[**Introduction to Azure virtual networks**](https://docs.microsoft.com/en-us/learn/modules/introduction-to-azure-virtual-networks/?ns-enrollment-type=LearningPath&ns-enrollment-id=learn.wwl.designing-implementing-microsoft-azure-networking-solutions-az-700)

[**Explore Azure Virtual Networks**](https://docs.microsoft.com/en-us/learn/modules/introduction-to-azure-virtual-networks/2-explore-azure-virtual-networks/?ns-enrollment-type=LearningPath&ns-enrollment-id=learn.wwl.designing-implementing-microsoft-azure-networking-solutions-az-700)

All resources in a VNet can communicate outbound to the internet, by default. You can communicate inbound to a resource by assigning a public IP address or a public Load Balancer.

There are three key mechanisms through which Azure resource can communicate: VNets, VNet service endpoints and VNet peering

You can connect your on-premises computers and networks to a virtual network using any of the following options: Point-to-site virtual private network (VPN), Site-to-site VPN, Azure ExpressRoute.

* + You can filter network traffic between subnets using any combination of network security groups and network virtual appliances like firewalls, gateways, proxies, and Network Address Translation (NAT) services.

The smallest supported IPv4 subnet is /29, and the largest is /8 (using CIDR subnet definitions). IPv6 subnets must be exactly /64 in size

All Azure resource types have a scope that defines the level that resource names must be unique. There are four levels you can specify a scope: [management group](https://docs.microsoft.com/en-us/azure/governance/management-groups/overview), subscription, [resource group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview), and resource. Scopes are hierarchical, with each level of hierarchy making the scope more specific.

Basic DDoS protection is integrated into the Azure platform by default and at no additional cost. Standard DDoS Protection is a paid service that offers enhanced DDoS mitigation capabilities via adaptive tuning, attack notification, and telemetry for all protected resources within a virtual network.

**Configure public IP services**

Basic and Standard IPs have an adjustable inbound originated flow idle timeout of 4-30 minutes, with a default of 4 minutes, and a fixed outbound originated flow idle timeout of 4 minutes.

Basic IPs can be allocated statically or dynamically and are open by default, so the use of Network security groups is recommended but optional for restricting inbound or outbound traffic. They do not support availability zone scenarios.

Standard SKU public IP addresses always use the static allocation method. Standard IPs are secure by default and closed to inbound traffic. You must explicitly allow inbound traffic by using a network security group. Standard IPs are zone-redundant by default and optionally zonal.

If your business needs to have datacenters in different regions, you will have a different public IP address range for each region. You can use technology like Azure Traffic Manager to balance traffic between region-specific instances.

**Design name resolution for your virtual network**

A DNS zone hosts the DNS records for a domain. So, to start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone.

* + The same zone name can be reused in a different resource group or a different Azure subscription.
  + The name of the zone must be unique within the resource group, and the zone must not exist already.
  + The same zone name can be reused in a different resource group or a different Azure subscription.
  + Where multiple zones share the same name, each instance is assigned different name server addresses.
  + Root/Parent domain is registered at the registrar and pointed to Azure NS.
  + Child domains are registered in AzureDNS directly.

To delegate your domain to Azure DNS, you first need to know the name server names for your zone. Each time a DNS zone is created Azure DNS allocates name servers from a pool. Once the Name Servers are assigned, Azure DNS automatically creates authoritative NS records in your zone. Once the DNS zone is created, and you have the name servers, you need to update the parent domain. In the registrar’s DNS management page, edit the NS records and replace the NS records with the ones Azure DNS created (MUST use all four name servers).

Setting up a subdomain follows the same process as typical delegation. The only difference is that NS records must be created in the parent zone (e.g. contoso.com) in Azure DNS, rather than in the domain registrar.

A record set is a collection of records in a zone that have the same name and are the same type. A record set cannot contain two identical records. Empty record sets (with zero records) can be created, but do not appear on the Azure DNS name servers. Record sets of type CNAME can contain one record at most.

The Add record set page will change depending on the type of record you select. For an A record, you will need the TTL (Time to Live) and IP address. The time to live, or TTL, specifies how long each record is cached by clients before being requeried.

DNS servers within a virtual network can forward DNS queries to the recursive resolvers in Azure. This enables you to resolve host names within that virtual network. For example, a domain controller (DC) running in Azure can respond to DNS queries for its domains and forward all other queries to Azure. Forwarding queries allows VMs to see both your on-premises resources (via the DC) and Azure-provided host names (via the forwarder). Access to the recursive resolvers in Azure is provided via the virtual IP 168.63.129.16.

DNS forwarding also enables DNS resolution between virtual networks and allows your on-premises machines to resolve Azure-provided host names. In order to resolve a VM's host name, the DNS server VM must reside in the same virtual network and be configured to forward host name queries to Azure. Because the DNS suffix is different in each virtual network, you can use conditional forwarding rules to send DNS queries to the correct virtual network for resolution. The following image shows two virtual networks and an on-premises network doing DNS resolution between virtual networks, by using this method.

<pz/o-o-l-ot 
naua u, s•AA 

Internal DNS defines a namespace as follows: .internal.cloudapp.net.

Any VM created in the VNet is registered in the internal DNS zone and gets a DNS domain name like myVM.internal.cloudapp.net. It's important to recognize that it's the Azure Resource name that is registered, not the name of the guest OS on the VM.

Private DNS zones in Azure are available to internal resources only. They are global in scope, so you can access them from any region, any subscription, any VNet, and any tenant.

For scenarios which require more flexibility than Internal DNS allows, you can create your own private DNS zones. These zones enable you to:

* + Configure a specific DNS name for a zone.
  + Create records manually when necessary.
  + Resolve names and IP addresses across different zones.
  + Resolve names and IP addresses across different VNets.

Two ways to link VNets to a private zone:

* + Registration: Each VNet can link to one private DNS zone for registration. However, up to 100 VNets can link to the same private DNS zone for registration.
  + Resolution: There may be many other private DNS zones for different namespaces. You can link a VNet to each of those zones for name resolution. Each VNet can link to up to 1000 private DNS Zones for name resolution.

Private DNS 
Zone 
Resolution 
Link up to 1000 
zones for resolution 
Private DNS Zone 
Registration 
Link to 1 zone only 
for registration 
VNet 

Note: to link the Vnet to a private DNS Zone, this configuration is done via the private DNS zone resource by creating a virtual link

Note: the 'Enable auto registration' option during the creation of a virtual network link in Private DNS Zone, enables automatic creation of DNS records in the zone for the VMs connected to the Vnet.

**Enable cross-virtual network connectivity with peering**

 There are two types of VNet peering.

* + Regional VNet peering connects Azure virtual networks in the same region.
  + Global VNet peering connects Azure virtual networks in different regions. When creating a global peering, the peered virtual networks can exist in any Azure public cloud region or China cloud regions, but not in Government cloud regions. You can only peer virtual networks in the same region in Azure Government cloud regions.

Region 1 
VNet1 
Region 2 
Regional VNet 
Global 
VNet2 
VNet Peering 
VNet3 

The next hop address for Global Vnet peer is VNetGlobalPeering.

Gateway transit is supported for both VNet Peering and Global VNet Peering.

**Implement virtual network traffic routing**

Service endpoints are enabled for individual subnets within a virtual network, so the route is only added to the route table of a subnet a service endpoint is enabled for. The public IP addresses of Azure services change periodically, and Azure manages the updates to the routing tables when necessary.

n Azure, each subnet can have zero or one associated route table. When you create a route table and associate it to a subnet, the routes within it are combined with, or override, the default routes Azure adds to a subnet.

You can specify the following next hop types when creating a user-defined route:

* + Virtual appliance
  + Virtual network gateway
  + None
  + Virtual network
  + Internet

When creating a subnet, Routes are automatically added to the route table for all subnets with Virtual network gateway propagation enabled. When you are using ExpressRoute, propagation ensures all subnets get the routing information.

You can view the effective routes for a network interface with the Get-AzEffectiveRouteTable command.

Forced tunneling lets you redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel for inspection and auditing. If you don't configure forced tunneling, Internet-bound traffic from your VMs in Azure always traverses from the Azure network infrastructure directly out to the Internet, without the option to allow you to inspect or audit the traffic. Forced tunneling can be configured by using Azure PowerShell. It can't be configured using the Azure portal. Forced tunneling in Azure is configured using virtual network custom user-defined routes.

Forced tunneling must be associated with a VNet that has a route-based VPN gateway.

**Configure internet access with Azure Virtual NAT**

NAT is compatible with standard SKU public IP address resources or public IP prefix resources or a combination of both. You can use a public IP prefix directly or distribute the public IP addresses of the prefix across multiple NAT gateway resources.

NAT provides up to 64,000 concurrent flows for UDP and TCP respectively, for each attached public IP address. NAT can support up to 16 public IP addresses

NAT gateway resource can be regional or zonal (zone-isolated)

To deploy NAT: Create regional or zonal (zone-isolated) NAT gateway resource, Assign IP addresses, Configure virtual network subnet to use a NAT gateway. User-defined routes are not necessary.

NAT is compatible with standard SKU public IP, public IP prefix, and load balancer resources. Basic resources (for example basic load balancer) and any products derived from them aren't compatible with NAT. Basic resources must be placed on a subnet not configured with NAT.

NAT can't be deployed on a subnet with an IPv6 prefix.

NAT can't span multiple virtual networks.

[**Design and implement hybrid networking**](https://docs.microsoft.com/en-gb/learn/modules/design-implement-hybrid-networking/?ns-enrollment-type=LearningPath&ns-enrollment-id=learn.wwl.designing-implementing-microsoft-azure-networking-solutions-az-700)

**Design and implement Azure VPN Gateway**

VPN gateways can also be used for connections between virtual networks in Azure.

Within each virtual network gateway there are two or more virtual machines (VMs). These VMs have been deployed to a special subnet that you specify, called the gateway subnet. They contain routing tables for connections to other networks, along with specific gateway services. VPN Gateways require a gateway subnet. You can create a Gateway subnet before you create a VPN gateway, or you can create it during the creation of the VPN Gateway. You should not deploy any additional resources into the gateway subnet.

When you're planning a VPN gateway, there are three architectures to consider:

* + Point to site over the internet
  + Site to site over the internet
  + Site to site over a dedicated network, such as Azure ExpressRoute

The resizing of VpnGw SKUs is allowed within the same generation, except resizing of the Basic SKU.

P2S connection requires a RouteBased VPN type.

If you want to create a S2S VPN gateway connection and a P2S VPN gateway connection for the same virtual network, use VPN type RouteBased because P2S requires a RouteBased VPN type.

Once a virtual network gateway has been created, you can't change the VPN type. You must delete the virtual network gateway and create a new one.

**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. PolicyBased VPNs can only be used on the Basic gateway SKU. This VPN type is not compatible with other gateway SKUs.

**RouteBased VPNs** use "routes" in the IP forwarding or routing table to direct packets into their corresponding tunnel interfaces. The policy (or traffic selector) for RouteBased VPNs are configured as any-to-any (or wild cards).

To provide better availability for your VPN connections, there are a few options available:

* + VPN Gateway redundancy (Active-standby)
  + Multiple on-premises VPN devices
  + Active-active Azure VPN gateway
  + Combination of both

Every Azure VPN gateway consists of two instances in an active-standby configuration. The standby instance would take over (failover) automatically and resume the S2S VPN or VNet-to-VNet connections.

In Active-Active 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.

**Connect networks with Site-to-site VPN connections**

VM 
VM 
Gateway 
On-premises 
network 
Azure Virtual Network 
Gateway 
subnet 
Gateway 
Internet 
Virtual Network 
VM 
Management subnet 
Jumpbox 

**Connect devices to networks with Point-to-site VPN connections**

Point-to-site VPN can use one of the following protocols:

* + OpenVPN® Protocol, an SSL/TLS based VPN protocol. OpenVPN can be used to connect from Android, iOS (versions 11.0 and above), Windows, Linux, and Mac devices (macOS versions 10.13 and above).
  + Secure Socket Tunneling Protocol (SSTP), a proprietary TLS-based VPN protocol. SSTP is only supported on Windows devices.
  + IKEv2 VPN, a standards-based IPsec VPN solution. IKEv2 VPN can be used to connect from Mac devices (macOS versions 10.11 and above).

 There are two mechanisms that Azure offers to authenticate a connecting point-to-site user;

* + Authenticate using native Azure certificate authentication
  + Authenticate using native Azure Active Directory authentication or using Active Directory (AD) Domain Server

Native Azure AD authentication is only supported for OpenVPN protocol and Windows 10 and requires the use of the Azure VPN Client. With native Azure AD authentication, you can leverage Azure AD's conditional access as well as multifactor authentication (MFA) features for VPN.

AD Domain authentication is a popular option because it allows users to connect to Azure using their organization domain credentials. It requires a RADIUS server that integrates with the AD server. Organizations can also leverage their existing RADIUS deployment.

The RADIUS server is deployed either on-premises or in your Azure VNet. During authentication, the Azure VPN Gateway passes authentication messages back and forth between the RADIUS server and the connecting device. Thus, the Gateway must be able to communicate with the RADIUS server. If the RADIUS server is present on-premises, then a VPN S2S connection from Azure to the on-premises site is required for reachability. The RADIUS server can also integrate with AD certificate services. This lets you use the RADIUS server and your enterprise certificate deployment for P2S certificate authentication.

ASN 65010 
10.11.0.0/16 
Azure VPN 
P2S VPN 
Tunnel 
IPsec/1KE S2S VPN 
Tunnel 
VPN Client Address 
192.168.0.11 
On Premises 
10.51.0.0/16 
10.52.0.0/16 
onprem VPN 
RADIUS Server 
AD Domain 
Services 

**Connect remote resources by using Azure Virtual WANs**

A Virtual WAN is a security delineation; each instance of a Virtual WAN is self-contained in terms of connectivity and hence also provides security isolation.

Each Virtual WAN is implemented as a hub-and-spoke topology, and can have one or more hubs, which support connectivity between different types of endpoints including connectivity vendors like AT&T, Verizon, and T-Mobile. All hubs are connected in a full mesh topology in a Standard Virtual WAN.

A secured virtual hub is an Azure Virtual WAN Hub with associated security and routing policies configured by Azure Firewall Manager. You can use a secured virtual hub to filter traffic between virtual networks (V2V), virtual networks and branch offices (B2V) and traffic to the Internet (B2I/V2I). A secured virtual hub provides automated routing. There's no need to configure your own UDRs (user defined routes) to route traffic through your firewall.

You can have multiple virtual Hubs, but only one virtual hub per region. When you create a virtual hub, it creates a virtual hub VNet and a virtual hub vpngateway.

Region I 
Express Route 
Ill 
Branch 
Hub-to- Hub 
Connection 
Site-to-Site VPN 
Branch 
VNet 
Site-to-Ste VPN 
Branch 
Branch 
2 
Point-to.Site VPN 
Remote Users 

Gateway scale units allow you pick the aggregate throughput of the VPN gateway being created in the virtual hub to connect sites to. If you pick 1 scale unit = 500 Mbps, it implies that two instances for redundancy will be created, each having a maximum throughput of 500 Mbps.

There are two types of Virtual WANs: Basic and Standard. The following table shows the available configurations for each type.

|  |  |  |
| --- | --- | --- |
| **Virtual WAN type** | **Hub type** | **Available configurations** |
| Basic | Basic | Site-to-site VPN only |
| Standard | Standard | ExpressRoute  User VPN (P2S)  VPN (site-to-site)  Inter-hub and VNet-to-VNet transiting through the virtual hub |

The minimum address space is /24 (private) to create a hub.

You can use Virtual WAN to connect a VNet to a virtual hub in a different tenant. This architecture is useful if you have client workloads that must be connected to be the same network but are on different tenants.

Remote Tenant 
VM 
Parent Tenant 
VNet 
Virtual WAN 
VNet 
VM 

Before you can connect a cross-tenant VNet to a Virtual WAN hub, you must have the following configuration already set up:

* + A Virtual WAN and virtual hub in the parent subscription.
  + A virtual network configured in a subscription in the remote tenant.
  + Non-overlapping address spaces in the remote tenant and address spaces within any other VNets already connected to the parent virtual hub.

For the parent subscription with the virtual hub to modify and access the virtual networks in the remote tenant, you need to assign Contributor permissions to your parent subscription from the remote tenant subscription.

Add the remote tenant subscription and the parent tenant subscription to the current session of PowerShell by running the Add-AzAccount command. If you are signed into the parent, you only need to run the command for the remote tenant.

The routing capabilities in a virtual hub are provided by a router that manages all routing between gateways using Border Gateway Protocol (BGP).  This router also provides transit connectivity between virtual networks that connect to a virtual hub and can support up to an aggregate throughput of 50 Gbps.

Each connection is associated to one route table. Associating a connection to a route table allows the traffic to be sent to the destinations indicated as routes in the route table. The routing configuration of the connection will show the associated route table. Multiple connections can be associated to the same route table. All VPN, ExpressRoute, and User VPN connections are associated to the same (default) route table. Each virtual hub has its own Default route table, which can be edited to add a static route(s). Routes added statically take precedence over dynamically learned routes for the same prefixes.

Connections dynamically propagate routes to a route table. With a VPN connection, ExpressRoute connection, or P2S configuration connection, routes are propagated from the virtual hub to the on-premises router using BGP. Routes can be propagated to one or multiple route tables. A None route table is also available for each virtual hub. Propagating to the None route table implies that no routes are required to be propagated from the connection.

1 1,0 、 16 
10 一 、 、 16 
19 、 · 1E3-11 · 、 4 

Labels provide a mechanism to logically group route tables. This is especially helpful during propagation of routes from connections to multiple route tables. For example, the Default Route Table has a built-in label called 'Default'. When users propagate connection routes to 'Default' label, it automatically applies to all the Default Route Tables across every hub in the Virtual WAN.

Route tables now have features for association and propagation. A pre-existing route table is a route table that does not have these features. If you have pre-existing routes in hub routing and would like to use the new capabilities, consider the following:

* + Standard Virtual WAN Customers with pre-existing routes in virtual hub: If you have pre-existing routes in Routing section for the hub in Azure portal, you will need to first delete them and then attempt creating new route tables (available in the Route Tables section for the hub in Azure portal).
  + Basic Virtual WAN Customers with pre-existing routes in virtual hub: If you have pre-existing routes in Routing section for the hub in Azure portal, you will need to first delete them, then upgrade your Basic Virtual WAN to Standard Virtual WAN.

Virtual Hub Reset is available only in the Azure portal. Resetting provides you a way to bring any failed resources such as route tables, hub router, or the virtual hub resource itself back to its rightful provisioning state.

**Create a network virtual appliance (NVA) in a virtual hub**

NVAs are for connectivity to networking partners (e.g. Cisco, VMWare SDWAN, etc.) and are deployed directly into a Virtual WAN hub and have an externally facing public IP address. This enables customers who want to connect their branch Customer Premises Equipment (CPE) to the same brand NVA in the virtual hub to take advantage of proprietary end-to-end SD-WAN capabilities. The NVAs available in the Azure Marketplace are designed to be deployed directly into a virtual hub and nowhere else (i.e. cannot be deployed in a VNet)

When you create an NVA in the Virtual WAN hub, like all Managed Applications, there will be two Resource Groups created in your subscription;

* + **Customer resource group**: This will contain an application placeholder for the Managed Application. Partners can use this to expose whatever customer properties they choose here
  + **Managed resource group**: This Resource Group will contain the NetworkVirtualAppliances resource and is controlled by the publisher of the Managed Application

 An NVA Infrastructure Unit is a unit of aggregate bandwidth capacity for an NVA in the Virtual WAN hub. An NVA Infrastructure Unit is similar to a VPN Scale Unit in terms of the way you think about capacity and sizing. 1 NVA Infrastructure Unit represents 500 Mbps of aggregate bandwidth for all branch site connections coming into this NVA. Azure supports from 1-80 NVA Infrastructure Units for a given NVA virtual hub deployment.

[**Design and implement Azure ExpressRoute**](https://docs.microsoft.com/en-gb/learn/modules/design-implement-azure-expressroute/?ns-enrollment-type=LearningPath&ns-enrollment-id=learn.wwl.designing-implementing-microsoft-azure-networking-solutions-az-700)

**Explore Azure ExpressRoute**

ExpressRoute lets you extend your on-premises networks into the Microsoft cloud over a private connection with the help of a connectivity provider. You can create a connection between your on-premises network and the Microsoft cloud in four different ways, CloudExchange Co-location, Point-to-point Ethernet Connection, Any-to-any (IPVPN) Connection, and ExpressRoute Direct.

M icrosoft 
Azure 
Cloud exchange 
co-location 
Microsoft 
Azure 
Point-to-point Ethernet 
connection 
"Service provider model" 
Microsoft 
Azure 
WAN 
Any-to-any (IPVPN) 
connection 
Microsoft 
Azure 
ExpressRoute site 
"Direct model" 

ExpressRoute circuits can be anywhere from 50Mbps to 10Gbps. You can dynamically upgrade bandwidth but not downgrade it.

When provisioning an ExpressRoute, you must configure it on the Azure Portal first and then provide the "Service key" to your connectivity provider.

You can connect directly into the Microsoft's global network at a peering location strategically distributed across the world. ExpressRoute Direct provides dual 100 Gbps or 10-Gbps connectivity, which supports Active/Active connectivity at scale.

Key features that ExpressRoute Direct provides includes:

* + Massive Data Ingestion into services like Storage and Cosmos DB
  + Physical isolation for industries that are regulated and require dedicated and isolated connectivity like: Banking, Government, and Retail
  + Granular control of circuit distribution based on business unit

ExpressRoute using a Service 
Provider 
Uses service providers to enable fast 
onboarding and connectivity into 
exlstng infrastructure 
Integrates with hundreds of 
providers including Ethernet and 
MPLS 
Circuits SKUs from 50 Mbps to 10 
Gbps 
Optimized for single tenant 
ExpressRoute Direct 
Requires 100 Gbps/10 Gbps infrastructure and full management of all layers 
Direct/Dedicated capacity for regulated industries and massive data ingestion 
Customer may select a combination of the following circuit SKUs on IOO•Gbps 
ExpressRoute Direct: 5 Gbps 10 Gbps 40 Gbps 100 Gbps Customer may select a 
combination of the following circuit SKUs on 10-Gbps ExpressRoute Direct: 1 Gbps 2 Gbps 5 
Gbps 10 Gbps 
Optimized for single tenant with multiple business units and multiple work environments 

When Microsoft peering gets configured on your ExpressRoute circuit, the Microsoft Edge routers establish a pair of Border Gateway Protocol (BGP) sessions with your edge routers through your connectivity provider. No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter. A route filter can have only one rule, and the rule must be of type 'Allow'. This rule can have a list of BGP community values associated with it.

You can enable ExpressRoute circuit either by Layer 2 connections or managed Layer 3 connections. In both cases, if there are more than one Layer-2 devices in the ExpressRoute connection path, the responsibility of detecting any link failures in the path lies with the overlying BGP session.

You can control the BGP timers by configuring a lower BGP keep-alive and hold-time on your edge peering device. If the BGP timers are not the same between the two peering devices, the BGP session will establish using the lower time value. The BGP keep-alive can be set as low as three seconds, and the hold-time as low as 10 seconds. BFD provides low-overhead link failure detection in a sub second time interval. To enable BFD, you only need to configure BFD on both your primary and secondary devices. Configuring BFD is two-step process. You configure the BFD on the interface and then link it to the BGP session.

You can configure IPSec VPN over ExpressRoute:

Private Peering 
WAN 
VPN cm nection 
ExressR0LRe 

The connectivity establishment is straightforward:

* + Establish ExpressRoute connectivity with an ExpressRoute circuit and private peering.
  + Establish the VPN connectivity.

An important aspect of this configuration is routing between the on-premises networks and Azure over both the ExpressRoute and VPN paths. For traffic from on-premises networks to Azure, the Azure prefixes (including the virtual hub and all the spoke virtual networks connected to the hub) are advertised via both the ExpressRoute private peering BGP and the VPN BGP. This results in two network routes (paths) toward Azure from the on-premises networks:

* + One over the IPsec-protected path
  + One directly over ExpressRoute without IPsec protection

To overcome the above: Advertise more specific prefixes on the VPN BGP session for the VPN-connected network. You can advertise a larger range that encompasses the VPN-connected network over ExpressRoute private peering, then more specific ranges in the VPN BGP session. Advertise disjoint prefixes for VPN and ExpressRoute. If the VPN-connected network ranges are disjoint from other ExpressRoute connected networks, you can advertise the prefixes in the VPN and ExpressRoute BGP sessions, respectively.

Note: If you advertise the same prefixes over both ExpressRoute and VPN connections, Azure will use the ExpressRoute path directly without VPN protection.

There are 2 ways in which redundancy can be planned for an ExpressRoute deployment.

* + Configure ExpressRoute and site to site VPN coexisting connections
    - Only route-based VPN gateway is supported
    - The ASN of Azure VPN Gateway must be set to 65515
    - The gateway subnet must be /27 or a shorter prefix
    - Coexistence in a dual stack VNet is not supported
  + Create a zone redundant VNET gateway in Azure Availability zones
    - Zone redundant gateways: To automatically deploy your virtual network gateways across availability zones
    - Zonal (within the same zone) gateways: To deploy gateways in a specific zone. All instances of the gateway are deployed in the same Availability Zone.

Zone-redundant gateways and zonal gateways both rely on the Azure public IP resource Standard SKU. The configuration of the Azure public IP resource determines whether the gateway that you deploy is zone-redundant, or zonal. If you create a public IP resource with a Basic SKU, the gateway will not have any zone redundancy, and the gateway resources will be regional.

* + Zone redundant gateways: For a VPN gateway, the two gateway instances will be deployed in any 2 out of the three zones to provide zone-redundancy. For an ExpressRoute gateway, since there can be more than two instances, the gateway can span across all the three zones. Standard Public IP
  + Zonal gateways: All the gateway instances will be deployed in the same zone. Standard Public IP
  + Regional gateways: the gateway is deployed as a regional gateway and does not have any zone-redundancy built into the gateway.

You can configure a Site-to-Site VPN connection as a backup for ExpressRoute. This connection applies only to virtual networks linked to the Azure private peering path. There is no VPN-based failover solution for services accessible through Azure Microsoft peering.

The ExpressRoute circuit is always the primary link. Data flows through the Site-to-Site VPN path only if the ExpressRoute circuit fails. To avoid asymmetrical routing, your local network configuration should also prefer the ExpressRoute circuit over the Site-to-Site VPN. You can prefer the ExpressRoute path by setting higher local preference for the routes received the ExpressRoute.

You can enable ExpressRoute Premium to extend connectivity across geopolitical boundaries. The Standard SKU is region dependant.

You can transfer data cost-effectively by enabling the Local SKU. With Local SKU, you can bring your data to an ExpressRoute location near the Azure region you want. With Local, Data transfer is included in the ExpressRoute port charge.

You can enable ExpressRoute Global Reach (addon to premium) to exchange data across your on-premises sites by connecting your ExpressRoute circuits. For example, if you have a private data center in California connected to an ExpressRoute circuit in Silicon Valley and another private data center in Texas connected to an ExpressRoute circuit in Dallas. With ExpressRoute Global Reach, you can connect your private data centers together through these two ExpressRoute circuits. Your cross-data-center traffic will traverse through Microsoft's network.

**Design an ExpressRoute deployment**

ExpressRoute SKUs:

* + Local SKU - With Local SKU, you are automatically charged with an Unlimited data plan.
  + Standard and Premium SKU - You can select between a Metered or an Unlimited data plan. All ingress data are free of charge except when using the Global Reach add-on.

Note: The usage of the Microsoft Global Network is charged on top of the of the ExpressRoute Direct.

ExpressRoute billing:

* + Unlimited data. Billing is based on a monthly fee; all inbound and outbound data transfer is included free of charge.
  + Metered data. Billing is based on a monthly fee; all inbound data transfer is free of charge. Outbound data transfer is charged per GB of data transfer. Data transfer rates vary by region.
  + ExpressRoute premium add-on. ExpressRoute premium is an add-on to the ExpressRoute circuit providing the following:
    - Increased route limits from 4,000 to 10,000
    - Global connectivity for services (services deployed in any region)
    - Increased number of Vnet links per ExpresRoute circuit

Provider* O 
Equinix 
Peering location* @ 
Seattle 
Bandwidth * G) 
50Mbps 
Standard 
SKU* @ 
Premium 
Metered 
Billing model* O 
Unlimited 
Allow classic operations O 
@ NO 

**Configure an ExpressRoute gateway**

Note: You must register your subscription in the network service provider's environment

To connect your Azure virtual network and your on-premises network via ExpressRoute, you must create a virtual network gateway first. A virtual network gateway serves two purposes: to exchange IP routes between the networks and to route network traffic. The GatewayType setting specifies whether the gateway is used for ExpressRoute, or VPN traffic. Each virtual network can have only one virtual network gateway per gateway type.

* + **VPN** - To send encrypted traffic across the public Internet, you use the gateway type 'VPN'. This is also referred to as a VPN gateway. Site-to-Site, Point-to-Site, and VNet-to-VNet connections all use a VPN gateway.
  + **ExpressRoute** - To send network traffic on a private connection, you use the gateway type 'ExpressRoute'. This is also referred to as an ExpressRoute gateway and is the type of gateway used when configuring ExpressRoute.

**Configure peering for an ExpressRoute deployment**

You can configure private peering and Microsoft peering for an ExpressRoute circuit. Peerings can be configured in any order you choose. However, you must make sure that you complete the configuration of each peering one at a time. To configure peering(s), the ExpressRoute circuit must be in a provisioned and enabled state.

The recommended configuration is that private peering is connected directly to the core network, and the public and Microsoft peering links are connected to your DMZ. Each peering requires separate BGP sessions (one pair for each peering type). The BGP session pairs provide a highly available link. If you are connecting through layer 2 connectivity providers, you are responsible for configuring and managing routing.

Microsoft Cloud 
Customer's premises 
Internet edge 
DMZ / 
Core Network 

Max. # prefixes 
supported per peering 
IP address ranges 
supported 
AS Number requirements 
IP protocols supported 
Routing Interface IP 
addresses 
Routing Interface IP 
addresses 
MD5 Hash support 
Private Peering 
4000 by default, 10,000 with ExpressRoute 
premium 
Any valid IP address within your WAN. 
Private and public AS numbers. You must own the 
public AS number if you choose to use one. 
Ipv4, IPv6 (preview) 
RFC1918 and public IP addresses 
Microsoft Peering 
200 
Public IP addresses owned by you or your 
connectivity provider. 
Private and public AS numbers. However, you 
must prove ownership of public IP addresses. 
IPv4, IPv6 
public IP addresses registered to you in routing 
registries. 
Yes 

Route filters are a way to consume a subset of supported services through Microsoft peering. Microsoft 365 services such as Exchange Online, SharePoint Online, and Skype for Business, are accessible through the Microsoft peering. When Microsoft peering gets configured in an ExpressRoute circuit, all prefixes related to these services gets advertised through the BGP sessions that are established. A BGP community value is attached to every prefix to identify the service that is offered through the prefix.

There are two ways to reduce the size of your route table (services advertised by Microsoft);

* + Use BGP communities to filter unwanted routes
  + Define route filters and apply them to your ExpressRoute circuit. A route filter is a new resource that lets you select the list of services you plan to consume through Microsoft peering. ExpressRoute routers only send the list of prefixes that belong to the services identified in the route filter.

When Microsoft peering gets configured on your ExpressRoute circuit, the Microsoft Edge routers establish a pair of BGP sessions with your edge routers through your connectivity provider. No routes are advertised to your network. To enable route advertisements to your network, you must associate a route filter. The route filter resource must be in the same resource-group as the ExpressRoute circuit.

Note: You can reset the Microsoft peering and the Azure private peering on an ExpressRoute circuit independently.

**Connect an ExpressRoute circuit to a virtual network**

ExpressRoute circuits do not map to any physical entities. A circuit is uniquely identified by a standard GUID called as a service key (s-key).

You can link up to 10 virtual networks to a standard ExpressRoute circuit. All virtual networks must be in the same geopolitical region when using a standard ExpressRoute circuit. A single VNet can be linked to up to 16 ExpressRoute circuits. The ExpressRoute circuits can be in the same subscription, different subscriptions, or a mix of both.

Note: When you set up site-to-site VPN over Microsoft peering, you are charged for the VPN gateway and VPN egress.

VPN tunnels over Microsoft peering can be terminated either using VPN gateway or using an appropriate Network Virtual Appliance (NVA) available through Azure Marketplace. You can exchange routes statically or dynamically over the encrypted tunnels without exposing the route exchange to the underlying Microsoft peering.

**Connect geographically dispersed networks with ExpressRoute global reach**

You can enable ExpressRoute Global Reach between the private peering of any two ExpressRoute circuits, if they are in the supported countries/regions. The circuits are required to be created at different peering locations. If the two circuits are in different Azure subscriptions, you need authorization from one Azure subscription. Then you pass in the authorization key when you run the configuration command in the other Azure subscription.

**Improve data path performance between networks with ExpressRoute FastPath**

FastPath is designed to improve the data path performance between your on-premises network and your virtual network. When enabled, FastPath sends network traffic directly to virtual machines in the virtual network, bypassing the gateway. FastPath still requires a virtual network gateway to be created to exchange routes between virtual network and on-premises network.

To configure FastPath, the virtual network gateway must be either:

* + Ultra-Performance
  + ErGw3AZ

Note: If you plan to use FastPath with IPv6-based private peering over ExpressRoute, make sure to select ErGw3AZ for SKU. Note that this is only available for circuits using ExpressRoute Direct.

While FastPath supports most configurations, it does not support the following features:

* + UDR on the gateway subnet: This UDR has no impact on the network traffic that FastPath sends directly from your on-premises network to the virtual machines in Azure virtual network.
  + VNet Peering: If you have other virtual networks peered with the one that is connected to ExpressRoute, the network traffic from your on-premises network to the other virtual networks (i.e., the so-called "Spoke" VNets) will continue to be sent to the virtual network gateway. The workaround is to connect all the virtual networks to the ExpressRoute circuit directly.
  + Basic Load Balancer: If you deploy a Basic internal load balancer in your virtual network or the Azure PaaS service you deploy in your virtual network uses a Basic internal load balancer, the network traffic from your on-premises network to the virtual IPs hosted on the Basic load balancer will be sent to the virtual network gateway. The solution is to upgrade the Basic load balancer to a Standard load balancer.
  + Private Link: If you connect to a private endpoint in your virtual network from your on-premises network, the connection will go through the virtual network gateway.

The 'circuit owner' is an authorized Power User of the ExpressRoute circuit resource. The circuit owner can create authorizations that can be redeemed by 'circuit users'. Circuit users are owners of virtual network gateways that are not within the same subscription as the ExpressRoute circuit. Circuit users can redeem authorizations (one authorization per virtual network).

The circuit owner creates an authorization, which creates an authorization key to be used by a circuit user to connect their virtual network gateways to the ExpressRoute circuit. An authorization is valid for only one connection. Individual Resource ID and the Authorization Key are required for each circuit.

The circuit user needs the resource ID and an authorization key from the circuit owner. To redeem a connection authorization they need to create a Connection resource

**Troubleshoot ExpressRoute connection issues**

If enabling a peering fails, check if the primary and secondary subnets assigned match the configuration on the linked CE/PE-MSEE. Also check if the correct VlanId, AzureASN, and PeerASN are used on MSEEs and if these values map to the ones used on the linked CE/PE-MSEE. If MD5 hashing is chosen, the shared key should be same on MSEE and PE-MSEE/CE pair.

On a /30 subnet assigned for interface, Microsoft will pick the second usable IP address of the subnet for the MSEE interface. Therefore, ensure that the first usable IP address of the subnet has been assigned on the peered CE/PE-MSEE.

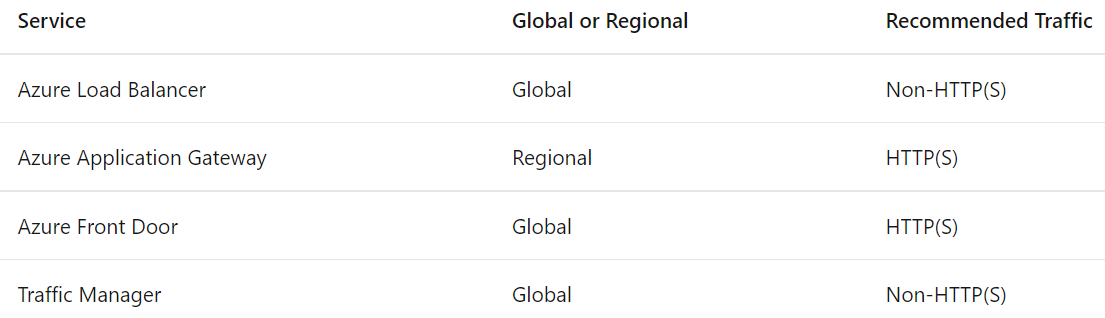
ExpressRoute uses Network insights to provide a detailed topology mapping of all ExpressRoute components (peerings, connections, gateways) in relation with one another. Network insights for ExpressRoute also have preloaded metrics dashboard for availability, throughput, packet drops, and gateway metrics. You can analyze metrics for Azure ExpressRoute with metrics from other Azure services using metrics explorer by opening Metrics from the Azure Monitor menu.

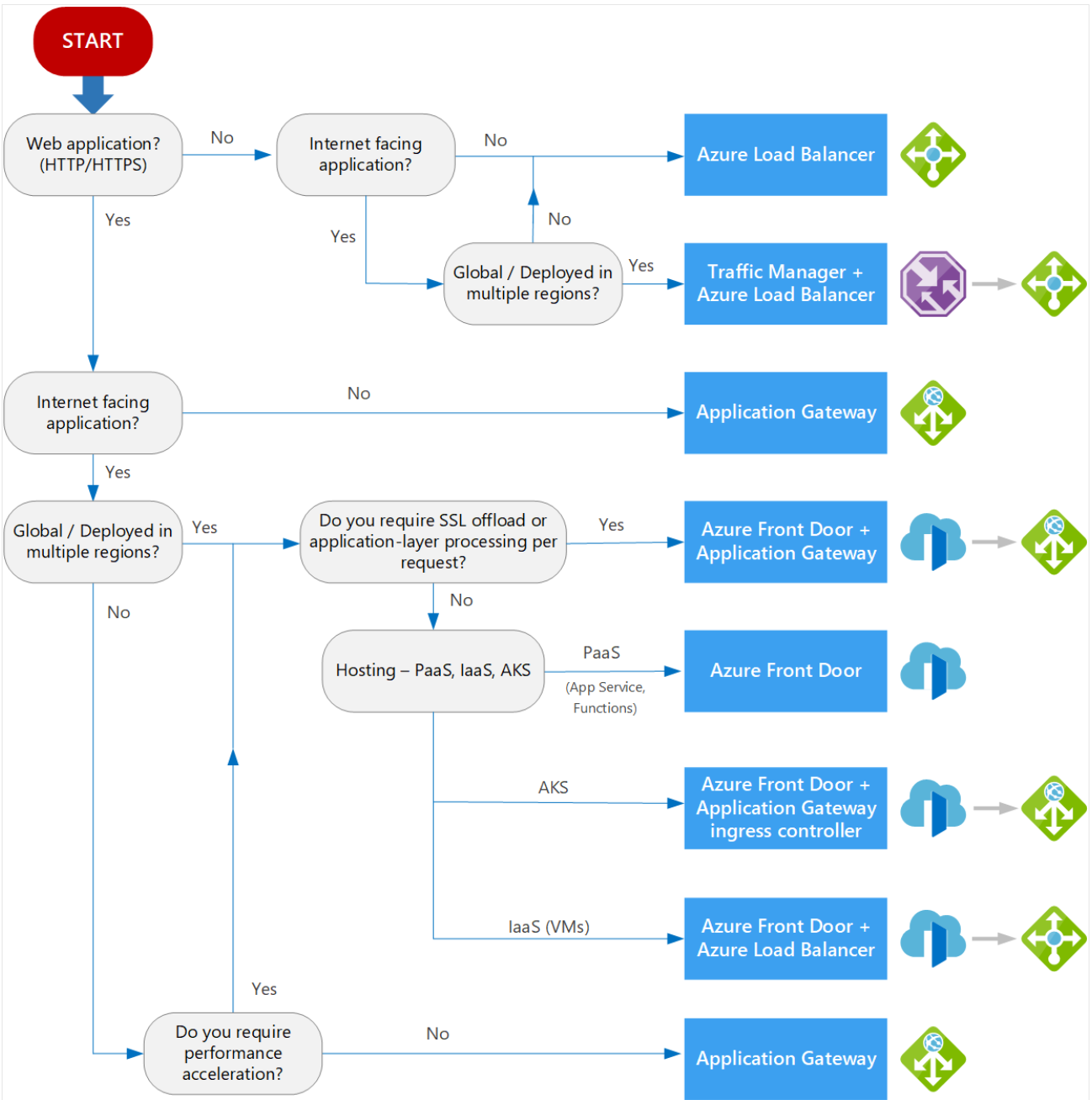
**Load balance non-HTTP(S) traffic in Azure**

**Explore load balancing**

Load balancing options for Azure:

* + **Azure Load balancer**: layer 4 load balancer, zone redundant
  + **Traffic Manager**: DNS based, cannot failover as fast as Front Door.
  + **Azure Application Gateway**: Application Delivery Control services. Offering layer 7 load balancing
  + **Azure Front Door**: global load balancing and site acceleration service for web applications. Layer 7 capabilities for your application like SSL offload, path-based routing, fast failover, caching, etc.



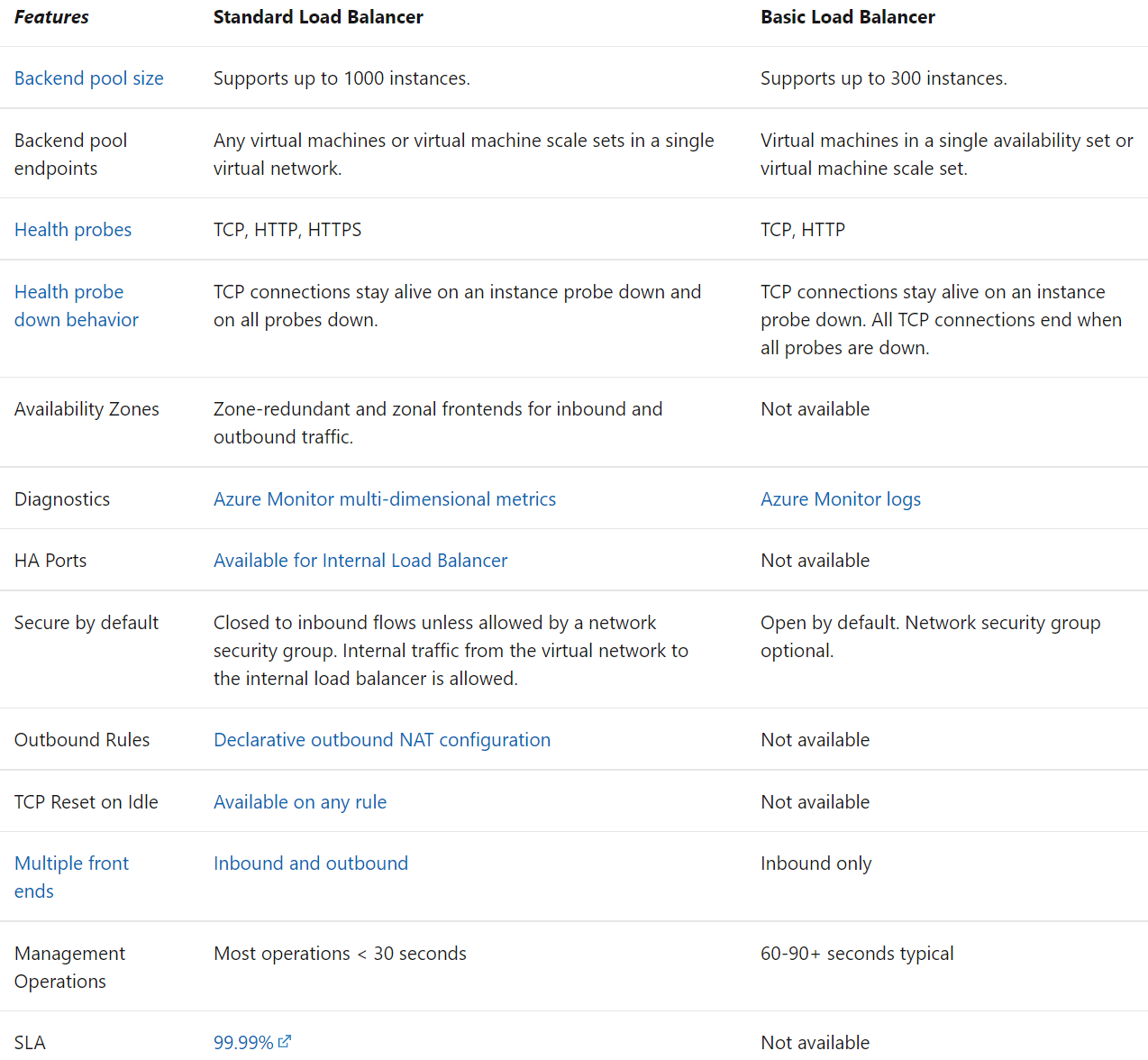


**Design and implement Azure load balancer using the Azure portal**

A public load balancer can provide outbound connections for virtual machines (VMs) inside your virtual network. These connections are accomplished by translating their private IP addresses to public IP addresses. An internal load balancer is used where private IPs are needed at the frontend only. Internal load balancers are used to load balance traffic from internal Azure resources to other Azure resources inside a virtual network.

A Load Balancer can either be zone redundant, zonal, or non-zonal.

Two SKUs are available when you create a load balancer in Azure: Basic load balancers and Standard load balancers.



Standalone VMs, availability sets, and virtual machine scale sets can be connected to only one SKU, never both. Load balancer and the public IP address SKU must match when you use them with public IP addresses.

Interval: Specify the interval time in seconds between probe attempts. The default is 5 seconds

Unhealthy threshold: Specify the number of consecutive probe failures that must occur before a virtual machine is considered to be in an unhealthy state. The default is 2 failures.

Session persistence: You can choose None, or Client IP, or Client IP and protocol. Session persistence specifies that traffic from a client should be handled by the same virtual machine in the backend pool for the duration of a session. None specifies that successive requests from the same client may be handled by any virtual machine. Client IP specifies that successive requests from the same client IP address will be handled by the same virtual machine. Client IP and protocol specifies that successive requests from the same client IP address and protocol combination will be handled by the same virtual machine.

Floating IP: Choose between Disabled or Enabled. With Floating IP set to Disabled, Azure exposes a traditional load balancing IP address mapping scheme for ease of use (the VM instances' IP). With Floating IP set to Enabled, it changes the IP address mapping to the Frontend IP of the load balancer to allow for additional flexibility.

**Explore Azure Traffic Manager**

Traffic Manager uses DNS to direct the client requests to the appropriate service endpoint based on a traffic-routing method. Traffic manager also provides health monitoring for every endpoint. The endpoint can be any Internet-facing service hosted inside or outside of Azure.

Traffic Manager supports external, non-Azure endpoints enabling it to be used with hybrid cloud and on-premises deployments, including the burst-to-cloud, migrate-to-cloud, and failover-to-cloud scenarios.

Traffic Manager provides two key benefits:

* + Distribution of traffic according to one of several traffic-routing methods
  + Continuous monitoring of endpoint health and automatic failover when endpoints fail

Azure Traffic Manager supports six traffic-routing methods to determine how to route network traffic to the various service endpoints.

Routing 
method 
Priority 
Weighted 
Performance 
Geographic 
MultiValue 
Subnet 
When to use 
Select this routing method when you want to have a primary service endpoint for all traffic. You can provide 
multiple backup endpoints in case the primary or one of the backup endpoints is unavailable. 
Select this routing method when you want to distribute traffic across a set of endpoints based on their weight. Set 
the weight the same to distribute evenly across all endpoints. 
Select the routing method when you have endpoints in different geographic locations, and you want end users to 
use the "closest" endpoint for the lowest network latency. 
Select this routing method to direct users to specific endpoints (Azure, External, or Nested) based on where their 
DNS queries originate from geographically, With this routing method, it enables you to be compliant with 
scenarios such as data sovereignty mandates, localization of content & user experience and measuring traffic from 
different regions. 
Select this routing method for Traffic Manager profiles that can only have IPv4/IPv6 addresses as endpoints. When 
a query is received for this profile, all healthy endpoints are returned. 
Select this routing method to map sets of end-user IP address ranges to a specific endpoint. When a request is 
received, the endpoint returned will be the one mapped for that request's source IP address. 

**Priority**: With Azure Resource Manager, you configure the endpoint priority explicitly using the 'priority' property for each endpoint. This property is a value between 1 and 1000. A lower value represents a higher priority. Endpoints can't share priority values. Setting the property is optional. When omitted, a default priority based on the endpoint order is used.

**Weighted**: The weight is an integer from 1 to 1000. This parameter is optional. If omitted, Traffic Managers uses a default weight of '1'. The higher weight, the higher the priority.

Within a Traffic Manager profile, you can only configure one traffic routing method at a time. You can select a different traffic routing method for your profile at any time. Your changes will be applied within a minute without any downtime. All Traffic Manager profiles have health monitoring and automatic failover of endpoints. You can combine traffic routing methods by using nested Traffic Manager profiles to gain the benefits of multiple traffic-routing methods. Nested profiles enable you to override the default Traffic Manager behavior to support larger and more complex traffic-routing configurations for your application deployments.

Traffic Manager supports three types of endpoints:

* + **Azure endpoints** - Use this type of endpoint to load-balance traffic to a cloud service, web app, or public IP address in the same subscription within Azure.
  + **External endpoints** - Use this type of endpoint to load balance traffic for IPv4/IPv6 addresses, FQDNs, or for services hosted outside Azure. These services can either be on-premises or with a different hosting provider.
  + **Nested endpoints** - Use this type of endpoint to combine Traffic Manager profiles to create more flexible traffic-routing schemes to support the needs of larger, more complex deployments. With Nested endpoints, a child profile is added as an endpoint to a parent profile. Both the child and parent profiles can contain other endpoints of any type, including other nested profiles.

Azure Traffic Manager includes built-in endpoint monitoring and automatic endpoint failover. This feature helps you deliver high-availability applications that are resilient to endpoint failure, including Azure region failures.

Traffic Manager supports HTTP, HTTPS or TCP for probing your endpoint to check its health

The custom headers can be specified at a profile level to be applicable for all endpoints in that profile and / or at an endpoint level applicable only to that endpoint. You can use custom headers for health checks of endpoints in a multi-tenant environment. That way it can be routed correctly to their destination by specifying a host header.

Traffic Manager supports two probing intervals: 30 seconds (default), 10 seconds. Tolerated number of failures can be from 0 to 9.

Probe timeout: If the Probing Interval is set to 30 seconds, then you can set the Timeout value between 5 and 10 seconds. If no value is specified, it uses a default value of 10 seconds. If the Probing Interval is set to 10 seconds, then you can set the Timeout value between 5 and 9 seconds. If no Timeout value is specified, it uses a default value of 9 seconds.

When the monitoring protocol is set as HTTP or HTTPS, the Traffic Manager probing agent makes a GET request to the endpoint using the protocol, port, and relative path given. An endpoint is considered healthy if probing agent receives a 200-OK response, or any of the responses configured in the Expected status code \*ranges.

For HTTP or HTTPS monitoring protocol, a common practice on the endpoint side is to implement a custom page within your application - for example, /health.aspx. Using this path for monitoring, you can perform application-specific checks, such as checking performance counters or verifying database availability. Based on these custom checks, the page returns an appropriate HTTP status code.

All endpoints in a Traffic Manager profile share monitoring settings. If you need to use different monitoring settings for different endpoints, you can create nested Traffic Manager profiles.

**Load balance HTTP(S) traffic in Azure**

**Design Azure Application Gateway**

Application Gateway can make routing decisions based on additional attributes of an HTTP request, for example URI path or host headers. Application Gateway supports:

* + A web application firewall to protect against web application vulnerabilities.
  + End-to-end request encryption.
  + Autoscaling, to dynamically adjust capacity as your web traffic load change.
  + Redirection: Redirection can be used to another site, or from HTTP to HTTPS
  + Rewrite HTTP headers: HTTP headers allow the client and server to pass parameter information with the request or the response.
  + Custom error pages

There are two primary methods of routing traffic, path-based routing, and multiple site routing.

**Path based**: sends requests with different URL paths to different pools of back-end servers. For example, you could direct requests with the path /video/\* to a back-end pool containing servers that are optimized to handle video streaming, and direct /images/\* requests to a pool of servers that handle image retrieval.

**Multiple site routing**: configures more than one web application on the same application gateway instance. you register multiple DNS names (CNAMEs) for the IP address of the Application Gateway, specifying the name of each site. Application Gateway uses separate listeners to wait for requests for each site. Each listener passes the request to a different rule, which can route the requests to servers in a different back-end pool.

Contoso 
Pool 
Application 
Gateway 
WAF 
L7 LB 
contoso.com 
Fabrikam 
Pool 
fabrikam.com 

Multi-site configurations are useful for supporting multi-tenant applications, where each tenant has its own set of virtual machines or other resources hosting a web application

Application Gateway is available under a Standard\_v2 SKU. Web Application Firewall (WAF) is available under a WAF\_v2 SKU. The v2 SKU offers performance enhancements and adds support for critical new features like autoscaling, zone redundancy, support for static VIPs, Header Rewrite, Key Vault integration, Kubernetes Service Ingress Controller, performance enhancements and faster deployment and update time .

**Autoscaling**: Application Gateway or WAF deployments under the autoscaling SKU can scale out or in based on changing traffic load patterns. Autoscaling also removes the requirement to choose a deployment size or instance count during provisioning. In the Standard\_v2 and WAF\_v2 SKU, Application Gateway can operate both in fixed capacity (autoscaling disabled) and in autoscaling enabled mode. Each instance is roughly equivalent to 10 additional reserved Capacity Units. Zero signifies no reserved capacity and is purely autoscaling in nature. The instance counts can range from 0 to 125. The default value for maximum instance count is 20 if not specified.

Application Gateway v2 SKU supports the static VIP type exclusively. In v1 you must use the application gateway URL instead of the IP address for domain name routing to App Services via the application gateway.

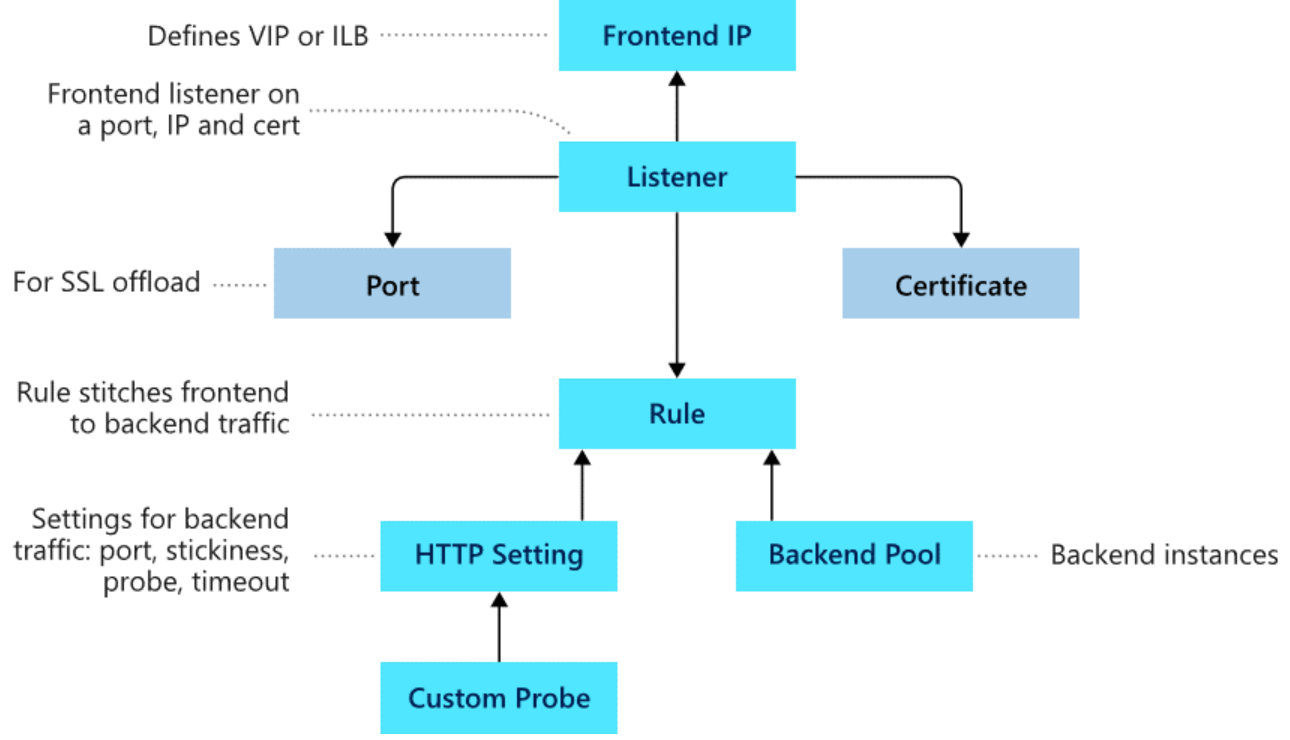
Application Gateway v2 supports integration with Key Vault for server certificates that are attached to HTTPS enabled listeners.

The v2 SKU offers up to 5X better TLS offload performance as compared to the Standard/WAF SKU.

With the v2 SKU, the pricing model is driven by consumption and is no longer attached to instance counts or sizes. The v2 SKU pricing has two components:

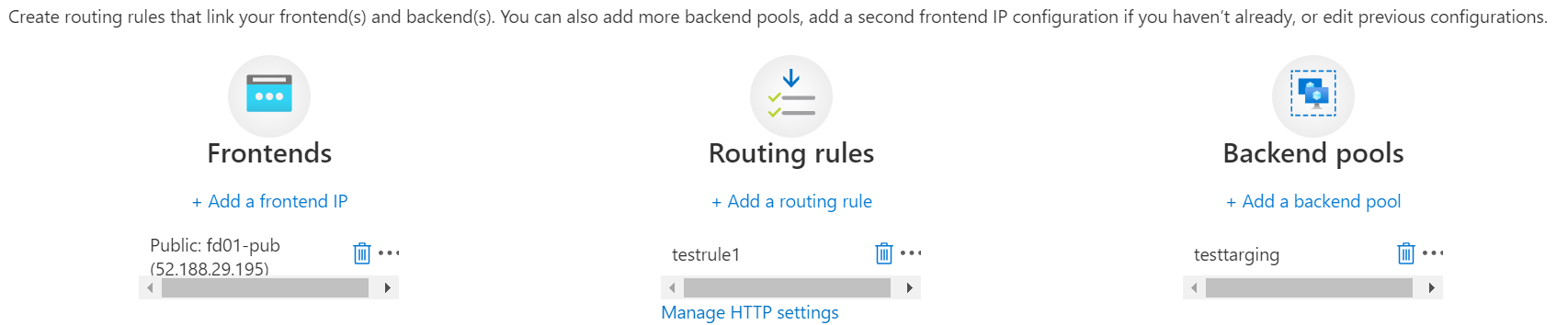
* + Fixed price: This is hourly (or partial hour) price to provision a Standard\_v2 or WAF\_v2 Gateway. Please note that 0 additional minimum instances still ensure high availability of the service which is always included with fixed price.
  + Capacity Unit price: This is a consumption-based cost that is charged in addition to the fixed cost. Capacity unit charge is also computed hourly or partial hourly. There are three dimensions to capacity unit - compute unit, persistent connections, and throughput.

**Configure Azure Application Gateway**



**Configuration Steps:** Create Application Gateway

* + Basics: Configure subscription, RG, instance name, autoscaling, VNet
  + Frontends: Public, Private or both IP address
  + Backends: Add a backend pool - Pool name, add with or without targets, target type (IP address or FQDN, VM, VMSS, App Service)
  + Configuration: per diagram below. Frontends is the one(s) configured in Step 2
    - Routing rule: rule name, Listener name, listener frontend IP/protocol/port, Listener type (basic or multi-site - keep in mind order of processing)
    - Backend targets: backend pool OR redirection, for POOL configure backend target and HTTP settings, for Redirection choose type choose target (listener or external site) and choose Include query string and Include Path (add multiple targets to create a path-based rule)
    - Backend Pool



For the Application Gateway v2 SKU, there must be a public frontend IP configuration. You can still have both a Public and a Private frontend IP configuration, but Private only frontend IP configuration (Only ILB mode) is currently not enabled for the v2 SKU.

The backend pool is used to route requests to the backend servers that serve the request. Backend pools can be composed of NICs, virtual machine scale sets, public IP addresses, internal IP addresses, fully qualified domain names (FQDN), and multi-tenant back-ends like Azure App Service.

Azure Application Gateway by default monitors the health of all resources in its back-end pool and automatically removes any resource considered unhealthy from the pool. By default, Application gateway sends the health probes with the same port that is defined in the back-end HTTP/HTTPS settings. A custom probe port can be configured using a custom health probe.

The source IP address Application Gateway uses for health probes depends on the backend pool:

* + If the server address in the backend pool is a public endpoint, then the source address is the application gateway's frontend public IP address.
  + If the server address in the backend pool is a private endpoint, then the source IP address is from the application gateway subnet's private IP address space.

Probe settings are: Interval 30 second, Time-out 30 seconds, Unhealthy threshold 3 seconds.

Note: Backend types supported include: IP address/FQDN, VM, VMSS, App Service

When you create an Application Gateway, you must configure a Routing Rule which consists of a rule name, Listener and Backend Targets.

When you create an application gateway you also create a default listener by choosing the protocol and port for the listener. You can choose whether to enable HTTP2 support on the listener. A listener is a logical entity that checks for incoming connection requests by using the port, protocol, host, and IP address. When you configure a listener, you must enter values that match the corresponding values in the incoming request on the gateway. You must choose between basic and multi-site.

* + Basic: All requests for any domain will be accepted and forwarded to backend pools.
  + Multi-site: Forward requests to different backend pools based on the host header or host names. You must specify a host name that matches with the incoming request. This is because Application Gateway relies on HTTP 1.1 host headers to host more than one website on the same public IP address and port.

To configure TLS termination and end-to-end TLS encryption, you must add a certificate to the listener to enable the application gateway to derive a symmetric key.

You can use application gateway to redirect traffic (Step 4 in configuration). It has a generic redirection mechanism which allows for redirecting traffic received at one listener to another listener or to an external site. This simplifies application configuration, optimizes the resource usage, and supports new redirection scenarios including global and path-based redirection.

The following types of redirection are supported:

* + 301 Permanent Redirect
  + 302 Found
  + 303 See Other
  + 307 Temporary Redirect

If redirection is configured for a basic rule, all requests on the associated listener are redirected to the target. This is global redirection. If redirection is configured for a path-based rule, only requests in a specific site area are redirected. An example is a shopping cart area that's denoted by /cart/\*. This is path-based redirection.

Application Gateway redirection support offers the following capabilities:

* + Global redirection: Redirects from one listener to another listener on the gateway. This enables HTTP to HTTPS redirection on a site.
  + Path-based redirection: Enables HTTP to HTTPS redirection only on a specific site area, for example a shopping cart area denoted by /cart/\*.
  + Redirect to external site: Requires a new redirect configuration object, which specifies the target listener or external site to which redirection is desired. The configuration element also supports options to enable appending the URI path and query string to the redirected URL. The redirect configuration is attached to the source listener via a new rule.

Request Routing Rule: When you create an application gateway using the Azure portal, you create a default rule (rule1). This rule binds the default listener (appGatewayHttpListener) with the default back-end pool (appGatewayBackendPool) and the default back-end HTTP settings (appGatewayBackendHttpSettings). After you create the gateway, you can edit the settings of the default rule or create new rules. For the v1 and v2 SKU, pattern matching of incoming requests is processed in the order that the paths are listed in the URL path map of the path-based rule.

Rule types:

* + Basic forwards all requests on the associated listener (for example, blog.contoso.com/\*) to a single back-end pool.
  + Path-based routes requests from specific URL paths to specific back-end pools.

Note: for a basic-rule only one backend pool is allowed. For a path-based rule, multiple backend pools that correspond to each URL path are allowed. Requests that don't match any URL path can be sent to a default backend pool.

Add a back-end HTTP setting for each rule. Requests are routed from the application gateway to the back-end targets by using the port number, protocol, and other information that's specified in this setting. For a basic rule, only one back-end HTTP setting is allowed. For a path-based rule, add multiple back-end HTTP settings that correspond to each URL path. Much like the request routing rules above, you can have a default HTTP setting for requests that do not match any URL path.

By using rewrite rules, you can add, remove, or update HTTP(S) request and response headers as well as URL path and query string parameters as the request and response packets move between the client and backend pools via the application gateway. The headers and URL parameters can be set to static values or to other headers and server variables. This helps with important use cases, such as extracting client IP addresses, removing sensitive information about the backend, adding more security, and so on.

URL-based routing: RequestRoutingRule of type PathBasedRouting is used to bind a listener to a urlPathMap. All requests that are received for this listener are routed based on policy specified in urlPathMap.

Application Gateway allows you to rewrite selected content of requests and responses. With this feature, you can translate URLs, query string parameters as well as modify request and response headers. It also allows you to add conditions to ensure that the URL or the specified headers are rewritten only when certain conditions are met. These conditions are based on the request and response information. HTTP header and URL rewrite features are only available for the Application Gateway v2 SKU. The rewrite types supported are:

* + **Request and response headers**: accomplish important tasks, such as adding security-related header fields like HSTS/ X-XSS-Protection, removing response header fields that might reveal sensitive information, and removing port information from X-Forwarded-For headers.
  + **URL path and query string**: Rewrite the host name, path, and query string of the request URL

To configure a rewrite rule, you need to create a rewrite rule set and add the rewrite rule configuration in it. A rewrite rule set contains:

* + Request routing rule association
  + Rewrite condition
  + Rewrite type
    - Rewriting request headers
    - Rewriting response headers
    - Rewriting URL components (URL path, URL Query String, Re-evaluate path map)

**Design and configure Azure Front Door**

Azure Front Door is a global, scalable entry-point that uses the Microsoft global edge network to create fast, secure, and widely scalable web applications by improving global connectivity. Front Door works at Layer 7 (HTTP/HTTPS layer) using anycast protocol with split TCP.

Edge Location 
Region 1 
Region 2 
Region 3 
Microsoft Global Network 

Based on your routing method you can ensure that Front Door will route your client requests to the fastest and most available application backend. An application backend is any Internet-facing service hosted inside or outside of Azure. Front Door provides a range of traffic-routing methods and backend health monitoring options to suit different application needs and automatic failover scenarios. Like Traffic Manager, Front Door is resilient to failures, including failures to an entire Azure region.

Azure Front Door redirects traffic at each of the following levels: protocol, hostname, path, query string. These functionalities can be configured for individual microservices since the redirection is path-based. This can simplify application configuration by optimizing resource usage and supports new redirection scenarios including global and path-based redirection.

While both Front Door and Application Gateway are layer 7 (HTTP/HTTPS) load balancers, the primary difference is that Front Door is a global service whereas Application Gateway is a regional service. Front Door can load balance between your different scale units/clusters/stamp units across regions, and Application Gateway allows you to load balance between your VMs/containers that is within the scale unit.

A Front Door routing rule configuration is composed of two major parts: a "left-hand side" and a "right-hand side". Front Door matches the incoming request to the left-hand side of the route. The right-hand side defines how Front Door processes the request.

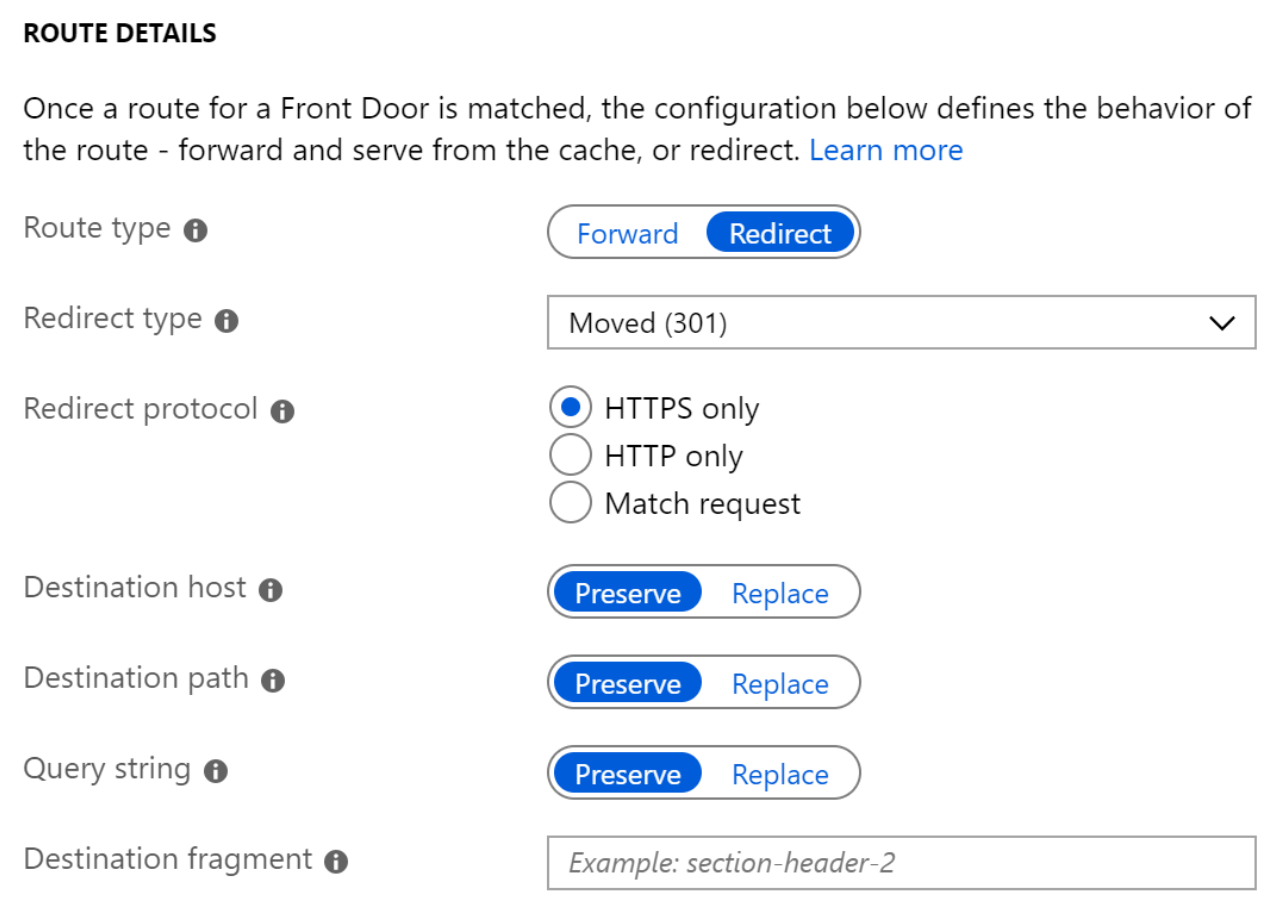
The following properties determine whether the incoming request matches the routing rule (or left-hand side):

* + HTTP Protocols (HTTP/HTTPS)
  + Hosts (for example, [www.foo.com](http://www.foo.com), \*.bar.com)
  + Paths (for example, /, /users/, /file.gif)

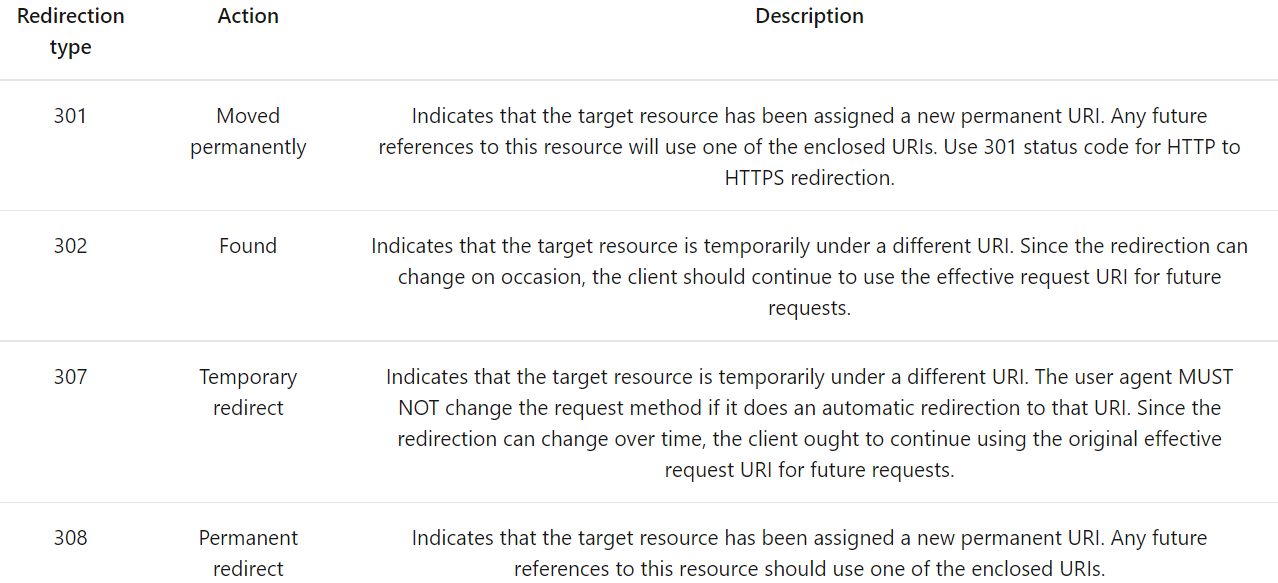
Front Door speeds up the processing of requests by using caching. If caching is enabled for a specific route, it uses the cached response. Front Door attempts to match to the most-specific match first looking only at the left-hand side of the route. It first matches based on HTTP protocol, then Frontend host, then the Path.

* + Frontend host matching: look for any routing rule with an exact match on the host. If not found, reject the request (sends 400 Bad Request error)
  + Path matching: look for any routing rule with an exact match on the path, if not found look for wildcard path that matches, if not found reject request (400 Bad request error)

Azure Front Door redirects traffic at each of the following levels: protocol, hostname, path, query string.



A redirect type sets the response status code for the clients to understand the purpose of the redirect. The following types of redirection are supported:



Note: Instead of 303 as per Application Gateway, we have 308.

You can set the protocol that will be used for redirection. The most common use case of the redirect feature is to set HTTP to HTTPS redirection. The three options are:

* + HTTPS only: recommended. Redirects incoming HTTP to HTTPS
  + HTTP only
  + Match request

"Destination host": As part of configuring a redirect routing, you can also change the hostname or domain for the redirect request. For cases where you want to replace the path segment of a URL as part of redirection, you can set "Destination Path" with the new path value. "Query string parameters" replace the query string parameters in the redirected URL. To replace any existing query string from the incoming request URL, set this field to 'Replace' and then set the appropriate value. e.g. redirect all traffic sent to <https://www.contoso.com/foo/bar> to <https://www.contoso.com/foo/bar?&utm_referrer=https%3A%2F%2Fwww.bing.com%2F>

The "destination fragment" is the portion of URL after '#', which is used by the browser to land on a specific section of a web page. You can set this field to add a fragment to the redirect URL.

Azure Front Door supports URL rewrite by configuring an optional Custom Forwarding Path to use when constructing the request to forward to the backend. By default, if a custom forwarding path isn't provided, the Front Door will copy the incoming URL path to the URL used in the forwarded request. The powerful part of URL rewrite is that the custom forwarding path will copy any part of the incoming path that matches to a wildcard path to the forwarded path (these path segments are the green segments in the example below):

Match Path: 
Custom Forwarding Path: 
Incoming URL Path: 
Forwarded Path: 
/foo/* 
/fwd/ 
/foo/a/b/c 
/fwd/a/b/c 

To determine the health and proximity of each backend for a given Front Door environment, each Front Door environment periodically sends a synthetic HTTP/HTTPS request to each of your configured backends.

Since Front Door has many edge environments globally, health probe volume for your backends can be quite high - ranging from 25 requests every minute to as high as 1200 requests per minute, depending on the health probe frequency configured. With the default probe frequency of 30 seconds, the probe volume on your backend should be about 200 requests per minute.

Supported HTTP methods for health probes: GET and HEAD. With HEAD the server MUST NOT return a message-body in the response. Because it has lower load and cost on your backends, for new Front Door profiles, by default, the probe method is set as HEAD.

**Design and implement network security**

**Secure your virtual networks in the Azure portal**

The Azure Security Benchmark (ASB) provides prescriptive best practices and recommendations to help improve the security of workloads, data, and services on Azure.

* + Security controls: These recommendations are generally applicable across your Azure tenant and Azure services. Each recommendation identifies a list of stakeholders that are typically involved in planning, approval, or implementation of the benchmark.
  + Service baselines: These apply the controls to individual Azure services to provide recommendations on that service’s security configuration.

Azure Security Center helps streamline the process for meeting regulatory compliance requirements, using the regulatory compliance dashboard. Azure Security Center is a unified infrastructure security management system that strengthens the security posture of your data centers and provides advanced threat protection across your hybrid workloads in the cloud as well as on premises.

Security Center automatically collects, analyzes, and integrates log data from your Azure resources, the network, and connected partner solutions - like firewall and endpoint protection solutions - to detect real threats and reduce false positives. A list of prioritized security alerts is shown in Security Center along with the information you need to quickly investigate the problem and steps to take to remediate an attack.

The left pane of the security alert page shows high-level information regarding the security alert: title, severity, status, activity time, description of the suspicious activity, and the affected resource. Alongside the affected resource are the Azure tags relevant to the resource. Use these to infer the organizational context of the resource when investigating the alert.

The right pane includes the Alert details tab containing further details of the alert to help you investigate the issue: IP addresses, files, processes, and more. Also in the right pane is the Take action tab. Use this tab to take further actions regarding the security alert. Actions such as:

* + Mitigate the threat: Provides manual remediation steps for this security alert
  + Prevent future attacks: Provides security recommendations to help reduce the attack surface, increase security posture, and thus prevent future attacks
  + Trigger automated response: Provides the option to trigger a logic app as a response to this security alert
  + Suppress similar alerts: Provides the option to suppress future alerts with similar characteristics if the alert isn’t relevant for your organization

Azure Security Center Adaptive Network Hardening is an agentless feature of Azure Security Center - nothing needs to be installed on your machines to benefit from this network hardening tool. It provides guidance about recommended network security group configurations like limiting ports and source IPs with reference to external network traffic rules.

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 in your virtual network.

Machine generated alternative text:



**Deploy Azure DDoS Protection by using the Azure portal**

Azure DDoS protection provides the following service tiers:

* + **Basic**: Automatically enabled as part of the Azure platform. Always-on traffic monitoring, and real-time mitigation of common network-level attacks, provide the same defenses utilized by Microsoft's online services. Across IPv4 and IPv6
  + **Standard**: Protection policies are tuned through dedicated traffic monitoring and machine learning algorithms. Policies are applied to public IP addresses associated to resources deployed in virtual networks. Does not apply to App Service Environments. Application layer protection can be added through the Azure Application Gateway Web Application Firewall or by installing a 3rd party firewall from Azure Marketplace.

|  |  |  |
| --- | --- | --- |
| DDoS attack | Attack description | DDoS Standard mitigation |
| Volumetric attacks | Flood network layer with substantial amount of seemingly legitimate traffic (UDP floods, amplification floods, etc) | Mitigation through absorbing and scrubbing traffic |
| Protocol attacks | Exploit weakness in the layer 3 and 4 protocol stack. e.g. SYN flood, reflection attacks, etc. | Mitigation through differentiating between malicious and legitimate traffic, by interacting with the client, and blocking malicious traffic |
| Resource (application) layer attacks | target web application packets, to disrupt the transmission of data between hosts. They include HTTP protocol violations, SQL injection, cross-site scripting, and other layer 7 attacks | Use a WAF as well as DDoS Protection Standard |

ithin a few minutes of attack detection, you are notified using Azure Monitor metrics. By configuring logging on DDoS Protection Standard telemetry, you can write the logs to available options for future analysis. Metric data in Azure Monitor for DDoS Protection Standard is retained for 30 days.

DDoS Standard Adaptive learning provides for automatic learning of per-customer (per- Public IP) traffic patterns for Layer 3 and 4.

You can integrate logging with Splunk (Azure Event Hubs), Azure Monitor logs, and Azure Storage for advanced analysis via the Azure Monitor Diagnostics interface. DDoS Protection Standard applies three autotuned mitigation policies (SYN, TCP, and UDP) for each public IP of the protected resource, in the virtual network that has DDoS enabled. You can view the policy thresholds by selecting the Inbound [SYN/TCP/UDP] packets to trigger DDoS mitigation metrics. The policy thresholds are autoconfigured via machine learning-based network traffic profiling. DDoS mitigation occurs for an IP address under attack only when the policy threshold is exceeded.

If the public IP address is under attack, the value for the Under DDoS attack or not metric changes to 1 as DDoS Protection performs mitigation on the attack traffic. It is recommended to configure an alert on this metric as you will then get notified if there is an active DDoS mitigation performed on your public IP address.

To configure DDoS you must first create a DDoS Protection Plan resource. You then assign the DDoS Plan to a Vnet through the Vnet's configuration page.

**Deploy Network Security Groups by using the Azure portal**

A network security group contains zero, or as many rules as desired, within Azure subscription limits. Each rule specifies the following properties:

* + Name
  + Priority: from 100 to 4096 (lowest number = highest priority)
  + Source / destination IP address, service tag or application security group
  + Protocol (TCP, UDP, ICMP, ESP, AH or any)
  + Direction
  + Port range
  + Action: Allow or deny

For inbound traffic Azure processes the rules in a network security group associated to a subnet first, if there is one, and then the rules in a network security group associated to the network interface, if there is one. For outbound traffic, Azure processes the rules in a network security group associated to a network interface first, if there is one, and then the rules in a network security group associated to the subnet, if there is one.

An Application Security Group (ASG) enables you to group virtual machines and define network security policies based on those groups.

**Design and implement Azure Firewall**

User configuration 
L3-L7 connectivity policies 
Microsoft Threat Intelligence 
Known malicious IPS and FQDNs 
Spoke I 
Spoke 2 
Spoke VNets 
j4¯e 
Azure 
Firewall 
• Threat intel, NAT. network and 
application traffic filtering rules 
allows inbound/outbound access 
Traffic is denied 
by default 
Central 
V Net 
Azure to on-prem 
traffic filtering 
On-premises 

Azure Firewall features:

* + Application FQDN filtering rules - You can limit outbound HTTP/S traffic or Azure SQL traffic to a specified list of fully qualified domain names (FQDN) including wild cards. This feature does not require TLS termination.
  + FQDN tags - These tags make it easy for you to allow well-known Azure service network traffic through your firewall. For example, say you want to allow Windows Update network traffic through your firewall. You create an application rule and include the Windows Update tag.
  + Service tags - A service tag represents a group of IP address prefixes to help minimize complexity for security rule creation. You cannot create your own service tag, nor specify which IP addresses are included within a tag. Microsoft manages the address prefixes encompassed by the service tag, and automatically updates the service tag as addresses change.
  + Threat intelligence - Threat intelligence-based filtering can be enabled for your firewall to alert and deny traffic from/to known malicious IP addresses and domains. The IP addresses and domains are sourced from the Microsoft Threat Intelligence feed.
  + Forced tunneling - You can configure Azure Firewall to route all Internet-bound traffic to a designated next hop instead of going directly to the Internet.
  + Web categories (preview) - Web categories let administrators allow or deny user access to web site categories such as gambling websites, social media websites, and others. Web categories are included in Azure Firewall Standard, but it is more fine-tuned in Azure Firewall Premium Preview.

With classic rules, rule collections are processed according to the rule type in priority order, lower numbers to higher numbers from 100 to 65,000.

With Firewall Policy, rules are organized inside Rule Collections which are contained in Rule Collection Groups. Rule Collections can be of the following types:

* + DNAT (Destination Network Address Translation)
  + Network
  + Application

You can define multiple Rule Collection types within a single Rule Collection Group, and you can define zero or more Rules in a Rule Collection, but the rules within a Rule Collection must be of the same type (i.e., DNAT, Network, or Application).

With Firewall Policy, rules are processed based on Rule Collection Group Priority and Rule Collection priority. Priority is any number between 100 (highest priority) and 65,000 (lowest priority).

In the case of a Firewall Policy being inherited from a parent policy, Rule Collection Groups in the parent policy always takes precedence regardless of the priority of the child policy.

Application rules are always processed after network rules, which are themselves always processed after DNAT rules regardless of Rule Collection Group or Rule Collection priority and policy inheritance.

If there is no network rule match, and if the protocol is either HTTP, HTTPS, or MSSQL, then the packet is then evaluated by the application rules in priority order. For HTTP, Azure Firewall looks for an application rule match according to the Host Header, whereas for HTTPS, Azure Firewall looks for an application rule match according to Server Name Indication (SNI) only.

Note: Application rules aren't applied for inbound connections. So, if you want to filter inbound HTTP/S traffic, you should use Web Application Firewall (WAF).

For enhanced security, if you modify a rule to deny access to traffic that had previously been allowed, any relevant existing sessions are dropped.

When deploying Azure Firewall, you can configure it to span multiple Availability Zones for increased availability. When you configure Azure Firewall in this way your availability increases to 99.99% uptime. Availability Zones can only be configured during firewall deployment. You cannot configure an existing firewall to include Availability Zones.

**Secure your networks with Azure Firewall Manager**

Azure Firewall Manager simplifies the process of centrally defining network and application-level rules for traffic filtering across multiple Azure Firewall instances. You can span different Azure regions and subscriptions in hub and spoke architectures for traffic governance and protection.

Global admin 
Global plicy 
e zscaw 
Virtual WAN 
Azure r9ion N 
Local 
Azure FirewaU 
Hub VNET 
Azure Firewall 
Secured vHub 
End 

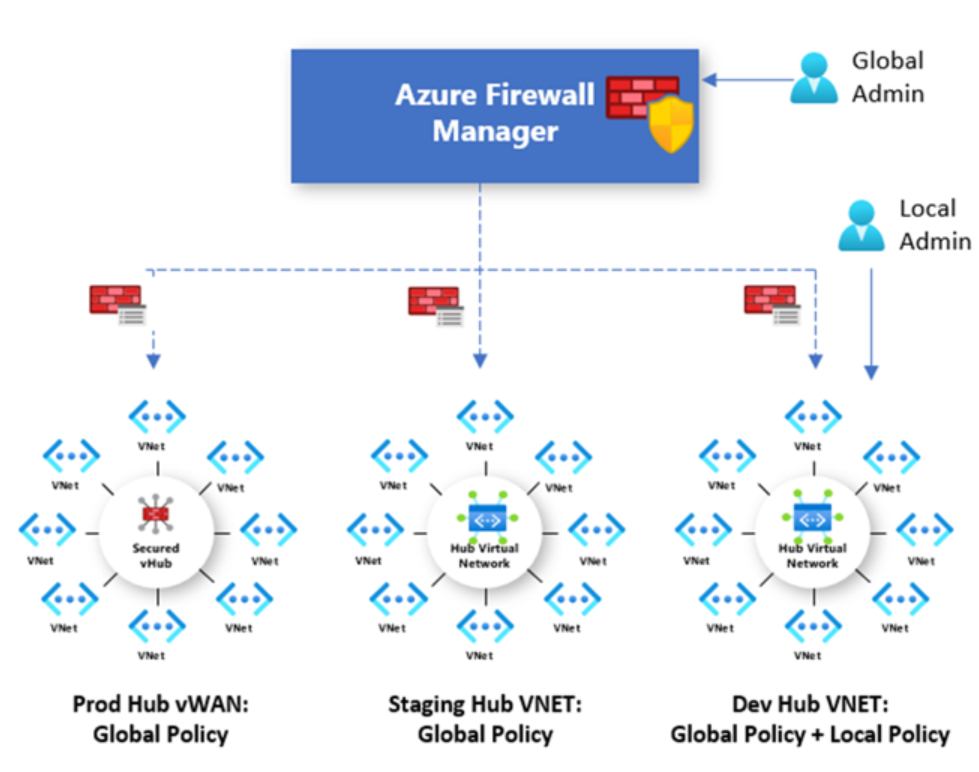
Firewall Manager can provide security management for two network architecture types:

* + **Secured Virtual Hub** - This is the name given to any Azure Virtual WAN Hub when security and routing policies have been associated with it.
  + **Hub Virtual Network** - This is the name given to any standard Azure virtual network when security policies are associated with it.

The key features offered by Azure Firewall Manager are:

* + Central Azure Firewall deployment and configuration
  + Hierarchical policies (global and local): centrally manage Azure Firewall policies across multiple secured virtual hubs. Your central IT teams can author global firewall policies to enforce organization wide firewall policy across teams. Locally authored firewall policies allow a DevOps self-service model for better agility
  + Integrated with third-party security-as-a-service for advanced security: available only with secured virtual hub deployments
  + Centralized route management: easily route traffic to your secured hub for filtering and logging without the need to manually set up User Defined Routes (UDR) on spoke virtual networks. This feature is available only with secured virtual hub deployments
  + Region availability

A Firewall policy is an Azure resource that contains NAT, network, and application rule collections and Threat Intelligence settings. It is a global resource that can be used across multiple Azure Firewall instances in Secured Virtual Hubs and Hub Virtual Networks. New policies can be created from scratch or inherited from existing policies. Inheritance allows DevOps to create local firewall policies on top of organization mandated base policy. Policies work across regions and subscriptions. You can create Firewall Policy and associations with Azure Firewall Manager. However, you can also create and manage a policy using REST API, templates, Azure PowerShell, and the Azure CLI. Once you create a policy, you can associate it with a firewall in a virtual WAN hub making it a Secured Virtual Hub and/or associate it with a firewall in a standard Azure virtual network making it a Hub Virtual Network.



Note: You can create a Hub Virtual Network using Azure Firewall Manager and peering spoke virtual networks to it using virtual network peering

Note: While in Hub Virtual Networks you need to configure user defined routes to route traffic to your hub virtual network firewall, in Secure Virtual Hub mode, you can do this without UDR on spokes and instead by using the Secure Virtual Hub Route Setting page.

You cannot have more than one hub per virtual WAN per region, however you can add multiple virtual WANs in the region to achieve this

**Implement a Web Application Firewall on Azure Front Door**

When you create a Web Application Firewall (WAF) policy, by default the WAF policy is in Detection mode. In Detection mode, WAF does not block any requests; instead, requests matching the WAF rules are logged at WAF logs. To see WAF in action, you can change the mode settings from Detection to Prevention. In Prevention mode, requests that match rules that are defined in Default Rule Set (DRS) are blocked and logged at WAF logs.

Azure-managed rule sets -enabled by default- provide an easy way to deploy protection against a common set of security threats. Since such rule sets are managed by Azure, the rules are updated as needed to protect against new attack signatures. You can disable individual rules.

Azure-managed Default Rule Set includes rules against the following threat categories:

* + Cross-site scripting
  + Java attacks
  + Local file inclusion
  + PHP injection attacks
  + Remote command execution
  + Remote file inclusion
  + Session fixation
  + SQL injection protection
  + Protocol attackers

You can also have custom rules. A custom WAF rule consists of a priority number, rule type, match conditions, and an action. There are two types of custom rules:

* + match rules: controls access based on a set of matching conditions
  + rate limit rules: controls access based on matching conditions and the rates of incoming requests

**Design and implement private access to Azure Services**

**Define Private Link Service and private endpoint**

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 in your virtual network. Private Link achieves this by replacing a resource's public endpoint with a private network interface. There are three key points to consider with this new architecture:

* + The Azure resource becomes, in a sense, a part of your virtual network.
  + You can configure the Azure resource to no longer expose its public IP address, which eliminates that potential security risk.

Private Endpoint is the key technology behind Private Link. It is the network interface that replaces the resource's public endpoint. Private Endpoint uses a private IP address from the VNet to bring the service into the VNet.

Azure Private Link service is the reference to your own service that is powered by Azure Private Link. Your service that is running behind Azure standard load balancer can be enabled for Private Link access so that consumers to your service can access it privately from their own VNets. Your customers can create a private endpoint inside their VNet and map it to this service. A Private Link service receives connections from multiple private endpoints. A private endpoint connects to one Private Link service.

Machine generated alternative text:
Network Security Group: Deny Inbound 
On-premises 
Express Route 
private 
peering 
Express Route 
Network Security Group: Deny Outbound 
1000.5 
Private 
endpoint 
subnet (10.0.1.0/24) 
Azure Private Link 
Traffic carried over 
Microsoft Network 
NAT IP 
(192.168.0.5) 
Private 
Link Service 
192.168.0.1 
Frontend IP 
(192.168.0.10) 
192.168.0.2 
VMSS 
Standard Load 
Balancer (ILB/PLB) 
subnet (192.168.0.0/24) 
Virtual Network (10.0.0.0/16) 
Consumer Network 
(Azure AD tenant A. Subscription A. Region A) 
Virtual Network (192.168.0.0/16) 
Provider Network 
(Azure AD tenant B, Subscription B, Region B) 

When configuring a Private Link, you can connect to a resource in your directory or using the resource ID.

Network connections can only be initiated by clients connecting to the Private Endpoint, Service providers do not have any routing configuration to initiate connections into service consumers. Connections can only be established in a single direction.

The Private Endpoint must be deployed in the same region and subscription as the virtual network.

The Private Link resource can be deployed in a different region than the virtual network and Private Endpoint.

**Explain virtual network service endpoints**

By default, Azure services are all designed for direct internet access. All Azure resources have public IP addresses, including PaaS services, such as Azure SQL Database and Azure Storage. Use virtual network Service Endpoints to extend your private address space in Azure by providing a direct connection to your Azure services. Service Endpoints let you secure your Azure resources to only your virtual network. Service traffic will remain on the Azure backbone and doesn't go out to the internet.

To enable a Service Endpoint, you must do the following two things:

* + Turn off public access to the service.
  + Add the Service Endpoint to a virtual network.

If you implement a virtual network Service Endpoint for a service, such as Azure Storage or Azure SQL Database, Azure adds a route to a virtual network subnet for the service. The address prefixes in the route are the same address prefixes, or CIDR ranges, as those of the corresponding service tag.

**Integrate Private Link with DNS**

Private DNS zones are typically hosted centrally in the same Azure subscription where the hub VNet is deployed. This central hosting practice is driven by cross-premises DNS name resolution and other needs for central DNS resolution such as Active Directory. In most cases, only networking/identity admins have permissions to manage DNS records in these zones.

The following diagram shows a typical high-level architecture for enterprise environments with central DNS resolution and where name resolution for Private Link resources is done via Azure Private DNS:

1 4 0 1 41 
~ 1 亂 10 1 ~ ~ 1 ㄩ 
1 1 【 40 
63 , 1 16 
100.1-13S 

In the above diagram:

* + On-premises DNS servers have conditional forwarders configured for each Private Endpoint public DNS zone forwarder pointing to the DNS forwarders (10.100.2.4 and 10.100.2.5) hosted in the hub VNet.
  + The DNS servers 10.100.2.4 and 10.100.2.5 hosted in the hub VNet use the Azure-provided DNS resolver (168.63.129.16) as a forwarder.
  + All Azure VNets have the DNS forwarders (10.100.2.4 and 10.100.2.5) configured as the primary and secondary DNS servers.
  + Central networking and/or central platform teams must ensure that whenever Private Endpoints are created, the corresponding records are automatically created in the centralized private DNS zone that matches the service created.

Note: It's important to correctly configure your DNS settings to resolve the Private Endpoint IP address to the fully qualified domain name (FQDN) of the connection string. The network interface associated with the Private Endpoint contains the information to configure your DNS. The network interface information includes FQDN and private IP addresses for your Private Link resource. Existing Microsoft Azure services might already have a DNS configuration for a public endpoint. This configuration must be overridden to connect using your private endpoint.

IP address 168.63.129.16 is a virtual public IP address that is used to facilitate a communication channel to Azure platform resources.

Azure creates a canonical name DNS record (CNAME) on the public DNS. The CNAME record redirects the resolution to the private domain name. You can override the resolution with the private IP address of your Private Endpoints. Your applications don't need to change the connection URL. When resolving to a public DNS service, the DNS server will resolve to your Private Endpoints. The process doesn't affect your existing applications.

Private networks already using the private DNS zone for a given type, can only connect to public resources if they don't have any Private Endpoint connections, otherwise a corresponding DNS configuration is required on the private DNS zone in order to complete the DNS resolution sequence.

Azure Firewall DNS proxy can be used as DNS forwarder for On-premises workloads and Virtual network workloads using a DNS forwarder.

**Azure Private Endpoint DNS configuration**: It's important to correctly configure your DNS settings to resolve the Private Endpoint IP address to the fully qualified domain name (FQDN) of the connection string. You can use the following options to configure your DNS settings for Private Endpoints:

* + **Use the host file (only recommended for testing)**
  + **Use a private DNS zone**
  + **Use your DNS forwarder (optional):** Create a DNS forwarding rule to use a private DNS zone on your DNS server hosted in a virtual network.

**DNS configuration scenarios**: The FQDN of the services resolves automatically to a public IP address. To resolve to the private IP address of the Private Endpoint, change your DNS configuration.

* + **Vnet workloads without custom DNS Server**: the client queries for the Private Endpoint IP address to the Azure-provided DNS service 168.63.129.16. Azure DNS will be responsible for DNS resolution of the private DNS zones. The same can be applied to peer-ed Vnets or on-prem clients (for on-prem, you need a DNS forwarded deployed in your Azure Vnet).

Virtual 
Network link 
Client VM 
Private DNS 
Azure Recursive 
16&63.129.16 
DNS Resolution Flow 
O DNS query for azsqll database.windows.net 
O Authoritative DNS query for azsql I .database.windows.net 
Response. CNAME azsql l.privatelink.database.windows.net 
O DNS query for azsqll.privatelink.database.windows.net 
Response: private ip address 10.5.0.5 
O Response: CNAME azsql I 
A azsqll.privatelink.database.windows.net 10-50.5 
o 
Private connection to 1050.5 
DNS traffic 
Virtual network lit* 
Rivate connection 
: azsql I database.windows.net 
VNet-consumer-mI 
10±00/24 

* + **On-premises workloads using a DNS forwarder**: identical model to the above. For on-premises workloads to resolve the FQDN of a Private Endpoint, use a DNS forwarder to resolve the Azure service public DNS zone in Azure. A DNS forwarder is a Virtual Machine running on the Virtual Network linked to the Private DNS Zone that can proxy DNS queries coming from other Virtual Networks or from on-premises. This is required as the query must be originated from the Virtual Network to Azure DNS.

**Integrate your App Service with Azure virtual networks**

Azure App Service has two variations on the VNet Integration feature:

* + **The multitenant systems** that support the full range of pricing plans except Isolated.
  + **The App Service Environment**, which deploys into your VNet and supports Isolated pricing plan apps.

The VNet Integration feature is used in multitenant apps. If your app is in App Service Environment, then it's already in a VNet and doesn't require use of the VNet Integration feature to reach resources in the same VNet.

Note: VNet Integration gives your app access to resources in your VNet, but it doesn't grant inbound private access to your app from the VNet. VNet Integration is used only to make outbound calls from your app into your Vnet

The VNet Integration feature has two variations:

* + Regional VNet Integration: When you connect to Azure Resource Manager virtual networks in the same region, you must have a dedicated subnet in the VNet you are integrating with.
  + Gateway-required VNet Integration: When you connect to VNet in other regions or to a classic virtual network in the same region, you need an Azure Virtual Network gateway provisioned in the target VNet.

(i.e. If the VNet is in the same region, either create a new subnet or select an empty preexisting subnet. If the VNet is in another region, you must have a VNet gateway provisioned with point to site enabled.)

Gateway-required VNet Integration provides access to resources only in the target VNet or in networks connected to the target VNet with peering or VPNs. Gateway-required VNet Integration doesn't enable access to resources available across Azure ExpressRoute connections or work with Service Endpoints.

Regional VNet integration supports access to Service Endpoints, resources across ExpressRoute and peer-ed vnets.

By default, your app routes only RFC1918 traffic into your VNet. If you want to route all your outbound traffic into your VNet, you must add the WEBSITE\_VNET\_ROUTE\_ALL setting in your app.

When you route all your outbound traffic into your VNet, it's subject to the NSGs and UDRs that are applied to your integration subnet. When WEBSITE\_VNET\_ROUTE\_ALL is set to 1, outbound traffic is still sent from the addresses that are listed in your app properties, unless you provide routes that direct the traffic elsewhere.

Regional VNet integration can't use port 25. An app that uses regional VNet Integration can use a network security group to block outbound traffic to resources in your VNet or the internet. To block traffic to public addresses, you must have the application setting WEBSITE\_VNET\_ROUTE\_ALL set to 1. The inbound rules in an NSG don't apply to your app because VNet Integration affects only outbound traffic from your app.

To control inbound traffic to your app, use the Access Restrictions feature. An NSG that is applied to your integration subnet is in effect regardless of any routes applied to your integration subnet. If WEBSITE\_VNET\_ROUTE\_ALL is set to 1 and you do not have any routes that affect public address traffic on your integration subnet, all your outbound traffic is still subject to NSGs assigned to your integration subnet. When WEBSITE\_VNET\_ROUTE\_ALL is not set, NSGs are only applied to RFC1918 traffic.

You can use route tables to route outbound traffic from your app to wherever you want. By default, route tables only affect your RFC1918 destination traffic. When you set WEBSITE\_VNET\_ROUTE\_ALL to 1, all your outbound calls are affected. Routes that are set on your integration subnet will not affect replies to inbound app requests. Common destinations can include firewall devices or gateways.

After your app integrates with your VNet, it uses the same DNS server that your VNet is configured with. By default, your app won't work with Azure DNS private zones. To work with Azure DNS private zones, you need to add the following app settings:

* + WEBSITE\_DNS\_SERVER with value 168.63.129.16
  + WEBSITE\_VNET\_ROUTE\_ALL with value 1

If you want to make calls to Private Endpoints, then you must make sure that your DNS lookups resolve to the Private Endpoint.

**Configure Azure Kubernetes Service (AKS) for regional VNET integration**

To provide network connectivity, AKS clusters can use kubenet (basic networking) or Azure CNI (advanced networking).

With kubenet, only the nodes receive an IP address in the virtual network subnet. Pods can't communicate directly with each other. Instead, User Defined Routing (UDR) and IP forwarding is used for connectivity between pods across nodes. By default, UDRs and IP forwarding configuration is created and maintained by the AKS service, but you have to the option to bring your own route table for custom route management.

Nodel (192.168.1.4/24) 
N0de2 (192.168.1.5/24) 
10.24402 
t 
10.244.0.3 
I 
10.24404 
10.244.1.2 
10.244.1.3 
Routing / NAT 
10.244.1.4 
Routing / NAT 
Azure virtual network 
Kubenet 
Node Subnet 
Kubenet 
192.168.1.0/24 
192.168.0.0/16 

Route tables and user-defined routes are required for using kubenet, which adds complexity to operations. Direct pod addressing isn't supported for kubenet due to kubenet design. Unlike Azure CNI clusters, multiple kubenet clusters can't share a subnet

Use kubenet when:

* + You have limited IP address space.
  + Most of the pod communication is within the cluster.
  + You don't need advanced AKS features such as virtual nodes or Azure Network Policy. Use Calico network policies.

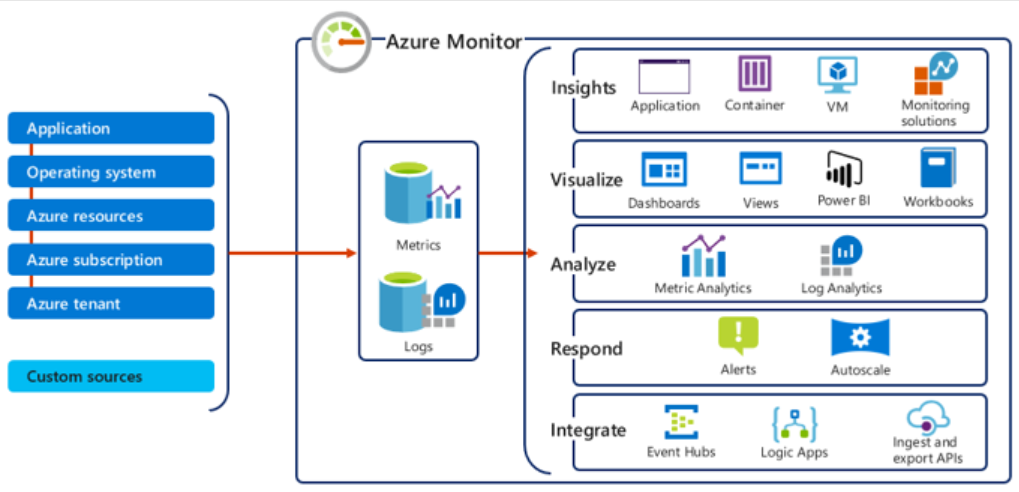
Use Azure CNI when:

* + You have available IP address space.
  + Most of the pod communication is to resources outside of the cluster.
  + You don't want to manage user defined routes for pod connectivity.
  + You need AKS advanced features such as virtual nodes or Azure Network Policy.

**Design and implement network monitoring**

**Monitor your networks using Azure monitor**

Azure Monitor



The data collected by Azure Monitor fits into one of two fundamental types:

* + Metrics - Metrics are numerical values that describe some aspect of a system at a particular point in time. They are lightweight and capable of supporting near real-time scenarios.
  + Logs - Logs contain different kinds of data organized into records with different sets of properties for each type. Telemetry such as events and traces are stored as logs in addition to performance data so that it can all be combined for analysis.

Azure Monitor Metrics is a feature of Azure Monitor that collects numeric data from monitored resources into a time series database, particularly useful for alerting and fast detection of issues.

There are four fundamental sources of metrics collected by Azure Monitor. Once these metrics are collected in the Azure Monitor metric database, they can be evaluated together regardless of their source.

* + Azure resources
  + Applications
  + Virtual machine agents
  + Custom metrics

The data collected by Azure Monitor Metrics is stored in a time-series database which is optimized for analyzing time-stamped data. Each set of metric values is a time series with the following properties:

* + The time the value was collected
  + The resource the value is associated with
  + A namespace that acts like a category for the metric
  + A metric name
  + The value itself

**Monitor network resources with Azure Monitor Network Insights**

You can use the Insights>Networks section in Azure Monitor to obtain a broad view of health and metrics for all your deployed network resources, without requiring any configuration. It also provides access to network monitoring features such as Connection Monitor, flow logging for network security groups (NSG) flow logs, and Traffic Analytics, and it provides other network diagnostic features.

Azure Monitor Network Insights is structured around these key components of monitoring:

* + Network health and metrics
  + Connectivity
  + Traffic
  + Diagnostic Toolkit

Machine generated alternative text:



The **Network health** tab of Azure Monitor Network Insights offers a simple method for visualizing an inventory of your networking resources, together with resource health and alerts. It is divided into four key functionality areas:

* + search and filtering
  + resource health and metrics: get an overview of the health status of your various network resources.
  + Alerts: provides a view of all alerts generated for the selected resources across all your subscriptions
  + dependency view: helps you visualize how a resource is configured

The **Connectivity** tab provides an easy way to visualize all tests configured via Connection Monitor and Connection Monitor (classic) for the selected set of subscriptions.

Tests are grouped by Sources and Destinations tiles and display the reachability status for each test. Reachable settings provide easy access to configurations for your reachability criteria, based on Checks failed(%) and RTT(ms).

The **Traffic** tab provides access to all NSGs configured for NSG flow logs and Traffic Analytics for the selected set of subscriptions, grouped by location.

The **Diagnostic Toolkit** drop-down list provides to access to the following network monitoring features:

* + Network Watcher packet capture
  + Network Watcher VPN Troubleshoot
  + Network Watcher Next hop
  + Network Watcher IP flow verify

**Monitor your networks using Azure network watcher**

Azure Network Watcher is a regional service that enables you to monitor and diagnose conditions at a network scenario level in, to, and from Azure. Scenario level monitoring enables you to diagnose problems at an end-to-end network level view. Network Watcher is enabled through the creation of a Network Watcher resource, which allows you to utilize Network Watcher capabilities. Network Watcher is designed to monitor and repair the network health of IaaS products which includes Virtual Machines, Virtual Networks, Application Gateways, and Load Balancers.

* + Automate remote network monitoring with packet capture
  + Gain insight into your network traffic using flow logs
  + Diagnose VPN connectivity issues
  + Verify IP flow
  + Next hop
  + Effective security rules
  + NSG flow logs: NSG Flow Logs maps IP traffic through a network security group. These capabilities can be used in security compliance and auditing.
  + Packet capture
  + Connection troubleshoot
  + Network topology

Note: To use Network Watcher, you must be an Owner, Contributor, or Network Contributor. If you create a custom role, the role must be able to read, write, and delete the Network Watcher.

When you create or update a virtual network in your subscription, Network Watcher will be enabled automatically in your Virtual Network's region. There is no impact to your resources or associated charge for automatically enabling Network Watcher.

Note: The Network Watcher instance is automatically created in a resource group named NetworkWatcherRG. The resource group is created if it does not already exist

**Connection Monitor** provides unified end-to-end connection monitoring in Azure Network Watcher. The Connection Monitor feature supports hybrid and Azure cloud deployments. Network Watcher provides tools to monitor, diagnose, and view connectivity-related metrics for your Azure deployments. Connection Monitor combines the best of two features: the Network Watcher Connection Monitor (Classic) feature and the Network Performance Monitor (NPM) Service Connectivity Monitor, ExpressRoute Monitoring, and Performance Monitoring feature.

Connection Monitor relies on lightweight executable files to run connectivity checks. It supports connectivity checks from both Azure environments and on-premises environments. The data that Connection Monitor collects is stored in the Log Analytics workspace. You set up this workspace when you created the connection monitor. Monitoring data is also available in Azure Monitor Metrics.

**Traffic Analytics** is a cloud-based solution that provides visibility into user and application activity in cloud networks. Traffic Analytics analyzes Network Watcher network security group (NSG) flow logs to provide insights into traffic flow in your Azure cloud and provide rich visualizations of data written to NSG flow logs. The key components of Traffic Analytics are:

* + NSGs
  + NSG flow logs
  + Log analytics
  + Log analytics workspace
  + Network watcher

Traffic analytics examines the raw NSG flow logs and captures reduced logs by aggregating common flows among the same source IP address, destination IP address, destination port, and protocol

Before enabling NSG flow logging, you must have a network security group to log flows for. If you do not have a network security group, then you must create one