**Best Practices to setup networking for workloads migrated to Azure**

**Content:**

1. VNet
2. Plan IP Addressing
3. Implement a hub and spoke network topology
4. Subnets
5. Setup Availability Zones
6. Hybrid cloud networking
7. High availability Site-to-Site VPN
8. ExpressRoute Routing
9. Application security groups

**VNet**

* Azure resources communicate privately, directly, and securely with each other over VNets.
* You can configure endpoint connections on VNets for VMs and services that require internet communication.
* A VNet is a logical isolation of the Azure cloud that's dedicated to your subscription.
* You can implement multiple VNets within each Azure subscription and Azure region.
* Each VNet is isolated from other VNets
* VNets can connect to each other using VNet peering
* A virtual network is scoped to a single region; however, multiple virtual networks from different regions can be connected together using Virtual Network Peering

When planning your VNet topology, you should consider how to arrange IP address spaces, how to implement a hub and spoke network, how to segment VNets into subnets, setting up DNS, and implementing Azure availability zones

**Plan IP Addressing**

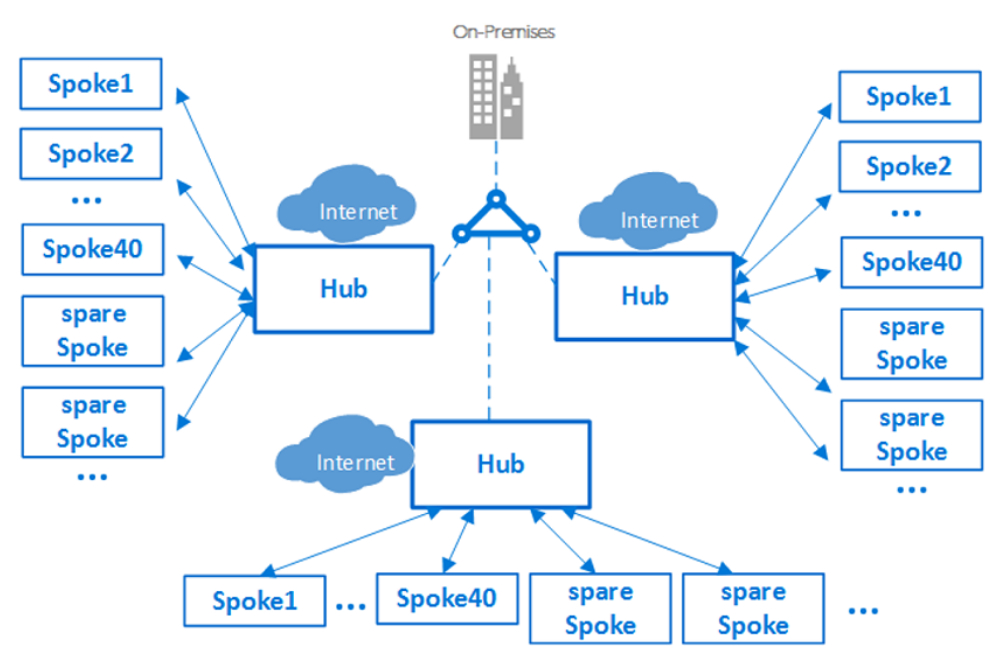
When you create VNets as part of your migration, it's important to plan out your VNet IP address space.

* You should assign an address space that isn't larger than a CIDR range of /16 for each VNet. VNets allow for the use of 65536 IP addresses, and assigning a smaller prefix than /16 would result in the loss of IP addresses. It's important not to waste IP addresses, even if they're in the private ranges defined by RFC 1918.
* The VNet address space shouldn't overlap with on-premises network ranges.
* Network Address Translation (NAT) shouldn't be used.
* Overlapping addresses can cause networks that can't be connected and routing that doesn't work properly. If networks overlap, you'll need to redesign the network or use network address translation (NAT).

**Implement Hub and Spoke network topology**

A hub and spoke network topology isolates workloads while sharing services such as identity and security

* Implementing a hub and spoke topology in Azure centralizes common services such as connections to on-premises networks, firewalls, and isolation between VNets. The hub VNet provides a central point of connectivity to on-premises networks, and a place to host services use by workloads hosted in spoke VNets.
* A hub and spoke configuration is typically used by larger enterprises. Smaller networks might consider a simpler design to save on costs and complexity.
* Spoke VNets can be used to isolate workloads, with each spoke managed separately from other spokes. Each workload can include multiple tiers, and multiple subnets that are connected with Azure load balancers.
* Hub and spoke VNets can be implemented in different resource groups, and even in different subscriptions. When you peer virtual networks in different subscriptions, the subscriptions can be associated to the same, or different, Azure Active Directory (Azure AD) tenants.



**Subnet**

To provide isolation within a VNet, you segment it into one or more subnets, and allocate a portion of the VNet's address space to each subnet.

* You can create multiple subnets within each VNet.
* By default, Azure routes network traffic between all subnets in a VNet.
* Your subnet decisions are based on your technical and organizational requirements.
* You create subnets using CIDR notation.
* When deciding on network range for subnets, it's important to note that Azure retains five IP addresses from each subnet that can't be used. For example, if you create the smallest available subnet of /29 (with eight IP addresses), Azure will retain five addresses, so you only have three usable addresses that can be assigned to hosts on the subnet

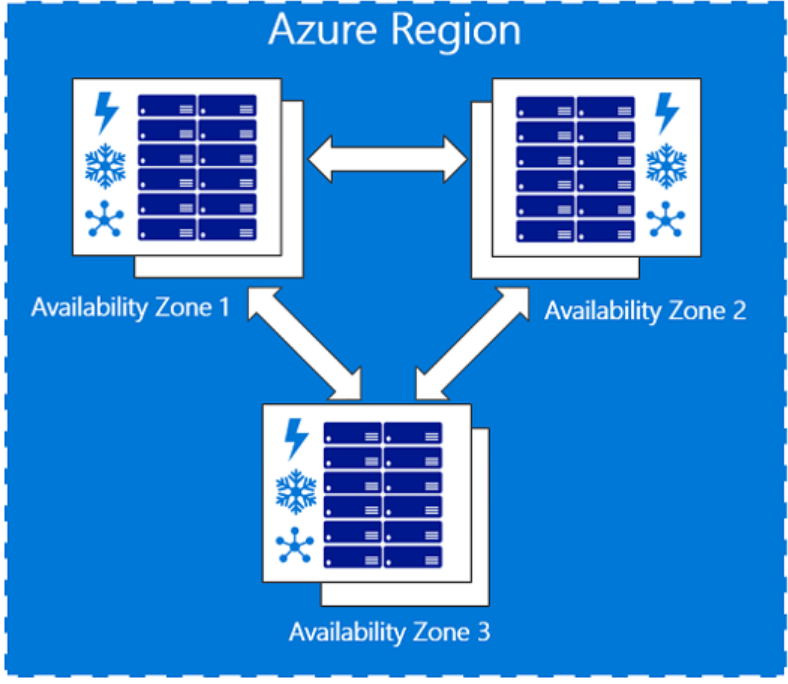
**Example:**

The table shows an example of a VNet with an address space of 10.245.16.0/20 segmented into subnets, for a planned migration.

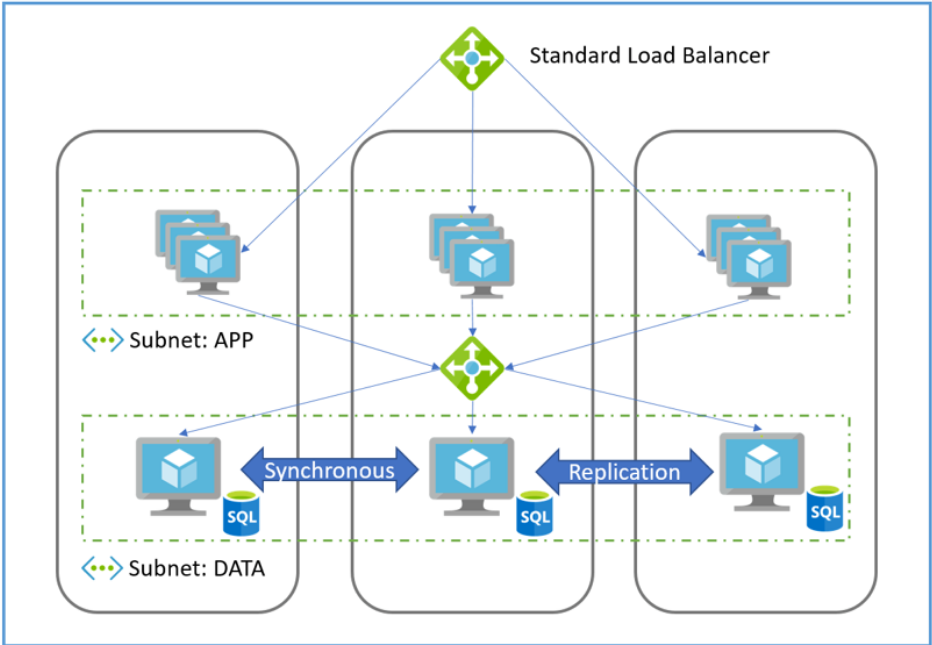
| **Subnet** | **CIDR** | **Addresses** | **Use** |
| --- | --- | --- | --- |
| DEV-FE-EUS2 | 10.245.16.0/22 | 1019 | Front-end/web tier VMs |
| DEV-APP-EUS2 | 10.245.20.0/22 | 1019 | App-tier VMs |
| DEV-DB-EUS2 | 10.245.24.0/23 | 507 | Database VMs |

**Setup Availability zones**

Availability zones increase high-availability to protect your apps and data from datacenter failures



* Zone-redundant services replicate your applications and data across availability zones to protect from single points of failure. - - With availability zones, Azure offers an SLA of 99.99% VM uptime
* You can plan and build high-availability into your migration architecture by colocating compute, storage, networking, and data resources within a zone, and replicating them in other zones. Azure services that support availability zones fall into two categories:
  + Zonal services: You associate a resource with a specific zone. For example VMs, managed disks, IP addresses).
  + Zone-redundant services: The resource replicates automatically across zones. For example, zone-redundant storage, Azure SQL Database.
* You can deploy a standard Azure load balanced with internet-facing workloads or app tiers, to provide zonal fault tolerance



**Hybrid Cloud Computing**

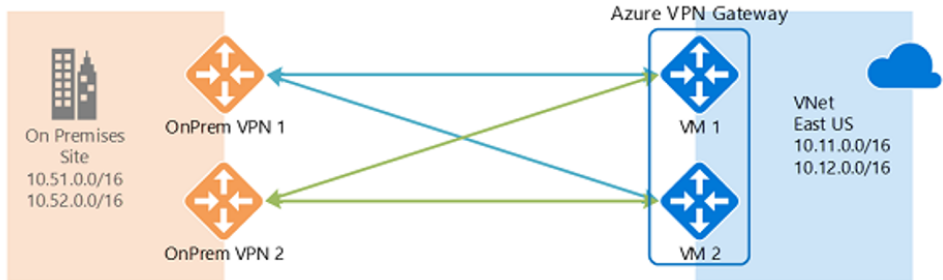
For a successful migration, it's critical to connect on-premises corporate networks to Azure. This creates an always-on connection known as a hybrid-cloud network, where services are provided from the Azure cloud to corporate users. There are two options for creating this type of network:

* **Site-to-site VPN**: You establish a site-to-site connection between your compatible on-premises VPN device and an Azure VPN gateway that's deployed in a VNet. Any authorized on-premises resource can access VNets. Site-to-site communications are sent through an encrypted tunnel over the internet.
* **Azure ExpressRoute**: You establish an Azure ExpressRoute connection between your on-premises network and Azure, through an ExpressRoute partner. This connection is private, and traffic doesn't go over the internet

**High Availability Site-to-Site VPN**

To implement a site-to-site VPN, you set up a VPN gateway in Azure.

* A VPN gateway is a specific type of VNet gateway that's used to send encrypted traffic between an Azure VNet and an on-premises location over the public Internet.
* You can also use a VPN gateway to send encrypted traffic between Azure VNets over the Microsoft network.
* Each VNet can have only one VPN gateway
* You need a VNet whose address range doesn't overlap with the on-premises network to which the VPN will connect.
* You create a gateway subnet in the network.
* You create a VPN gateway, specify the gateway type (VPN) and whether the gateway is policy-based or route-based. A Route Based VPN is recommended as more capable and future-proof.
* You create local network gateway on-premises, and configure your on-premises VPN device.
* You create a failover site-to-site VPN connection between the VNet gateway and the on-premises device. Using route-based VPN allows for either active-passive or active-active connections to Azure. Route-based also supports both site-to-site (from any computer) and point-to-site (from a single computer) connections concurrently.
* Border gateway protocol (BGP) is an optional feature you can use with Azure ExpressRoute and route-based VPN gateways to propagate your on-premises BGP routes to your VNets.



**Express Route Routing**

The Azure ExpressRoute service extends your on-premises infrastructure into the Microsoft cloud by creating private connections between the virtual Azure datacenter and on-premises networks

it's common to initially use a site-to-site VPN to establish connectivity between the virtual datacenter and on-premises resources, and then migrate to an ExpressRoute connection when a physical interconnection with your service provider is established

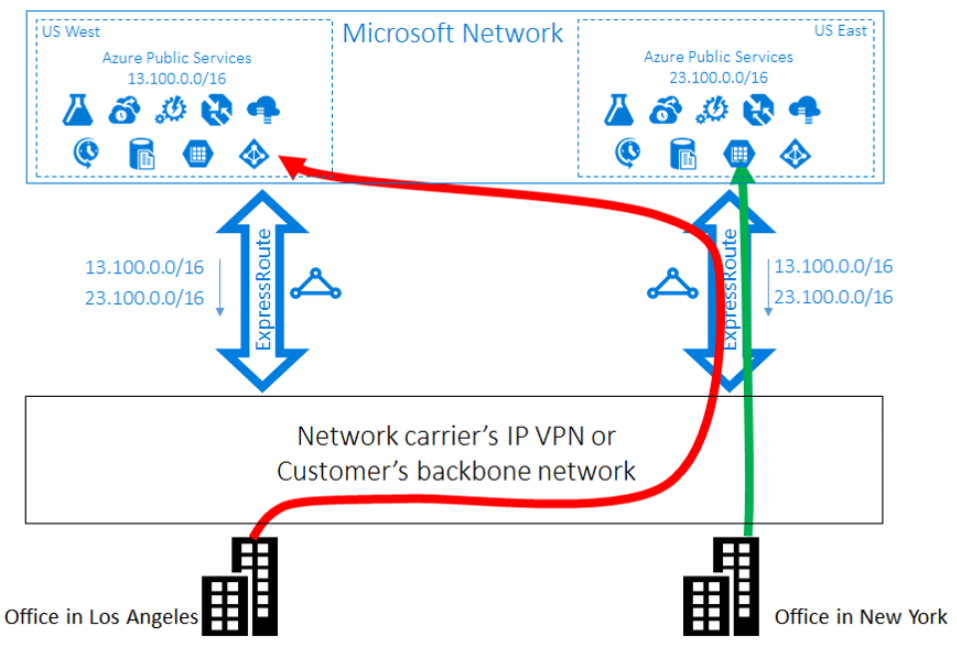
**Example**

* You have two offices in the US, one in Los Angeles and one in New York.
* Your offices are connected on a WAN, which can be either your own backbone network or your service provider's IP VPN.
* You have two ExpressRoute circuits, one in US West and one in US East, that are also connected on the WAN. Obviously, you have two paths to connect to the Microsoft network.

**Problem**:

Now imagine you have an Azure deployment (for example, Azure App Service) in both US West and US East.

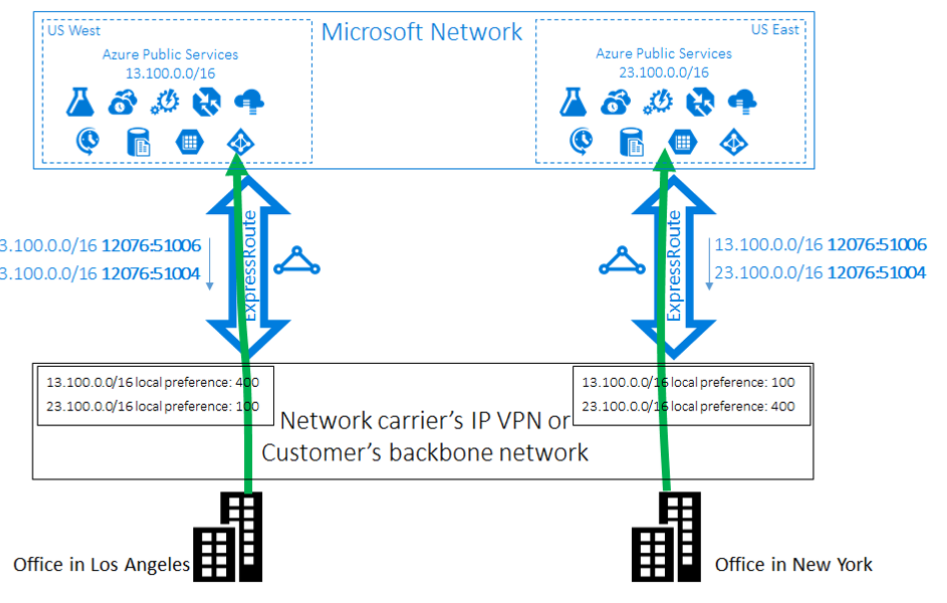
* You want users in each office to access their nearest Azure services for an optimal experience.
* Thus you want to connect users in Los Angeles to Azure US West and users in New York to Azure US East.
* This works for East Coast users, but not for those on the West Coast. The problem is:
  + On each ExpressRoute circuit, we advertise both prefixes in Azure US East (23.100.0.0/16) and Azure US West (13.100.0.0/16).
  + Without knowing which prefix is from which region, prefixes aren't treated differently.
  + Your WAN network can assume that both prefixes are closer to US East than US West, and thus route users from both offices to the ExpressRoute circuit in US East, providing a suboptimal experience for users in the Los Angeles office



**Solution**

To optimize routing for both office users, you need to know which prefix is from Azure US West and which is from Azure US East. You can encode this information by using BGP community values.

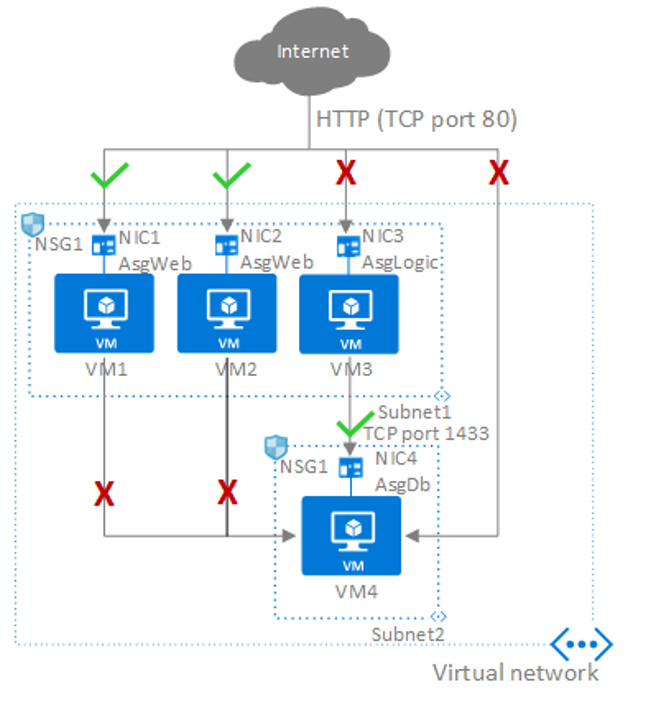
* You assign a unique BGP community value to each Azure region. For example, 12076:51004 for US East; 12076:51006 for US West.
* Now that it's clear which prefix belongs to which Azure region, you can configure a preferred ExpressRoute circuit.
* Because you're using BGP to exchange routing information, you can use BGP's local preference to influence routing.
* In our example, you assign a higher local preference value to 13.100.0.0/16 in US West than in US East, and similarly, a higher local preference value to 23.100.0.0/16 in US East than in US West.
* This configuration ensures that when both paths to Microsoft are available, users in Los Angeles will connect to Azure US West using the west circuit, and users New York connect to Azure US East using the east circuit. Routing is optimized on both sides



**Application security groups**

Application security groups enable you to configure network security as a natural extension of an app structure.

* You can group VMs and define network security policies based on application security groups.
* Application security groups enable you to reuse your security policy at scale without manual maintenance of explicit IP addresses.
* Application security groups handle the complexity of explicit IP addresses and multiple rule sets, allowing you to focus on your business logic.



**Example**

| **Network interface** | **Application security group** |
| --- | --- |
| NIC1 | AsgWeb |
| NIC2 | AsgWeb |
| NIC3 | AsgLogic |
| NIC4 | AsgDb |

* In our example, each network interface belongs to only one application security group, but in fact an interface can belong to multiple groups, in accordance with Azure limits.
* None of the network interfaces have an associated NSG. NSG1 is associated to both subnets and contains the following rules.

| **Rule name** | **Purpose** | **Details** |
| --- | --- | --- |
| Allow-HTTP-Inbound-Internet | Allow traffic from the internet to the web servers. Inbound traffic from the internet is denied by the DenyAllInbound default security rule, so no additional rule is needed for the AsgLogic or AsgDb application security groups. | Priority: 100  Source: internet  Source port: \*  Destination: AsgWeb  Destination port: 80  Protocol: TCP  Access: Allow. |
| Deny-Database-All | AllowVNetInBound default security rule allows all communication between resources in the same VNet, this rule is needed to deny traffic from all resources. | Priority: 120  Source: \*  Source port: \*  Destination: AsgDb  Destination port: 1433  Protocol: All  Access: Deny. |
| Allow-Database-BusinessLogic | Allow traffic from the AsgLogic application security group to the AsgDb application security group. The priority for this rule is higher than the Deny-Database-All rule, and is processed before that rule, so traffic from the AsgLogic application security group is allowed, and all other traffic is blocked. | Priority: 110  Source: AsgLogic  Source port: \*  Destination: AsgDb  Destination port: 1433  Protocol: TCP  Access: Allow. |