Module 1: Introduction to Azure Virtual Networks

## Chapter 1: Exploring VNETs

* VNETs are the building blocks of a private network
  + It has its own CIDR and can be linked with other VNETs or On-Prem
  + It enables resources in Azure to communicate with each other/Internet/On-Prem

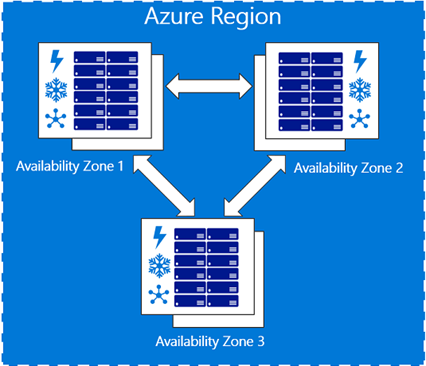
### **Capabilities of VNETs**:

* Communicate w/ Internet
  + Outbound to Internet by default
  + Use Public IP or Load Balancer for Inbound/Outbound
* Communicate w/ Azure Resources
  + **3 Key Ways to Comm.**
    - VNETs
      * Contained services can communicate directly
        + **Ex**. VM, ASE, Kubernetes, VMSS
    - VNET Service Endpoint
      * **Ex**. SQL DB, SA
    - VNET peering
* Comm w/ On-Prem resources
  + Use P2S VPN, S2S VPN, ExpressRoute
* Filter Traffic between subnets, peered VNETs using:
  + Firewall, Gateways, Proxy, NAT Services
* Routing
  + Use route tables, BGP routes

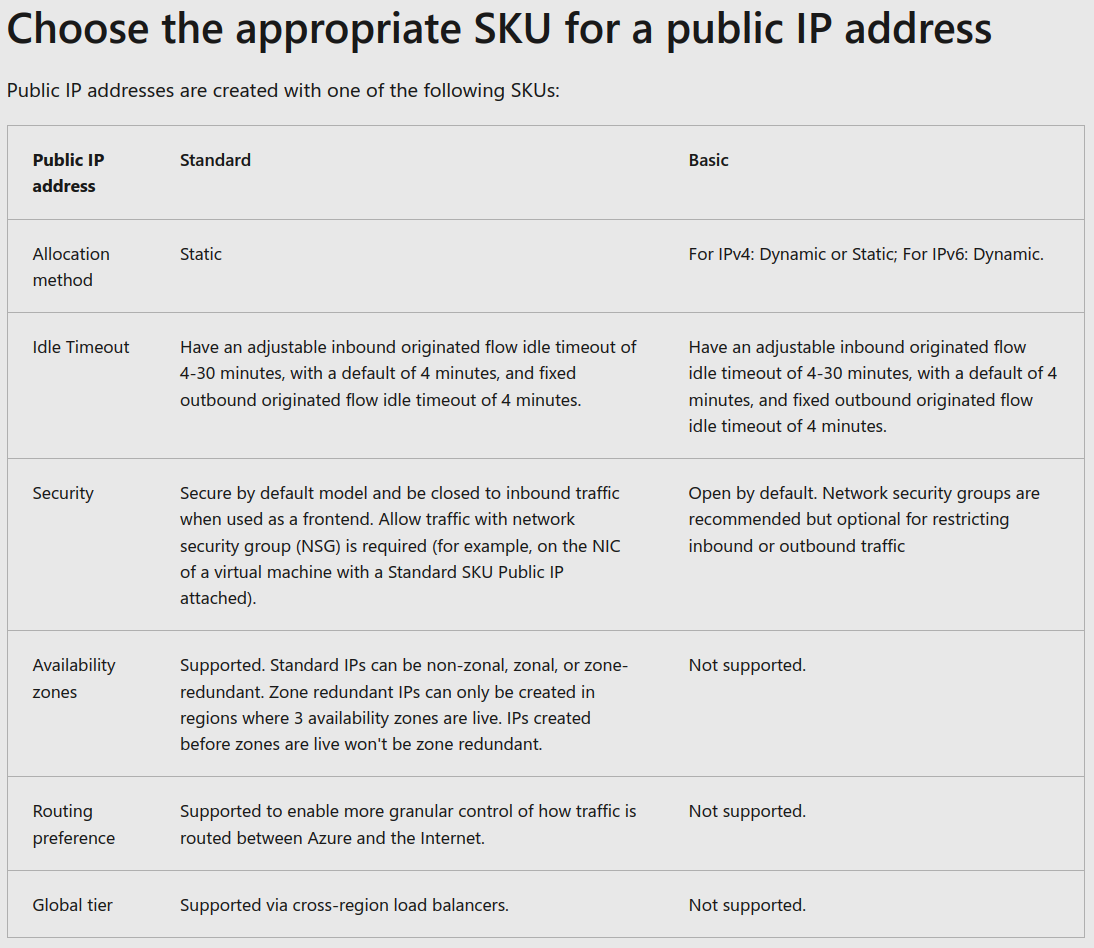
### Design Considerations

* **Virtual Networks** 
  + Recommended to use the address ranges enumerated in RFC 1918
    - **Set aside** by the IETF for private, non-routable address spaces:
      * 10.0.0.0 - 10.255.255.255 (10/8 prefix)
      * 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
      * 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
    - **Can't add** the following address ranges:
      * 224.0.0.0/4 (Multicast)
      * 255.255.255.255/32 (Broadcast)
      * 127.0.0.0/8 (Loopback)
      * 169.254.0.0/16 (Link-local)
      * 168.63.129.16/32 (Internal DNS)
  + **Resources like VMs are auto assigned a private IP based on subnet**
    - Azure reserves the first four and last IP address for a total of 5 IP addresses within each subnet. These are x.x.x.0-x.x.x.3 and the last address of the subnet.
  + **Consider**:
    - No-overlapping of address space
    - Is security isolation required?
    - Mitigate IP addressing limits
    - Connections between On-prem and Azure
    - Does the Azure Services create own VNET?
* **Subnets**
  + **Sizes**:
    - Smallest IPv4 subnet is /29, largest is /2
    - IPv6 subnets must be /64
  + **Consider**:
    - Subnets must have unique IP range
    - Some Az Resources need own subnet
    - Subnets can manage traffic
    - Limit access to subnets w/ Service Endpoint
* **Naming Conventions**
  + Since VNETs are RGP scoped:
    - **vnet-prod-westus-001** in each resource group.
  + Subnets are VNET scoped

### Regions and Subscriptions

* Resource can only be created in VNET of same region and subscription
  + But you can connect resources via VNET Peerings
* **Availability Zones**
  + 
  + **3 Types:**
    - Zonal (resource pinned to a specific zone)
      * ex. VMs, Managed Disks
    - Zone-Redundant
      * Resource is replicated across zones
    - Non-regional
      * Service is always available

## Chapter 2: Configure Public IPs

* Public IP addresses enable:
  + Internet/public-facing Azure services to communicate with Azure private resources
    - Each resources has its own public IP assigned by network engineer
    - Resources w/o public IP use NAT rules (Azure will dynamically give it a public IP)
  + **Example Resources**:
    - Virtual machine network interfaces
    - Virtual machine scale sets
    - Public Load Balancers
    - Virtual Network Gateways (VPN/ER)
    - NAT gateways
    - Application Gateways
    - Azure Firewall
    - Bastion Host
    - Route Server
* Types of Public IPs
  + **Dynamic**
    - Assigned address that can change over time and the unique pool is region based.
      * **Ex**. VM’s IP address changes when you start/stop the VM
  + **Static**
    - Assigned address won’t change over time. Only changes if resources is deleted
* **SKU for Public IP address resource**
  + 

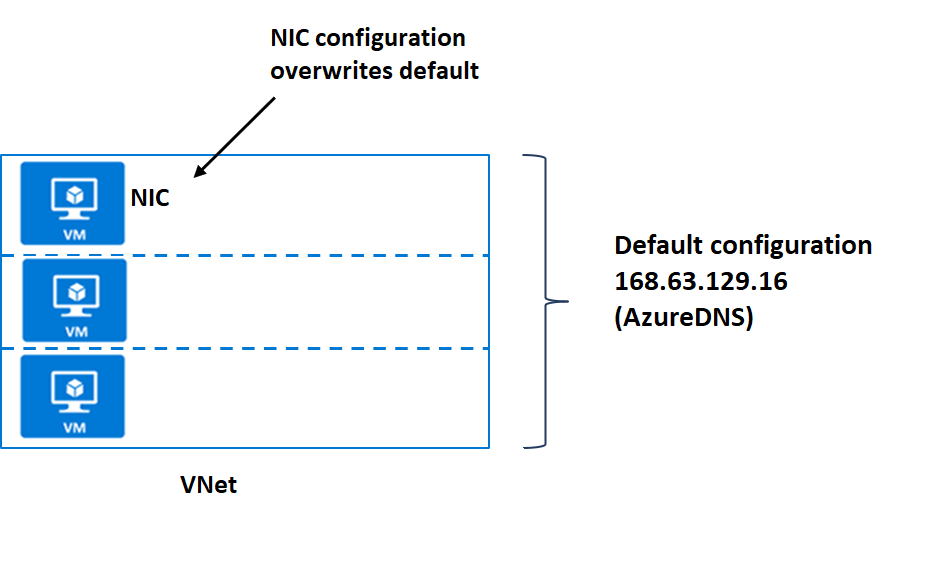
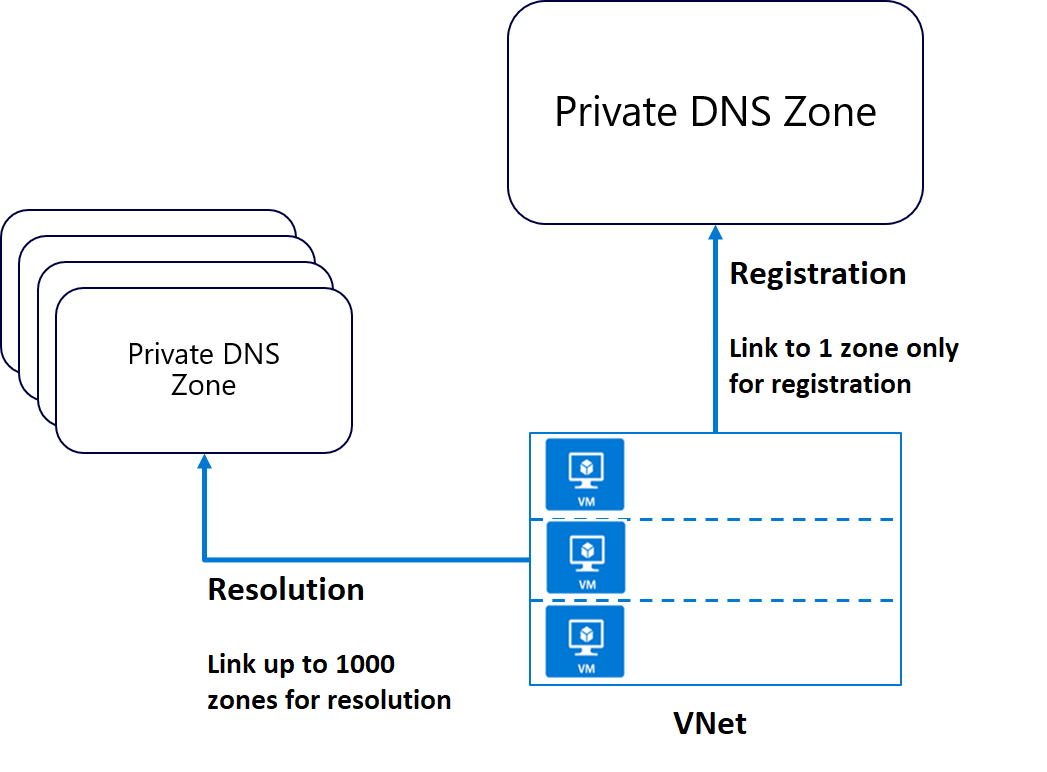
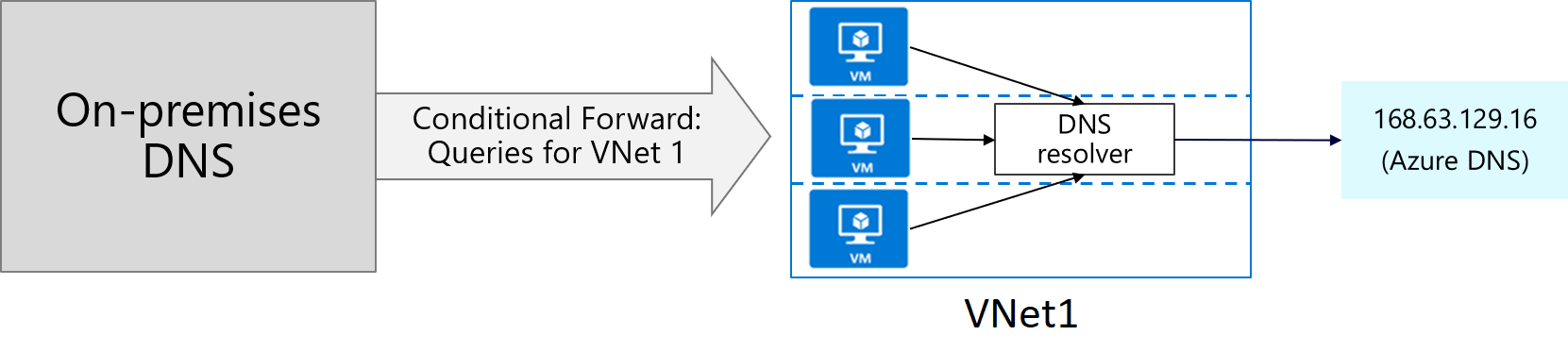
## Chapter 5: Design name resolution for VNET

* **DNS has 2 types**: (domain name)
  + Public & Private DNS for resources accessible from your own internal networks
* Azure DNS service hosts the DNS domains/records for your services
  + Create address records manually within relevant zones
    - Host records: A/AAAA (IPv4/IPv6)
    - Alias records: CNAME
* DNS zone hosts the DNS records for a domain
* **Considerations when creating DNS Zone**
  + Zone must be unique within RGP, but same zone name can be reused in a different RGP/subscriptions.
  + 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.

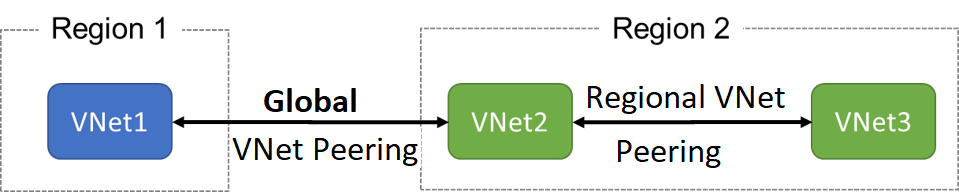
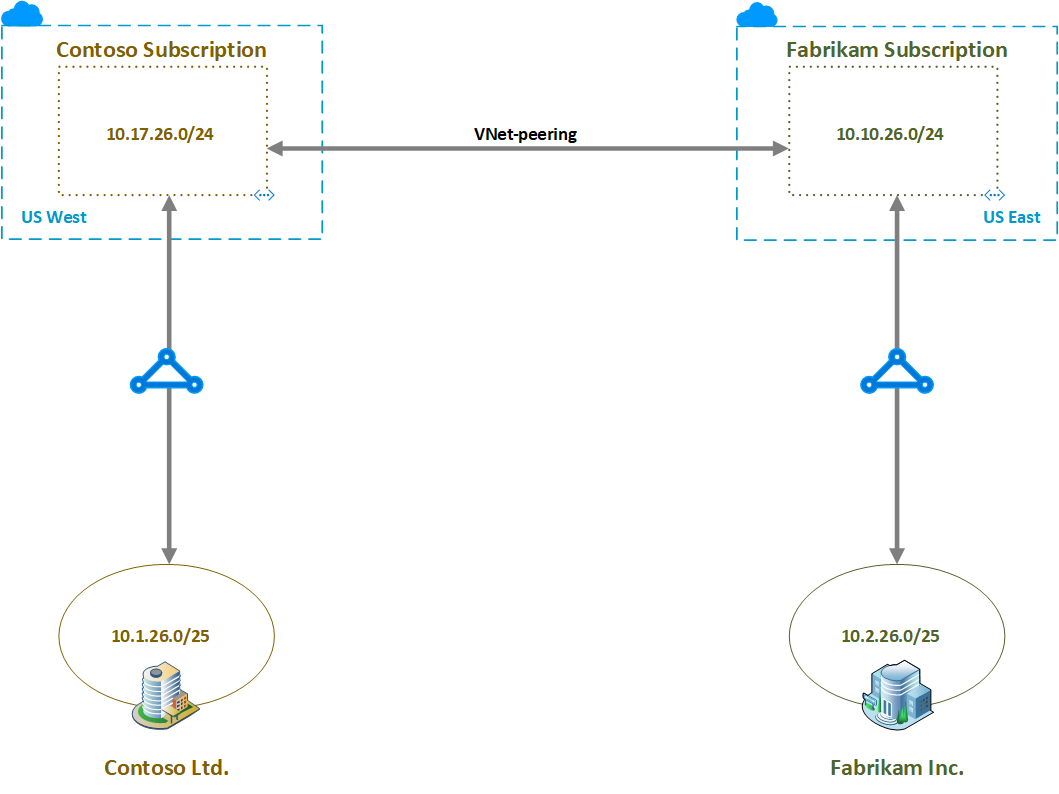
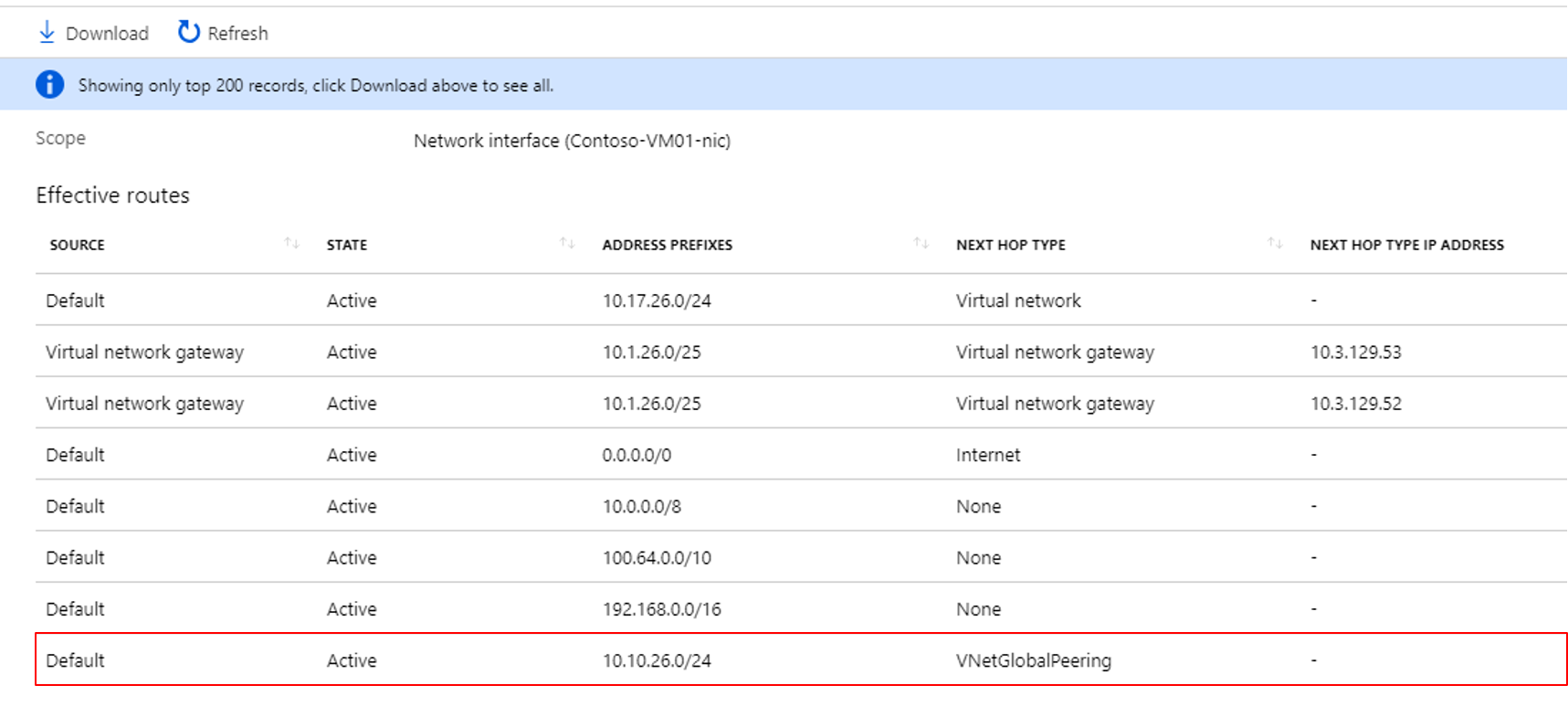
### Private DNS

* When resources deployed in virtual networks need to resolve domain names to internal IP addresses, they can use one the three methods:
  + **Azure-provided name resolution**
    - An internal DNS zone that always exists, supports automatic registration, requires no manual record creation, and is created when the VNET is created.
      * **Ex**. Internal DNS defines a namespace as follows: *.internal.cloudapp.net.*
        + Any VM created in the VNET will use this namespace
    - **Limitations of Internal DNS**
      * Can't resolve across different VNets.
      * Registers resource names, not guest OS names.
      * Does not allow manual record creation.
  + **Azure DNS Private Zones**
    - Only available to internal resources & accessible globally (any region, subscription, VNET, tenant)
      * Allows for more flexibility. Let’s you:
        + 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.
      * **When the new DNS zone is deployed**
        + Manually create resource records, or use auto-registration (create resource records based on the Azure resource name).

Support the full range of records including pointers, MX, SOA, service, and text records.

* + - * **Linking VNETS to Private DNS**
        + If necessary, you can override the default configuration by configuring an alternate DNS server at the VM NIC.
        + 
      * **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.
      * VNet is linked to a private DNS zone for registration and up to 100 private DNS zones for resolution.
        + 
  + **Name resolution that uses your own DNS server**
    - Companies use an internal Azure private DNS zone for auto registration, and then use a custom configuration to forward queries external zones from an external DNS server.
      * Forwarding takes two forms:
        + **Forwarding** - specifies another DNS server (SOA for a zone) to resolve the query if the initial server cannot.
        + **Conditional forwarding** - specifies a DNS server for a named zone, so that all queries for that zone are routed to the specified DNS server.
        + 

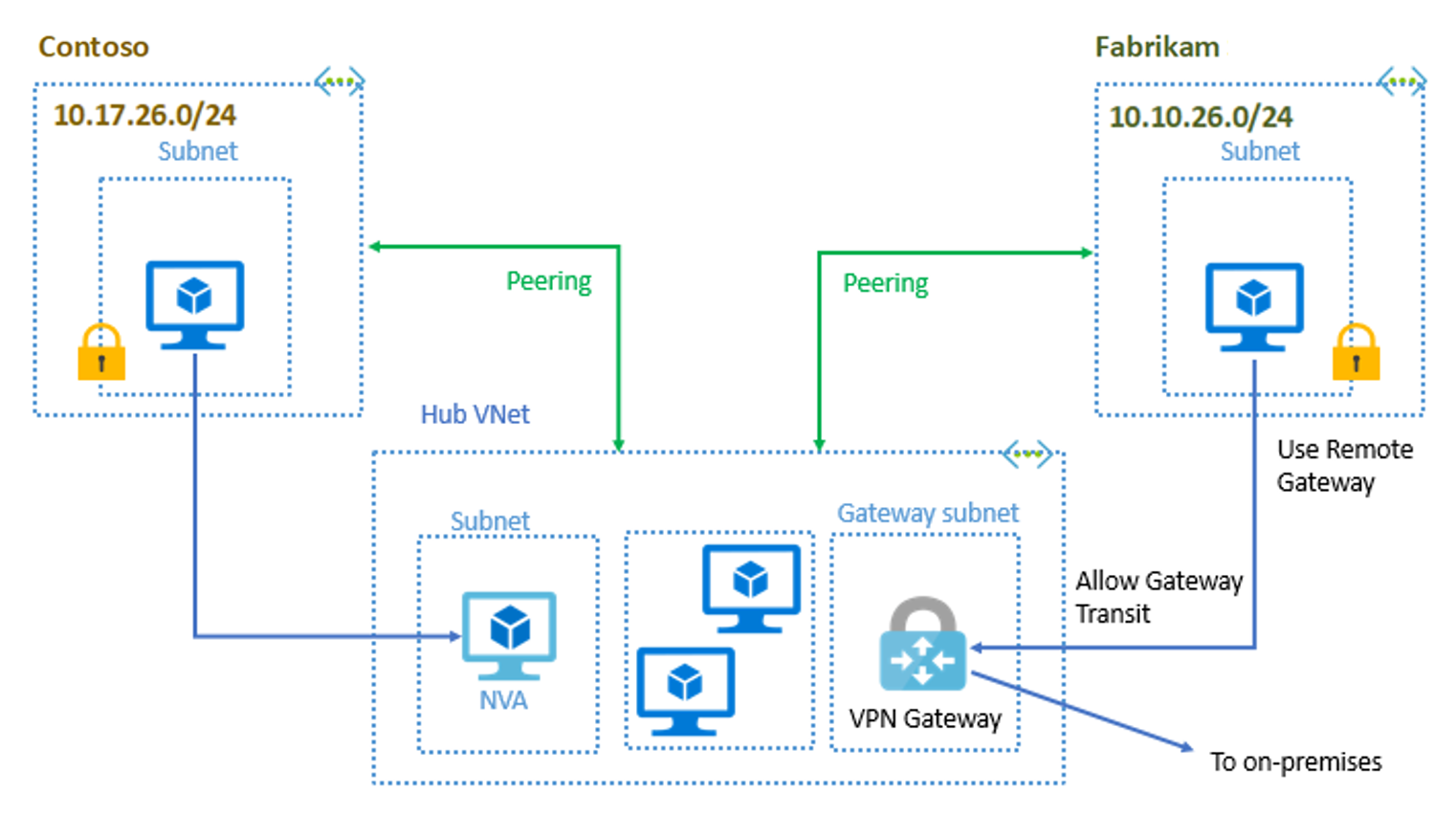
## Chapter 7: Enable cross-virtual network connectivity w/ peering

* Virtual network peering enables you to connect separate VNETs privately over the MS backbone
  + In same Azure region (**Regional** VNET peering)
  + In different regions (**Global** VNET peering).
    - Peered virtual networks can exist in any Azure public cloud region or China cloud regions, but NOT w/ Gov’t cloud regions. ONLY pair gov’t cloud regions with each other
    - 
* **Benefits of VNET Peering**
  + **Low-latency, high-bandwidth** connection between resources in diff VNETs.
  + Apply NSGs in either virtual network to **control access to other virtual networks** or subnets.
  + **Transfer data between VNETs** across Azure subscriptions, Azure Active Directory tenants, deployment models, and Azure regions.
  + Peer VNETs created through ARM.
  + Peer a VNET created through ARM w/ one created through the classic deployment model.
  + **No downtime** to resources when creating the peering, or after the peering is created.
  + **Example**:
    - 
    - Routing tables (RTs) show the routes known to the resources in each subscription.
      * When peered, RT shows the routes known to Contoso + final entry being *Global VNet peering entry to the Fabrikam 10.10.26.0/24 subnet*.
      * 
      * And Vice-versa (**when you create 1 peering, the other is auto-created too!**)

### Gateway Transit and Connectivity

* When peering VNETs, you can configure 1 VPN gateway (max) in the peered virtual network as a transit point.
  + Used for the peered VNET to access other resources (regionally and globally) outside the peering.
  + Examples of subnet gateways
    - Site-to-site (S2S) VPN to connect to an on-premises network.
    - VNET-to-VNET connection to another virtual network.
    - Point-to-site (P2S) VPN to connect to a client.
  + Peered VNETs can share this gateway
    - *\*\*\*NSGS can be applied in either virtual network to block access to other virtual networks or subnets.\*\*\**

### Service Chaining (directs traffic to gateway)

* User-defined routes (UDRs) can be used to direct traffic from a VNET to a specific Network Virtual Appliance (NVA)
  + AKA “Service Chaining”
  + **Enable service chaining by:**
    - Add user-defined routes pointing to VM in the peered virtual network as the next hop IP address.
    - User-defined routes can also point to virtual network gateways.
* Hub-and-Spoke Topology
  + Hub virtual network hosts infrastructure components such as an NVA, virtual machines and a VPN gateway
    - Other VNETs (Spokes) are peered to it:
      * 

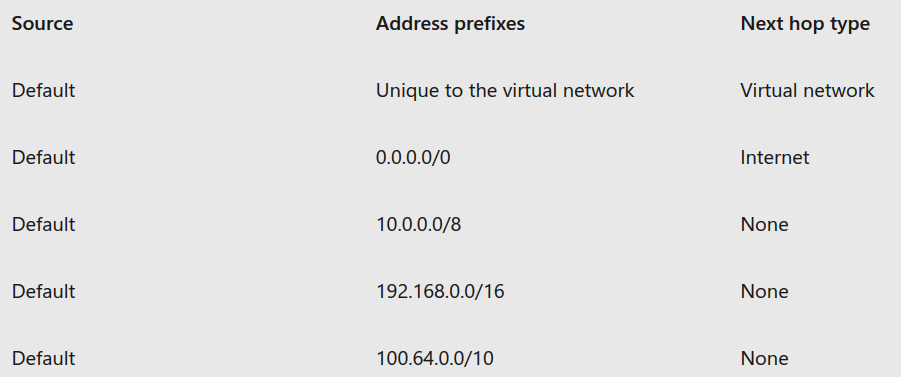
## Chapter 9: Implement VNET Traffic Routing

* Azure routes outbound traffic from a subnet based on the routes in a subnet's route table.

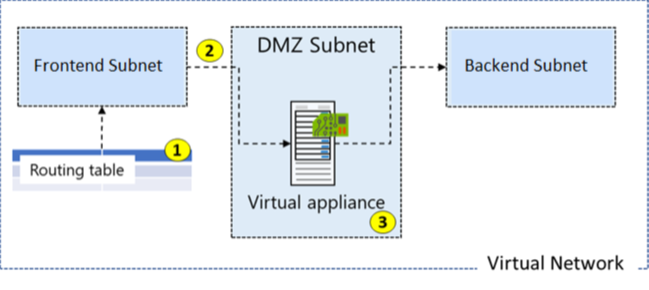
### System routes

* Azure automatically creates system routes and assigns the routes to each subnet in a virtual network.
  + Can’t create/remove these, but you can override them with Custom Routes

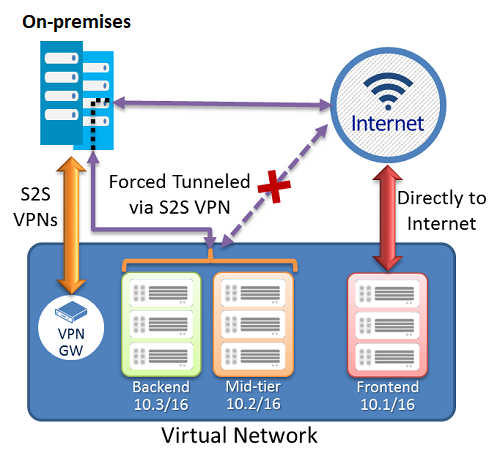
### Default routes

* Each route contains an address prefix and next hop type.
  + When traffic leaving a subnet is sent to an IP address within the address prefix of a route, the route that contains the prefix is the route Azure uses.
    - Whenever a virtual network is created, Azure automatically creates the following default system routes for each subnet within the virtual network:
      * 
* A **hop** is a way-point on the overall route. Therefore, the next hop is the next way-point that the traffic is directed to on its journey to its ultimate destination.
  + **Next hop types are:**
    - *VNET* - Routes traffic between address ranges within the address space of **a virtual network**.
    - Azure automatically routes traffic **between subnets** using the routes created for each address range
  + *Internet* - Routes traffic specified by the address prefix to the Internet
  + *None* - Traffic routed to the None next hop type is dropped
  + *VNET Peering* – Route auto-created to hop to other VNET
  + *VNET Gateway* -
    - If on-premises network gateway exchanges (BGP) routes with an Azure virtual network gateway, a route is added for each route propagated from the on-premises network gateway.
  + *VirtualNetworkServiceEndpoint* - Azure adds the public IP addresses for certain services to the route table when you **enable a service endpoint to the service**

### Custom Routes

* Useful when you want to ensure that traffic between two subnets passes:
  + Through a firewall appliance
  + Ensure that no traffic from a VNet could be routed to the internet.
* **UDRs** – each subnet has 0/1 RT associated with it
  + Possible **next hop types** when creating a UDR:
    - *Virtual appliance*: Virtual appliance === virtual machine (typically it runs a network application, such as a firewall)
      * When you create a route with the virtual appliance hop type, specify a next hop IP address.
        + Private IP address of a NIC (network interface) attached to a VM.
        + **OR** The private IP address of an Azure internal load balancer (AILB).
    - *Virtual network gateway -* The VNET gateway must be created with type VPN.
    - *None*
    - *VNET* - Specify when you want to override the default routing within a virtual network.
    - *Internet*
      * Specify explicitly route traffic to the Internet, or if you want traffic destined for Azure services with public IP addresses within the Azure backbone network.
  + Config UDRs
    - **Ex**. VNET w/ 3 subnets.
    - The subnets are Frontend, DMZ, and Backend.
      * In the DMZ subnet, there is a NVA.
      * Ensure all traffic from *Frontend* goes through NVA to *Backend*
      * 
      * Routes are automatically added to the route table for all subnets with **VNET gateway propagation** enabled. Useful for ExpressRoute so all subnets get routing info

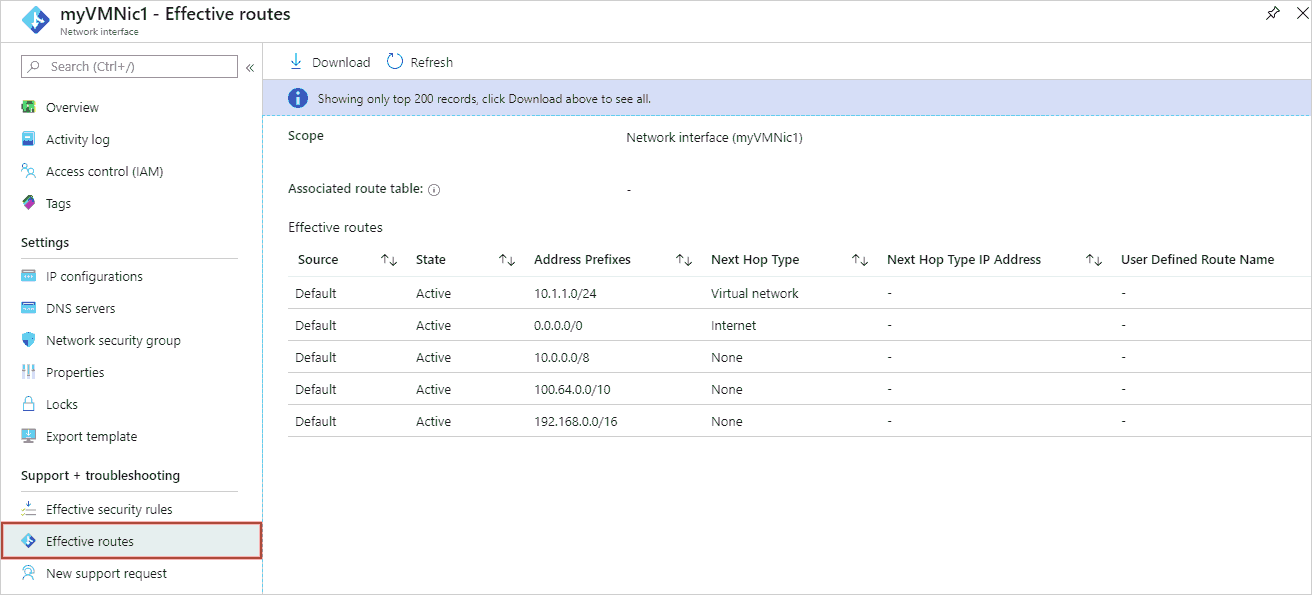
### Secure VNET with Forced Tunneling

* **Forced tunnelling**
  + Redirect or "force" all Internet-bound traffic back to your on-premises location via a Site-to-Site VPN tunnel
    - W/o forced tunnelling, Internet-bound traffic from your Azure VMs **always traverses** from the Azure Backbone to the Internet, without the option to allow you audit/inspect traffic
    - **Example**:
      * 
      * Backend/mid-tier traffic shouldn’t go to internet
  + **Configure Forced Tunnelling by**:
    - Using virtual network custom user-defined routes.
    - **How**:
      * Create a routing table.
      * Add a user-defined default route to the VPN Gateway.
      * Associate the routing table to the appropriate VNet subnet(s).
    - Forced tunnelling must be associated with a VNet that has a route-based VPN gateway.
      * You must set a default site connection among the cross-premises local sites connected to the virtual network.
      * The on-premises VPN device must be configured using 0.0.0.0/0 as traffic selectors.

## Configure Azure Route Server

* Route Server simplifies dynamic routing between your network virtual appliance (NVA) and your virtual network.
  + Allows exchange of routing information directly through (BGP) routing protocol
    - Between any supported NVA and the Azure Software Defined Network (SDN) in the Azure Virtual Network (VNET) automatically
* It simplifies configuration, management, and deployment of your NVA in your virtual network.
  + No need to update RT of NVA
  + No need to update UDRs when NVA changes
  + Peer multiple instances of NVA w/ Azure Route Server
  + Interface between NVA and Azure Route Server is based on a common BGP protocol

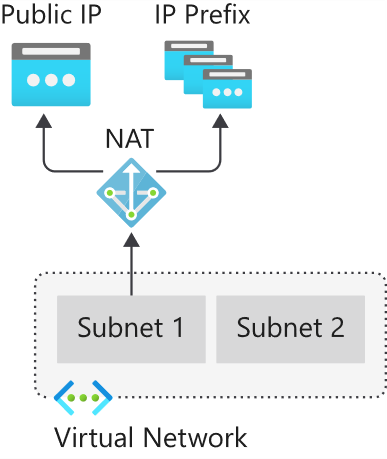
### Viewing Effective Routes:

* In Portal:
  + 
* In Powershell
  + Get-AzEffectiveRouteTable -NetworkInterfaceName myVMNic1 -ResourceGroupName myResourceGroup

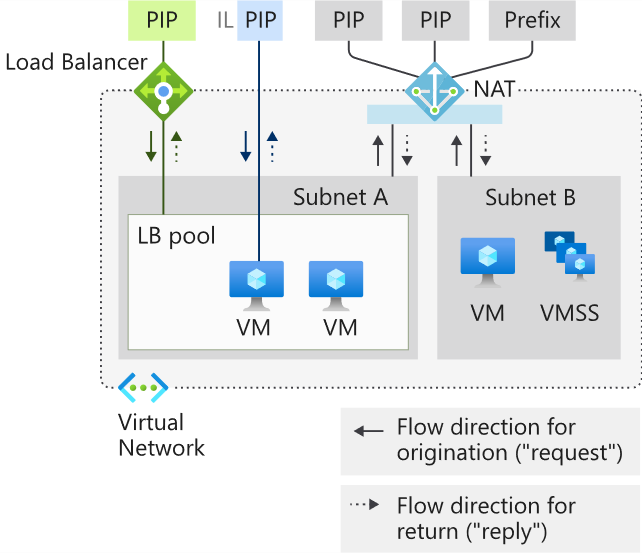
### Resolving Routing Issue Steps

* Add a custom route to override a default route.
* Change or remove a custom route that causes traffic to be routed to an undesired location.
* Ensure that the route table is associated to the correct subnet
* Ensure that devices (Azure VPN gateway or NVA) are operating as intended.

## Chapter 10: Configure internet access with Azure Virtual NAT

* Use a NAT service to map outgoing requests from internal resources to an external IP address, so that communication can take place.
  + *Example of outbound traffic flow from Subnet1 through the NAT gateway to be mapped to a Public IP address or a Public IP prefix*.
    - 

### Support dynamic workloads by scaling NAT

* With NAT, no need for extensive pre-planning or pre-allocate addresses because NAT scales to support dynamic workloads.
  + Using port network address translation (PNAT or PAT), 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 can be used together w/ LB, Public IP Address/Prefix to provide inbound Internet connectivity to your subnet(s).
  + - 
* **Limitations of NAT**
  + NAT is compatible with standard SKU public IP, public IP prefix, and load balancer resources.
  + IPv6 address aren’t supported, ONLY IPv4 address family
  + NAT can't span multiple virtual networks.
  + IP fragmentation isn't supported.