

Designing for Network Redundancy



Jeff Landry

AUTHOR



Overview



Azure load balancers

- Internal
- Public

Other load balancing solutions

- Application Gateway
- Traffic Manager

Extending on-premises to Azure

- ExpressRoute
- Other VPN options



Overview



How to create virtual networks

- Network security groups

Placing servers in virtual networks

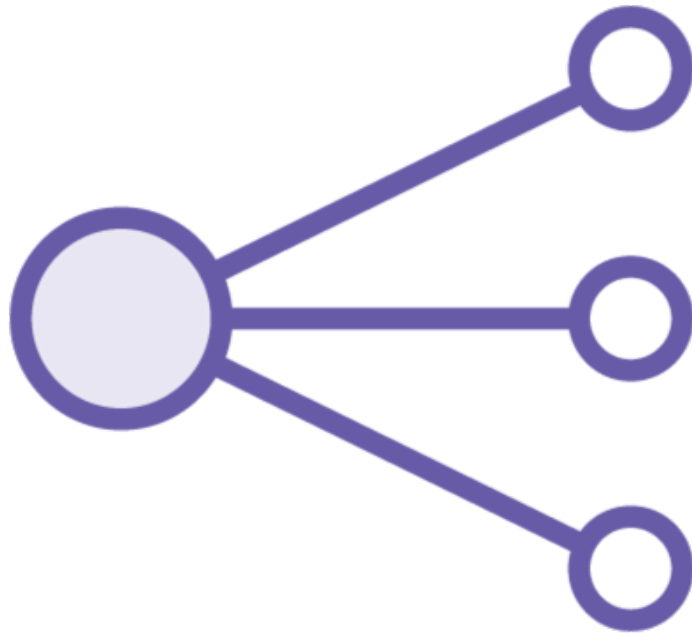
- Use availability zones

Virtual network peering



Designing for High Availability with Azure Load Balancers





Azure load balancers

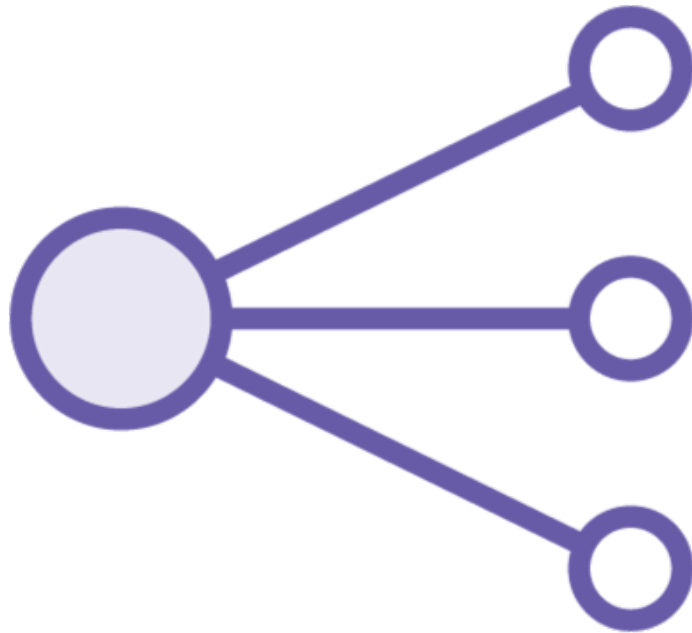
- Supports availability zones
- Operates at the transport layer
 - OSI model layer 4

TCP and UDP traffic routed based on:

- Source IP address and port
- Destination IP address and port

Even distribution of incoming traffic

- Across backend servers



Load balancing rule

- Session persistence

Health check

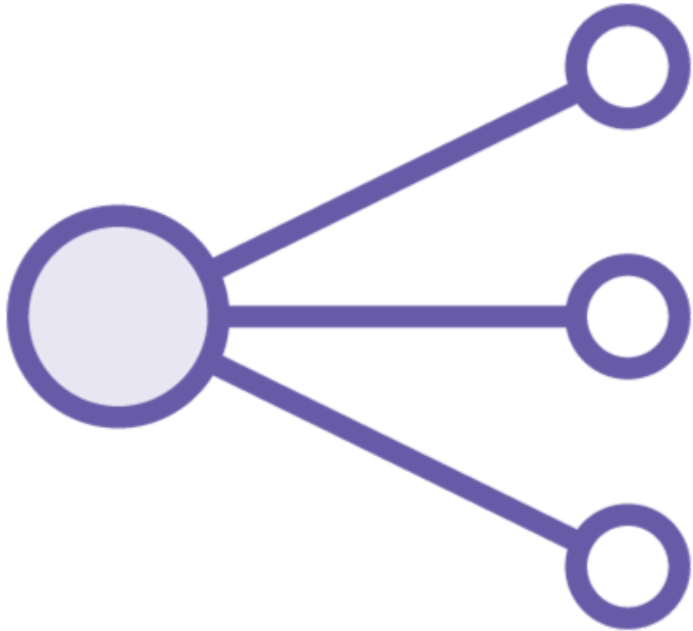
- Responding to HTTP requests

Easily scale your application

- Add or remove servers seamlessly

Supports TCP and UDP protocols

- SMTP
- HTTP
- HTTPS



Load balancer types

Public – External load balancer

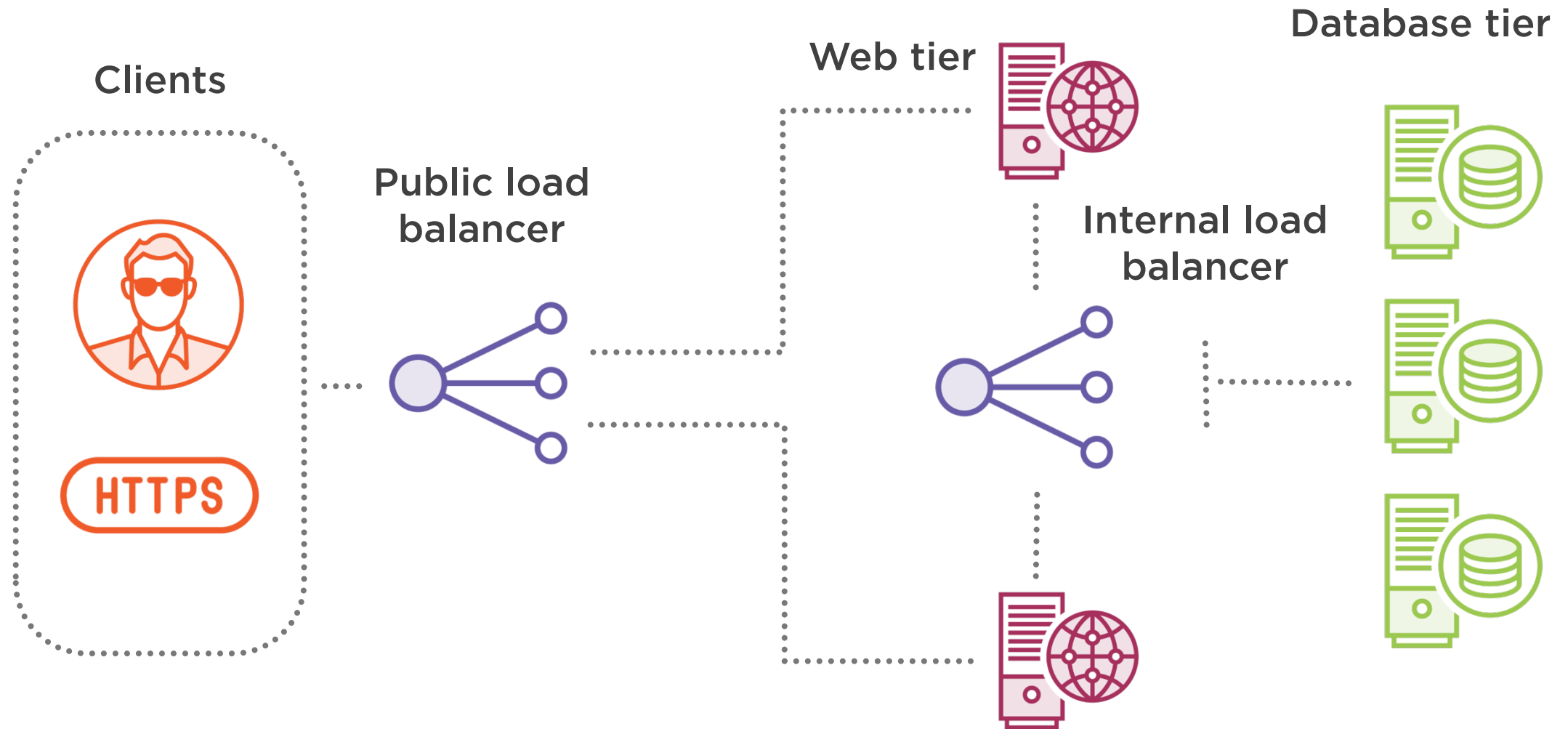
- Incoming traffic from the internet

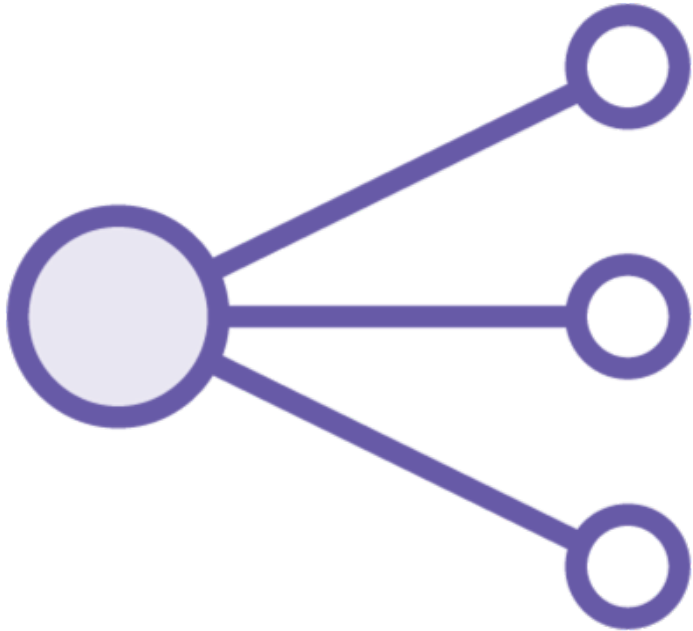
Internal – Private load balancer

- Incoming traffic from internal network

Both can be used within an environment

Public and Internal Load Balancers





Load balancers are aware of failures

- Will stop sending traffic to the server

Health probes

- Checks if server is healthy

Transparent from a client's perspective

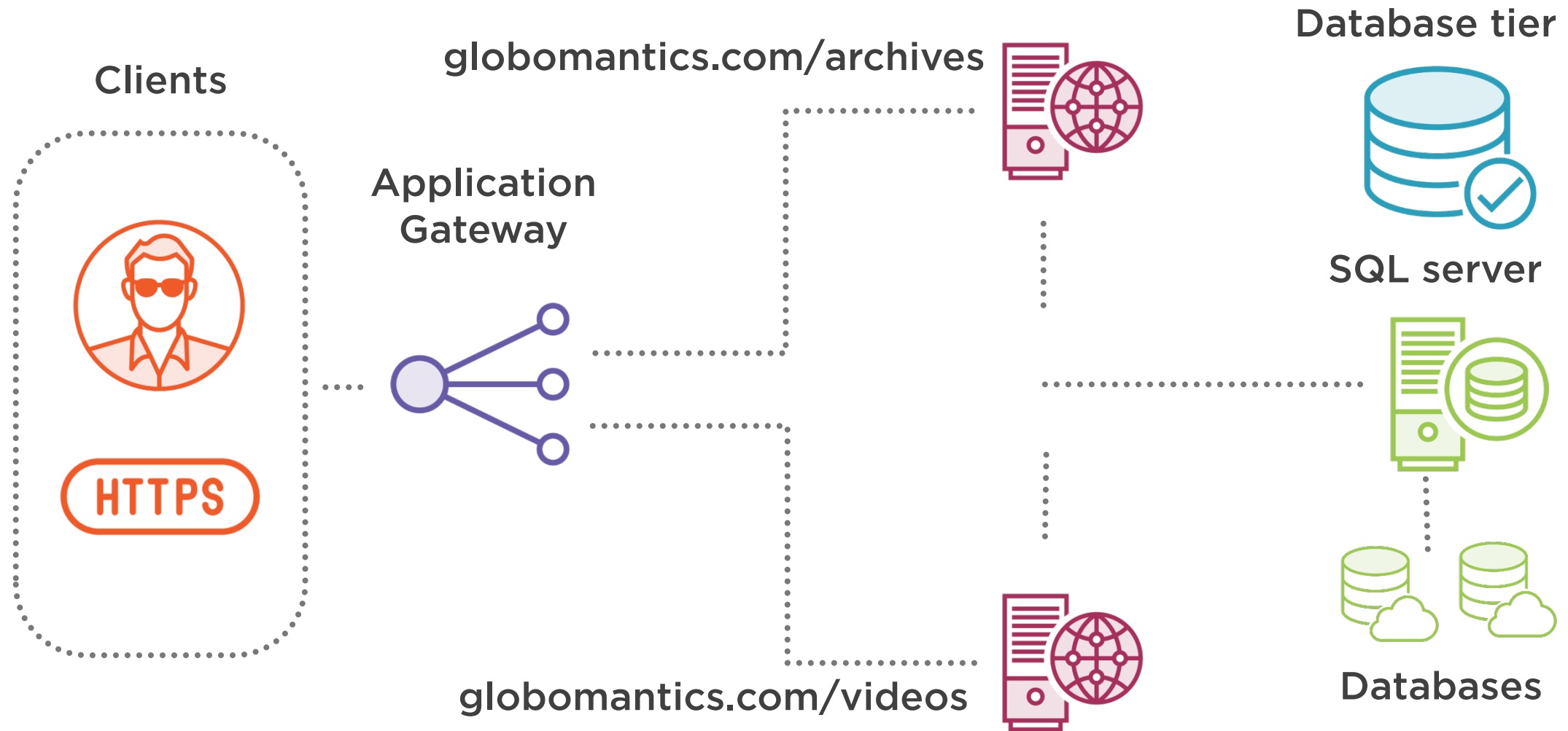
Great solution when designing HA

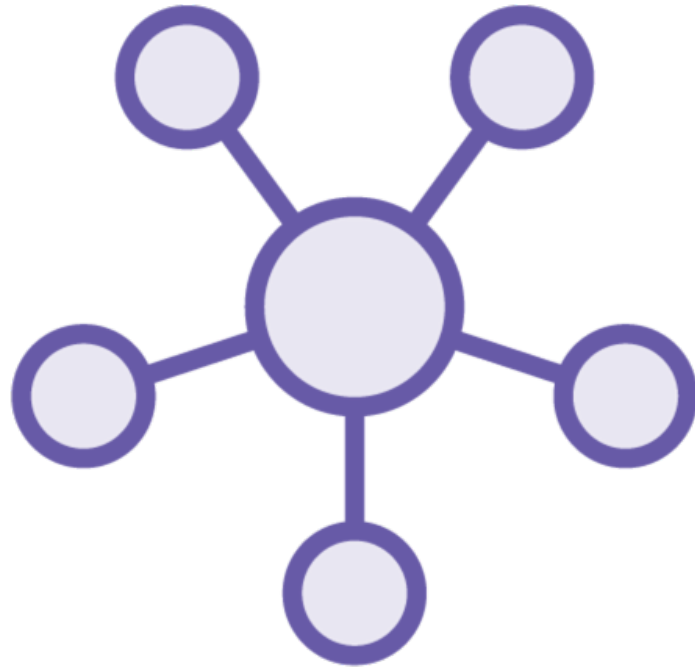
- Within a single region

Application Gateway and Traffic Manager



Application Gateway URL Path Based Routing





Application Gateway features

- Web application firewall rules - WAF
- SSL offload and SSL policy
- Cookie-based session affinity

Traffic Manager

- Load balancer for DNS-based traffic
 - Across Azure regions
 - Internet-facing services
 - Endpoints

Azure Front Door

- Traffic routed to the nearest region

Extending to Azure and High Availability



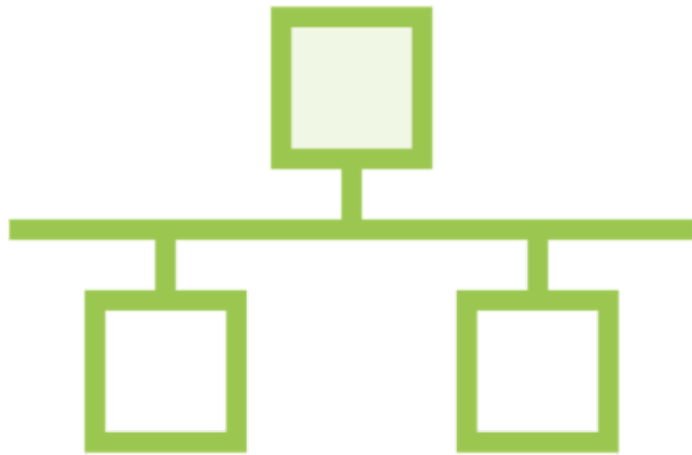
Extending on-premises to Azure

**Active Directory
Domain Services**

Internal databases

**Azure as a disaster
recovery site**





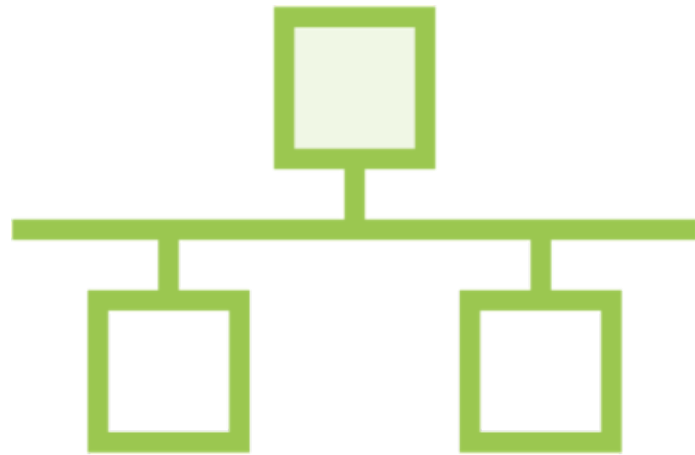
Connectivity options

- Virtual private network
 - VPN
- Azure VPN Gateway

Point-to-site VPN - P2S

- Only few users
 - Working from home
 - Hotel rooms
 - Other networks
 - Requires an installer package



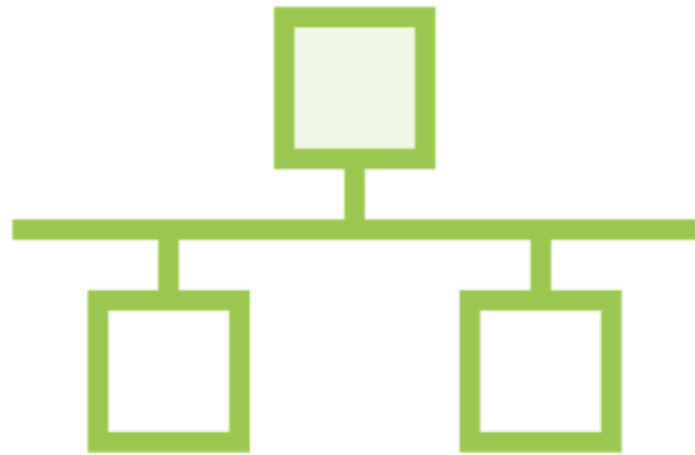


Site-to-site VPN - S2S

- IPsec/IKE VPN tunnel
- Requires compatible VPN device
- Public IPv4 address
- No installer package required

ExpressRoute

- High speed connexion
 - Uses a physical circuit
 - Does not use internet
 - Faster and secure
 - Bandwidth from 50 Mbps and 10 Gbps



Azure virtual network and high availability

ExpressRoute and S2S VPN failover

- ExpressRoute acts as primary

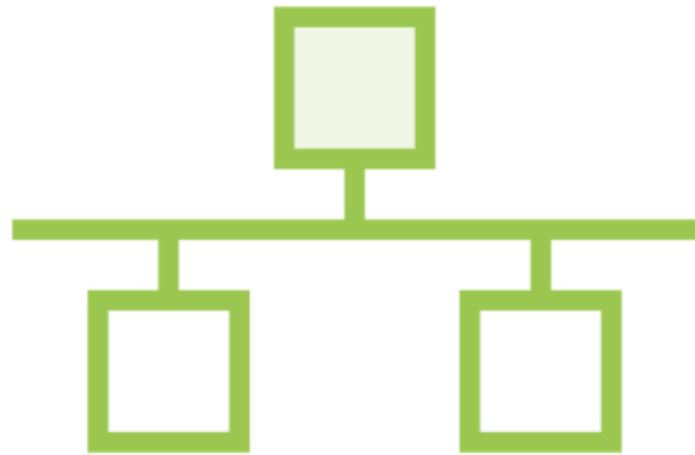
ExpressRoute no longer available

- Failover to S2S VPN



Virtual Network Peering





Virtual network

- Isolation boundary
- Address space
- Subnet

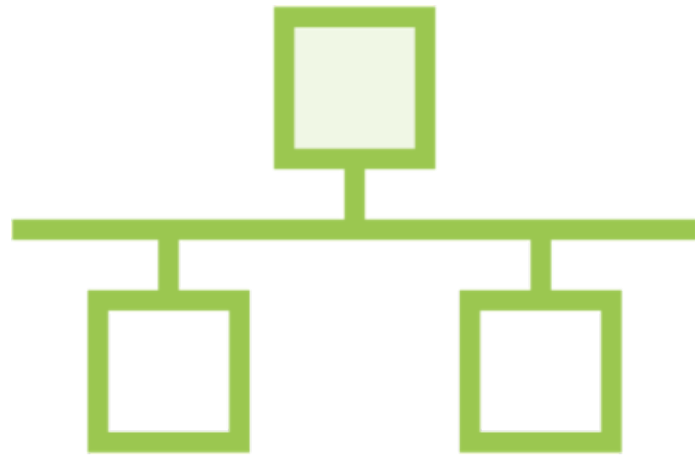
Within the same virtual network

- Virtual machines can communicate

No connectivity between virtual networks

- Default configuration





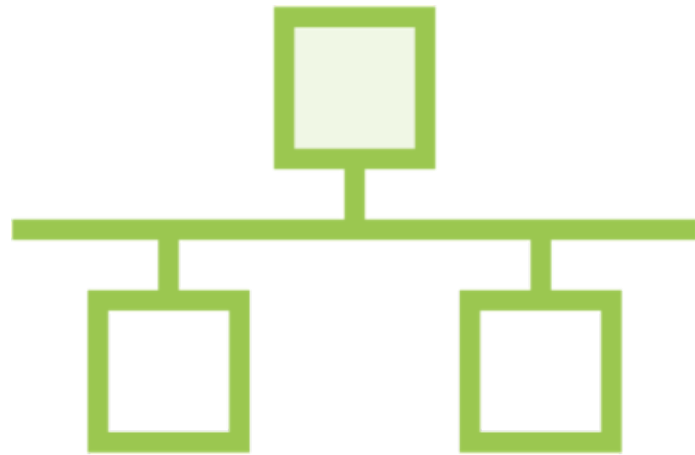
Virtual network peering

- Seamless connectivity
- Can exist in different subscriptions

Types of peering

- Virtual network peering
 - Within the same Azure region
 - Not protected from regional outage
- Global virtual network peering
 - Across different Azure regions

Resiliency and high availability



Network security group (NSG)

- Protects your virtual network

Binds to a virtual network interface

- VNIC

Binds at the subnet level

- Inbound and outbound rules inherited
 - Easier to manage

Overview



Load balancing solutions

- Public load balancer
- Private load balancer
- Combination of both

Application Gateway

- Web applications
- SSL offload
- Cookie-based session affinity
- Web application firewall rules



Overview



Traffic Manager

- DNS-based traffic
- Global load balancing solution

Extending on-premises to Azure

- Azure VPN Gateway
- ExpressRoute
- Combination of both

Azure virtual networks

- Virtual network peering

