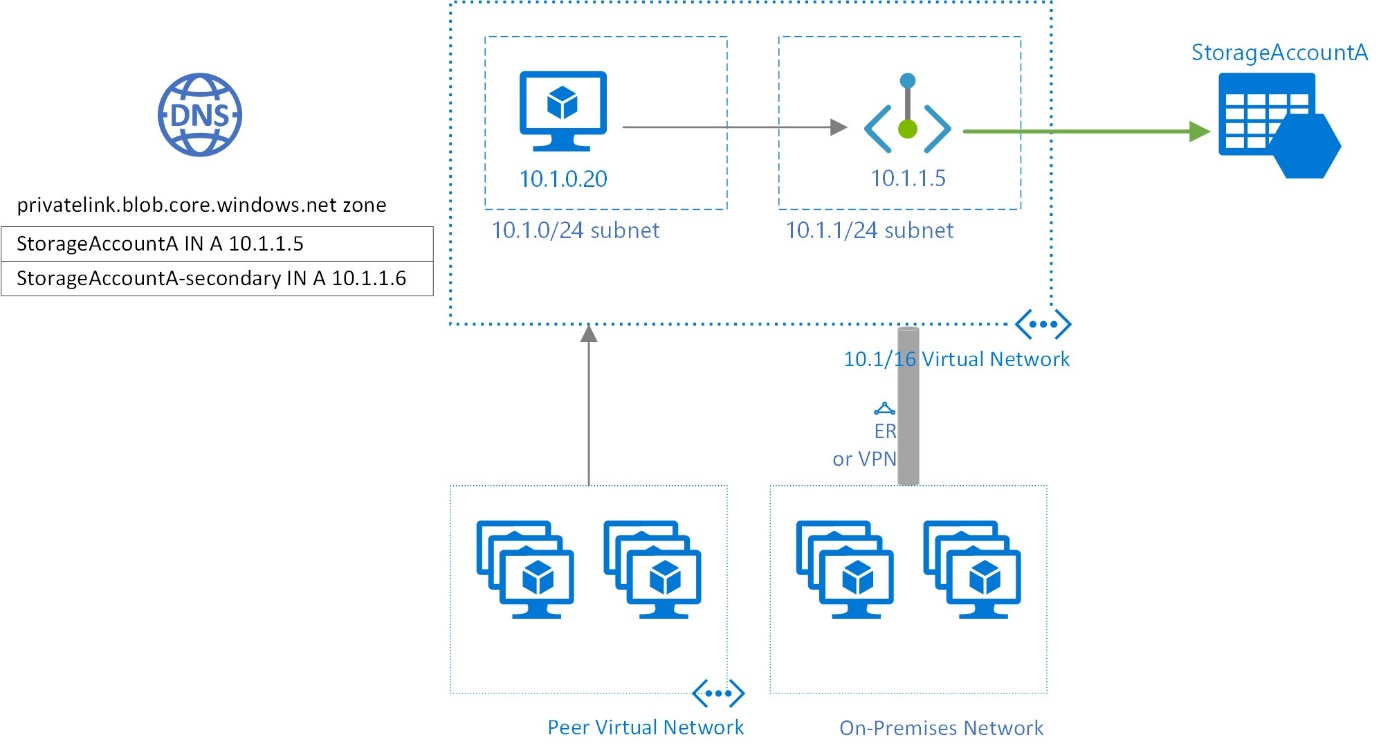
# Private Link

* Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a [private endpoint](https://learn.microsoft.com/en-us/azure/private-link/private-endpoint-overview) in your virtual network.
* Traffic between your virtual network and the service travels the Microsoft backbone network.
* Setup and consumption using Azure Private Link is consistent across Azure PaaS, customer-owned, and shared partner services.

<https://learn.microsoft.com/en-us/azure/storage/common/storage-private-endpoints?source=recommendations>

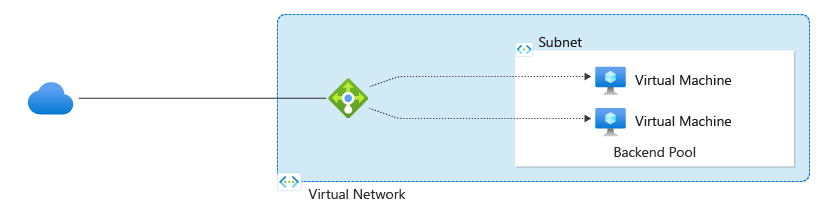
**LAB**

<https://learn.microsoft.com/en-us/azure/private-link/tutorial-private-endpoint-storage-portal>



# Load Balancers

* *Load balancing* refers to efficiently distributing incoming network traffic across a group of backend servers or resources.
* Operates at layer 4 of the (OSI) model.
* The service distributes inbound flows that arrive at the load balancer's frontend to backend pool instances.
* These flows are distributed according to configured load-balancing rules and health probes.
* The backend pool instances can be Azure virtual machines (VMs) or virtual machine scale sets.



[public load balancer](https://learn.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations)

can provide both inbound and outbound connectivity for the VMs inside your virtual network.

For inbound traffic scenarios, Azure Load Balancer can load balance internet traffic to your VMs.

For outbound traffic scenarios, the service can translate the VMs' private IP addresses to public IP addresses for any outbound connections that originate from your VMs.

[internal (or private) load balancer](https://learn.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configurations)

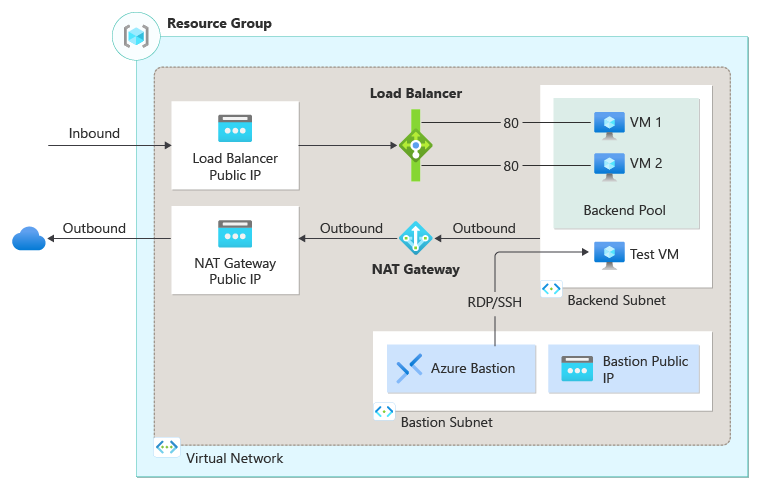
can provide inbound connectivity to your VMs in private network connectivity scenarios, such as accessing a load balancer frontend from an on-premises network in a hybrid scenario. Internal load balancers are used to load balance traffic inside a virtual network.

Load Balancer Components

1. [Frontend IP configuration](https://learn.microsoft.com/en-us/azure/load-balancer/components#frontend-ip-configuration-)
2. [Backend pool](https://learn.microsoft.com/en-us/azure/load-balancer/components#backend-pool)
3. [Health probes](https://learn.microsoft.com/en-us/azure/load-balancer/components#health-probes)
4. [Load Balancer rules](https://learn.microsoft.com/en-us/azure/load-balancer/components#load-balancer-rules)
5. [High Availability Ports](https://learn.microsoft.com/en-us/azure/load-balancer/components#high-availability-ports)
6. [Inbound NAT rules](https://learn.microsoft.com/en-us/azure/load-balancer/components#inbound-nat-rules)
7. [Outbound rules](https://learn.microsoft.com/en-us/azure/load-balancer/components#outbound-rules)

<https://learn.microsoft.com/en-us/azure/load-balancer/tutorial-cross-region-portal?tabs=azureportal>

<https://learn.microsoft.com/en-us/azure/load-balancer/quickstart-load-balancer-standard-public-portal?source=recommendations>

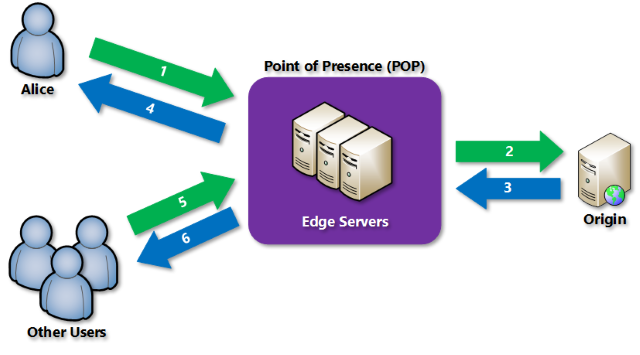


Install-WindowsFeature -name Web-Server -IncludeManagementTools

Remove-Item C:\inetpub\wwwroot\iisstart.htm

Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)

# CDN



* A content delivery network is a distributed network of servers that can efficiently deliver web content to users.
* A content delivery network store cached content on edge servers in point of presence (POP) locations that are close to end users, to minimize latency.
* It offers developers a global solution for rapidly delivering high-bandwidth content to users by caching their content at strategically placed physical nodes across the world.

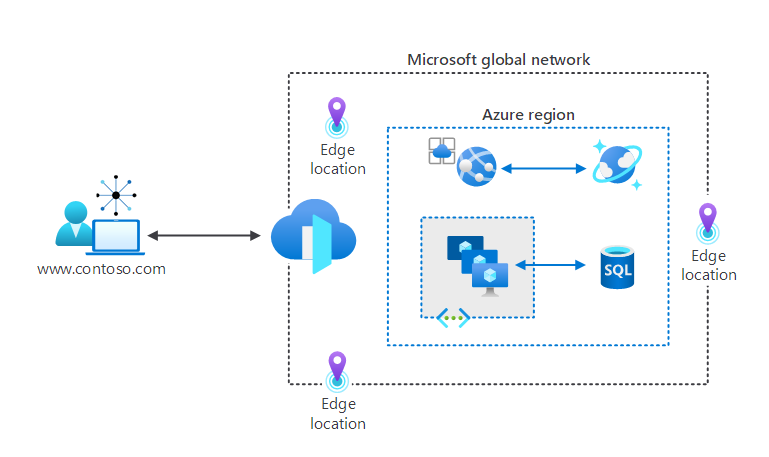
# Frontdoor

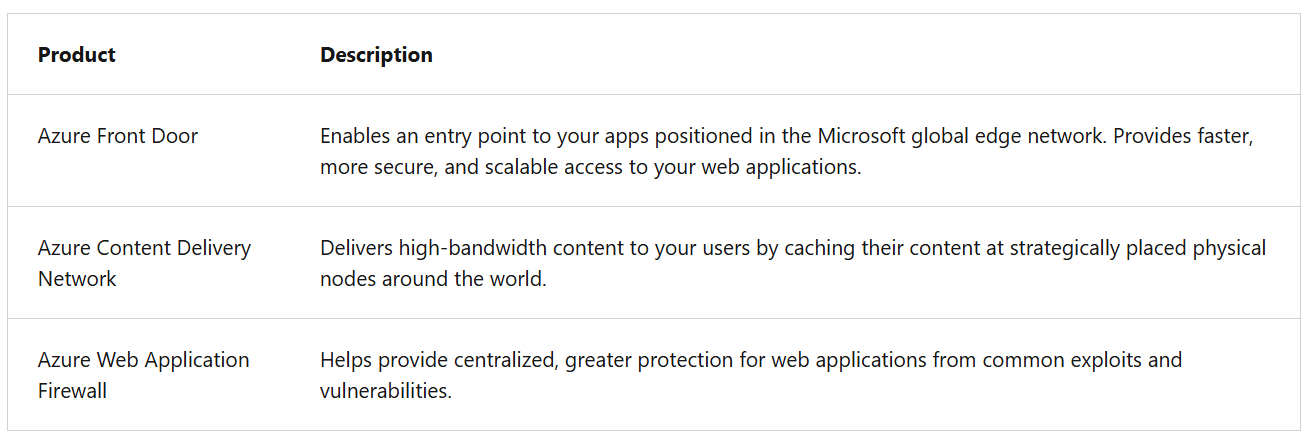
<https://learn.microsoft.com/en-us/azure/frontdoor/front-door-overview>

<https://learn.microsoft.com/en-us/azure/frontdoor/create-front-door-portal?tabs=quick>

<https://learn.microsoft.com/en-us/azure/frontdoor/best-practices?source=recommendations>

<https://learn.microsoft.com/en-us/training/modules/intro-to-azure-front-door/2-what-is-azure-front-door>

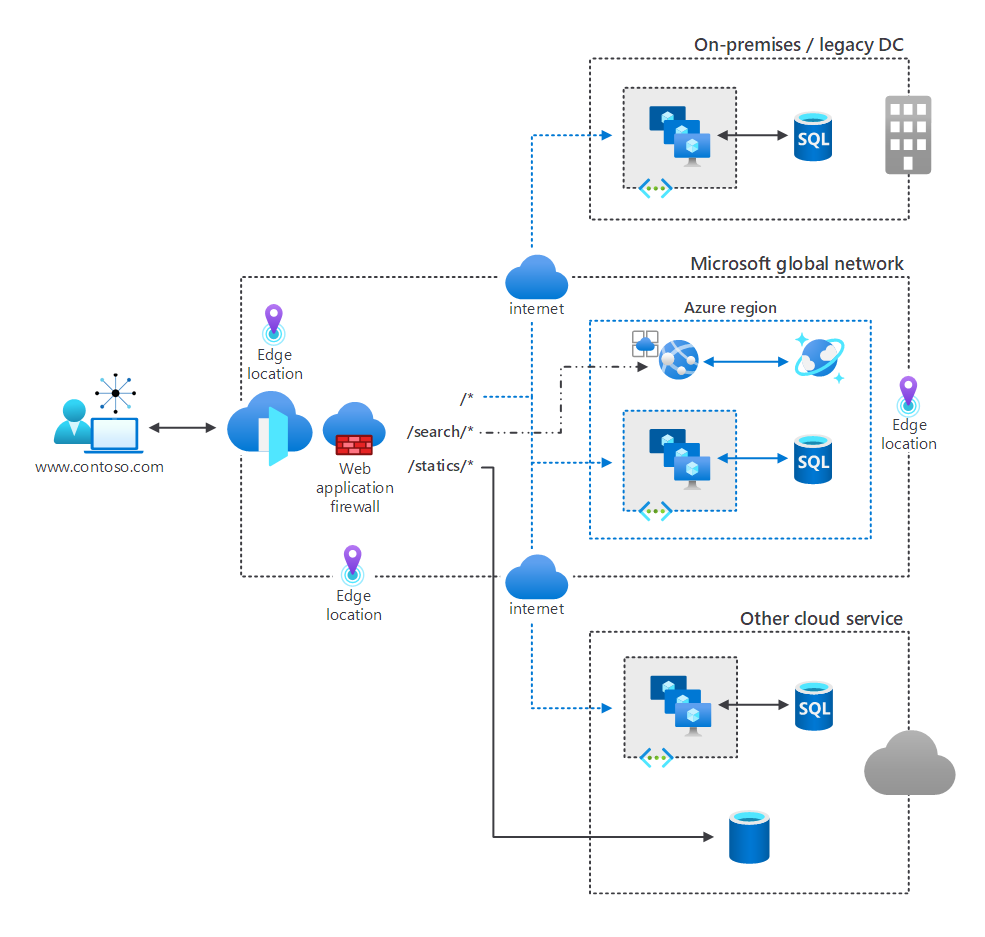




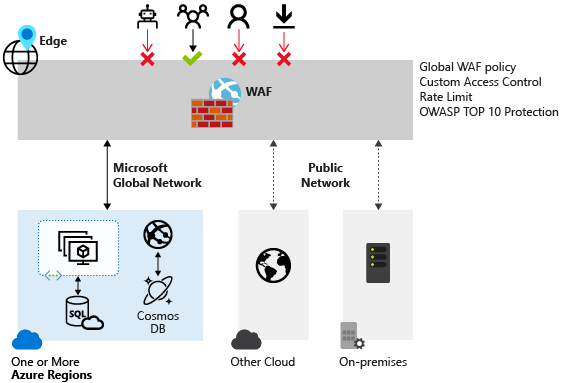
Azure Front Door is Microsoft's advanced cloud Content Delivery Network (CDN) designed to provide fast, reliable, and secure access to your applications' static and dynamic web content globally.

It offers a fast, reliable, and more secure modern cloud CDN by using the Microsoft global edge network to integrate with intelligent threat protection.

Users connect to your application through the Microsoft global network. Azure Front Door then routes user requests to the fastest and most available application backend.



# WAF



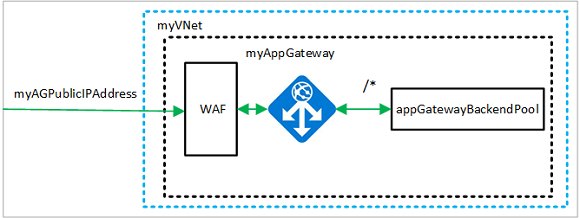
Inspects every incoming request delivered by Azure Front Door at the network edge.

Prevents malicious attacks close to the attack sources before they enter your VNet

Azure Web Application Firewall protects your:

* Web applications from web vulnerabilities and attacks
* Web applications from malicious bots

# Application Gateway



<https://learn.microsoft.com/en-us/azure/web-application-firewall/ag/application-gateway-web-application-firewall-portal>

Backend Pool – VM1, VM2

App GW – IP/app1, IP/app2

sudo apt update && sudo apt install -y nginx

sudo mkdir -p /var/www/html/app1

echo "This is App1" | sudo tee /var/www/html/app1/index.html

sudo systemctl restart nginx

sudo apt update && sudo apt install -y nginx

sudo mkdir -p /var/www/html/app2

echo "This is App2" | sudo tee /var/www/html/app2/index.html

sudo systemctl restart nginx