AZ-700 Master Cheat Sheet

Design and implement core networking infrastructure (25–30%)

Design and implement IP addressing for Azure resources

Plan and Implement Network Segmentation and Address Spaces

Network segmentation is a crucial security practice that divides your Azure virtual network (VNet) into smaller subnets. Each subnet acts as a separate network segment, isolating resources based on function or security requirements. This approach offers several benefits:

- **Enhanced Security:** By isolating workloads, a security breach in one subnet is less likely to affect others.
- **Improved Traffic Management:** Segmenting networks allows for optimized traffic flow by directing communication within specific subnets.
- **Granular Access Control:** Security groups can be applied to subnets, defining which resources can communicate with others.

Planning your address spaces involves selecting appropriate private and public IP address ranges for your VNets and subnets. Here are some key considerations:

- **Private Address Space:** Choose a private address range from the RFC 1918 space (e.g., 10.0.0.0/16, 172.16.0.0/12, 192.168.0.0/16) to accommodate your resources within the VNet.
- **Public IP Address Prefixes (Optional):** If your resources require internet accessibility, plan a public IP address prefix from Azure. This allows assigning public IP addresses to resources within the subnet for inbound connections.

Create a Virtual Network (VNet)

A VNet acts as the foundation for private IP addressing in Azure. It represents a logically isolated network within the Azure cloud dedicated to your resources. When creating a VNet, you define:

- **Resource Group:** A logical container to organize Azure resources.
- **Name:** A unique identifier for your VNet.
- **Location:** The Azure region where the VNet resides.
- **Address Space:** The private IP address range for your VNet.
- **Subnet Configuration (Optional):** You can define subnets during VNet creation or add them later.

Plan and Configure Subnetting for Services

Subnetting involves dividing your VNet's address space into smaller, more manageable networks called subnets. Each subnet can house specific resources requiring private IP addresses for communication within the VNet. Here's how subnetting is crucial for different Azure services:

- **VNet Gateways:** Subnets for internet connectivity (public gateway) and on-premises connectivity (VPN gateway) may be required.
- **Private Endpoints:** Subnets for private access to Azure services without traversing the internet can be configured.
- **Service Endpoints:** Subnets can be linked to specific Azure services for secure, direct communication.
- **Firewalls, Application Gateways:** Dedicated subnets may be needed to deploy these network security and traffic management solutions.
- **VNet-Integrated Platform Services:** Subnets can be designated for Azure services like Azure App Service or Azure SQL Database deployed within the VNet.
- **Azure Bastion:** A dedicated subnet is required for the Azure Bastion service, which provides secure RDP/SSH access to VMs in a private VNet.

Additional Considerations:

- **Subnet Delegation:** This feature simplifies resource deployment within a subnet by automatically assigning IP addresses from a designated range to resources upon creation.
- **Shared vs. Dedicated Subnets:** Decide if a subnet will host multiple service types or be dedicated to a specific function.

Plan and configure subnet delegation:

- **Subnet delegation:** This feature allows you to delegate Azure resource providers the authority to manage IP addresses within a specific subnet. This simplifies network management by letting services like Azure App Service or Azure Kubernetes Service (AKS) automatically assign IP addresses to their resources within the subnet.
- **Planning considerations:**
  - **Resource providers:** Identify which Azure resource providers need delegation (e.g., App Service, AKS).
  - **Security groups:** Ensure appropriate security groups are applied to control inbound and outbound traffic for delegated resources.
  - **Monitoring:** Monitor resource utilization within the subnet to avoid potential IP address exhaustion.

Plan and configure shared or dedicated subnets:

- **Shared subnets:** A single subnet can be shared by multiple types of Azure resources. This approach is efficient for workloads with similar security requirements.
- **Dedicated subnets:** Each type of resource or workload can have its own dedicated subnet. This enhances security by isolating traffic between different workloads.
- **Planning considerations:**
  - **Security:** Evaluate security needs for your workloads. Dedicated subnets offer greater isolation.

   - **Scalability:** Consider future growth and choose a subnet size that can accommodate additional resources.
   - **Management:** Dedicated subnets require more management overhead compared to shared subnets.

Create a prefix for public IP addresses:

- **Public IP address prefix:** A range of public IP addresses used to assign public IPs to Azure resources like VMs or web apps. This allows internet access to these resources.
- **Creating a prefix:** You can define a public IP address prefix during virtual network creation or allocate one separately.
- **Planning considerations:**
   - **Size:** Determine the number of public IP addresses you need based on your resource requirements.
   - **Static vs. Dynamic allocation:** Choose static allocation for specific resources or dynamic allocation for flexible deployments with Azure Resource Manager templates.

Choose when to use a public IP address prefix:

- **Public IP addresses are needed for:**
   - Resources requiring internet access (e.g., web servers, VPN gateways).
   - Accessing resources remotely via RDP or SSH.
- **Alternatives to public IP addresses:**
   - Azure Load Balancer can distribute traffic across VMs with private IP addresses.
   - Azure Private Link allows secure access to Azure services without exposing them to the internet.
- **Planning considerations:**
   - **Security:** Evaluate if public exposure is necessary for your resources. Consider alternatives for increased security.
   - **Cost:** Public IP addresses incur additional charges. Choose the most cost-effective access method.

Plan and implement a custom public IP address prefix (bring your own IP - BYOI):

- **BYOI Overview:** In Azure, you can use your own public IP address ranges for resources. This is beneficial if you have existing IP allocations you want to leverage or need specific IP ranges for compliance reasons.
- **Planning Considerations:**
   - **IP Ownership:** Ensure you have legal rights to the IP range you want to bring.

  - **Region Availability:** BYOI IPs are not available in all Azure regions. Check for compatibility before planning.
  - **Prefix Size:** Request a prefix large enough for your future needs.
  - **Validation:** Azure validates the ownership and routing information for your BYOI prefix.

- **Implementation Steps:**

1. Contact Microsoft support to initiate the BYOI process.
2. Provide ownership and routing information for your IP prefix.
3. Once approved, create a virtual network (VNet) and associate the BYOI prefix during creation.

Create a public IP address:

- **Public IP Overview:** A public IP address allows internet access for Azure resources like virtual machines or web apps.
- **Creation Methods:**
  - **Azure portal:** Select the desired resource and configure a public IP address during creation or allocation.
  - **Azure Resource Manager (ARM) templates:** Define public IP resources within your ARM template for infrastructure as code deployments.
  - **Azure CLI or PowerShell:** Use commands to create public IP resources programmatically.
- **Allocation Types:**
  - **Basic:** Ephemeral public IP, dynamically assigned by Azure upon resource creation. Useful for temporary deployments.
  - **Standard:** Static public IP with a fixed address. Ideal for production environments or long-running resources.

Associate public IP addresses to resources:

- **Association Process:** After creating a public IP address, you can associate it with various resources.
- **Supported Resources:**
  - Virtual Machines
  - Cloud Services (classic model)
  - App Service environments
  - Azure Kubernetes Service (AKS) nodes (with limitations)
  - Bastion hosts

- **Association Methods:**
  - **Azure portal:** During resource creation or through the resource's configuration blade.
  - **ARM templates:** Define the public IP association within your template for infrastructure as code.
  - **Azure CLI or PowerShell:** Use commands to associate public IPs with resources.

Additional Tips:

- Public IP addresses incur charges. Choose the allocation type (basic or standard) based on your resource's needs.
- Consider using Azure Load Balancer to distribute traffic across multiple resources with a single public IP for scalability and redundancy.
- For enhanced security, use Network Security Groups (NSGs) to control inbound and outbound traffic for resources with public IPs.

Design and implement name resolution

Design Name Resolution Inside a VNet:

By default, Azure provides a built-in DNS service for VNets. This service, known as the Azure-provided DNS, resolves names ending with the suffix .internal.cloudapp.net for resources within the VNet. However, you can also:

- **Use Custom DNS Servers:** You can configure your VNet to use external DNS servers hosted on-premises or by another provider. This offers more control over name resolution policies.
- **Link a Private DNS Zone:** Integrate a private DNS zone (explained later) with your VNet for more granular control over internal domain names.

Configure DNS Settings for a VNet:

When creating a VNet in Azure, you can specify the DNS server settings. This includes:

- **Name Servers:** Define the IP addresses of the DNS servers your VNet should use for name resolution. This can be the Azure-provided DNS, custom DNS servers, or a combination.
- **DNS Search Suffix:** This optional setting specifies the domain suffix that should be appended to unqualified names (those without a dot-separated suffix) during resolution attempts.

Design Public DNS Zones:

Public DNS zones are hosted in Azure DNS, a managed DNS service. They map public domain names (like "contoso.com") to public IP addresses of internet-facing resources like websites or applications. Designing a public DNS zone involves:

- **Planning the Zone Name:** Choose a unique and relevant domain name for your organization or application.

- **Record Types:** Define various record types like A records (maps domain names to IP addresses), CNAME records (aliases for other domains), and MX records (for mail servers).
- **Security Considerations:** Implement security measures like DNSSEC (Domain Name System Security Extensions) to protect your zone from spoofing attacks.

Design Private DNS Zones:

Private DNS zones, also hosted in Azure DNS, are used for internal name resolution within your organization's VNets. Unlike public zones, they are not accessible from the internet. Here's how to design a private zone:

- **Zone Naming:** Choose a descriptive name that reflects its purpose within your VNet.
- **Record Types:** Similar to public zones, define record types to map internal domain names to resources within the VNet (e.g., web servers, databases).
- **Integration with VNet:** Link the private zone with your VNet to enable resources within the VNet to resolve names hosted in the private zone.

Configure Public and Private DNS Zones:

- **Public DNS Zones:** These zones reside in Azure DNS and are accessible from the public internet. They allow you to manage DNS records for your custom domain names (e.g., contoso.com) and point them to Azure resources (websites, VMs) with public IP addresses.
- **Private DNS Zones:** These zones exist within your Virtual Network (VNet) and are not accessible from the internet. They provide internal name resolution for resources within the VNet, allowing VMs to communicate using FQDNs instead of IP addresses. You can either use:
    - **Azure-provided DNS:** This is the default option where Azure manages a hidden zone (.internal.cloudapp.net) for your VNet.
    - **Custom DNS Servers:** You can integrate your own DNS server solution within the VNet for more granular control over internal name resolution.

Link a Private DNS Zone to a VNet:

- To leverage a custom private DNS zone for internal name resolution in your VNet, you need to link them. This allows VMs within the VNet to automatically register their hostnames and IP addresses in the linked zone.

Design and Implement Azure DNS Private Resolver:

- Azure DNS Private Resolver acts as a managed DNS service for private DNS zones. It provides a single endpoint within your VNet for VMs to perform DNS queries. This simplifies configuration and eliminates the need to manage individual DNS servers for each VNet.

Here are some additional points to consider:

- **Record Types:** Both public and private zones allow you to configure various DNS record types like A records (map hostname to IP), CNAME records (alias for another hostname), and MX records (for email delivery).

- **Security:** Public DNS zones require proper access control to prevent unauthorized modifications. Private zones offer inherent security as they are not accessible from the internet.
- **Integration:** Azure DNS integrates with other Azure services like Azure Active Directory for secure zone management and Traffic Manager for load balancing across geographically distributed resources.

Additional Resources:

Microsoft documentation on Azure DNS: https://learn.microsoft.com/en-us/azure/dns/

Study guide for Exam AZ-700: https://learn.microsoft.com/en-us/training/courses/az-700t00

Design and implement VNet connectivity and routing

Service Chaining with Gateway Transit:

- **Concept:** Service chaining allows traffic to flow through a sequence of Azure network security services (firewalls, application gateways) before reaching its destination. Gateway transit simplifies this process by enabling a single VNet to act as a transit center for traffic flowing between other VNets and network security services.
- **Design Considerations:**
  - Identify the sequence of network security services the traffic needs to traverse.
  - Plan the location of the transit VNet for optimal performance.
  - Configure routing to direct traffic to the transit VNet and then the specific security services.
- **Benefits:**
  - Centralized security enforcement for multiple VNets.
  - Simplified management of network security policies.

VNet Peering:

- **Concept:** VNet peering directly connects VNets within the same region or across regions. It allows resources in peered VNets to communicate with each other using private IP addresses.
- **Implementation:**
  - Choose between regional or global peering based on VNet locations.
  - Configure peering connections in each VNet specifying the peer VNet's resource ID.
  - Optionally, configure route tables to control traffic flow between peered VNets.
- **Benefits:**

    - Low-latency, secure communication between VNets.
    - Simplifies network architecture for interconnected resources.

Azure Virtual Network Manager:

- **Concept:** Virtual Network Manager is a centralized tool for managing and deploying VNets at scale. It allows for defining network configurations as code and deploying them consistently across environments.
- **Implementation:**
    - Define VNet configurations using Azure Resource Manager (ARM) templates.
    - Utilize Virtual Network Manager to manage deployments, policies, and access control for VNets.
- **Benefits:**
    - Consistent and repeatable VNet deployments.
    - Simplified management of complex network configurations.
    - Improved governance and control over network resources.

User-Defined Routes (UDRs):

- **Concept:** UDRs provide granular control over traffic routing within a VNet. They define how packets are forwarded to specific subnets or next hops (gateways) based on destination IP addresses and prefixes.
- **Design Considerations:**
    - Identify the traffic flow patterns within the VNet.
    - Define UDRs to route traffic to the appropriate subnets or gateways.
    - Prioritize UDRs to ensure the most specific route takes precedence.
- **Benefits:**
    - Customized traffic flow management within a VNet.
    - Optimization of network performance for specific workloads.

Associating Route Table with Subnet:

- **Concept:** A route table defines the routing behavior for a subnet. A subnet can only be associated with one route table at a time. The route table dictates how traffic destined for addresses outside the subnet is forwarded.
- **Implementation:**
    - Create a route table with UDRs defining the desired traffic flow.
    - Associate the route table with the desired subnet in the VNet configuration.
- **Benefits:**

- Control traffic flow at the subnet level for specific workloads.
- Allows for differentiated routing policies within a VNet.

Forced Tunneling

- **Concept:** Forced tunneling ensures all traffic destined for the internet or on-premises networks traverses the Azure virtual network gateway (VPN gateway or ExpressRoute gateway). This mechanism prevents traffic from bypassing security controls within the VNet.
- **Configuration:**
    1. Create a route table with a default route pointing to the internet or on-premises network address space via the virtual network gateway's public IP address.
    2. Associate the route table with the desired subnets in your VNet.
- **Benefits:**
    - Enhanced security: Traffic is routed through firewalls and network security groups (NSGs) for inspection and control.
    - Centralized management: Policies are applied at the gateway level, simplifying administration.
- **Considerations:**
    - Increased latency: Traffic might take a longer path compared to direct egress.
    - Potential cost implications: Egress charges may apply depending on your pricing tier.

Diagnosing and Resolving Routing Issues

- **Troubleshooting Techniques:**
    1. **Verification:**
        - Confirm route table associations with subnets.
        - Validate route table entries (destination, next hop).
        - Check for overlapping address spaces.
    2. **Network Watcher:** Use Azure Network Watcher's tools like "Next hop" and "Effective route" to trace the path packets take and identify bottlenecks.
    3. **Logs:** Analyze diagnostic logs from virtual network gateways, VMs, and NSGs for routing-related events or errors.
- **Common Issues:**
    - Incorrect route table associations or entries.
    - Overlapping IP addresses causing conflicts.
    - Security group rules blocking traffic unexpectedly.

- Connectivity problems with the virtual network gateway.

Azure Route Server

- **Purpose:** A highly available, scalable routing solution for complex network topologies with multiple VNets, on-premises networks, and internet connectivity. It centralizes route management, simplifies configuration, and enhances routing performance.
- **Use Cases:**
  - Large-scale deployments with intricate routing requirements.
  - Scenarios requiring dynamic routing protocols like BGP for inter-network communication.
  - Need for centralized route advertisement and propagation across multiple VNets.
- **Benefits:**
  - Scalability: Handles high volumes of routes and traffic efficiently.
  - Centralized Management: Simplifies route configuration and updates across VNets.
  - Dynamic Routing: Supports protocols like BGP for optimal path selection.
  - High Availability: Redundancy ensures reliable route advertisement and traffic flow.
- **Implementation Considerations:**
  - Deployment complexity: Requires careful planning and configuration.
  - Licensing costs: Azure Route Server incurs additional charges.

Network Address Translation (NAT) Gateway

- **Function:** Enables outbound internet connectivity for resources within a private VNet that don't have public IP addresses assigned. The NAT gateway translates private IP addresses to a public IP address, allowing outbound communication without exposing individual VM addresses to the internet.
- **Use Cases:**
  - VNets with resources that need internet access without public IP addresses (e.g., internal databases).
  - Maintaining security by minimizing public exposure of resources.
  - Scenarios where public IP addresses are not cost-effective for all resources.
- **Implementation:**

1. Create a NAT gateway resource in your desired resource group and region.
2. Allocate a public IP address for the NAT gateway.
3. Associate the NAT gateway's subnet with a public IP address prefix.
4. Configure outbound security rules in the NAT gateway's NSG to allow required traffic.

- **Benefits:**
    - Secure outbound internet access: Resources remain private while having internet connectivity.
    - Centralized outbound control: Manage outbound traffic through the NAT gateway's security policies.
    - Cost optimization: Reduce costs associated with public IP addresses for all resources.

Additional Considerations:

- **UDRs (User-Defined Routes):** Provide granular control over traffic flow within a VNet. Use them to override default routes and direct traffic to specific destinations.
- **Service Chaining with Gateway Transit:** Configure a VNet gateway (VPN or ExpressRoute) to route traffic through another gateway (typically a NAT gateway) for security or network address translation purposes.
- **VNet Peering:** Directly connect VNets within the same region or across regions for private communication between resources.

Monitor networks

Configure monitoring, network diagnostics, next hop, and path visualization in Azure Network Watcher:

- **Network Watcher:** This service provides tools to monitor and diagnose your Azure virtual networks.
- **Monitoring:** You can configure Network Watcher to collect data on various network metrics like bandwidth utilization, packet latency, and connection status.
- **Network diagnostics:** This functionality allows you to troubleshoot network issues by performing tasks like:
    - **Next hop:** Identify the next hop for a specific destination IP address, helping you understand traffic routing.
    - **Path visualization:** Visually trace the network path taken by packets between resources, providing insights into potential bottlenecks.
- **Logs:** Network Watcher can also collect logs for network security groups (NSGs) and application security groups (ASGs), aiding in security analysis.

Monitor and troubleshoot network health by using Azure Network Watcher:

- Network Watcher offers various tools to proactively monitor and troubleshoot network health:
    - **Connectivity checks:** Verify connectivity between resources within a virtual network or between your Azure environment and on-premises locations.
    - **IP flow analysis:** Analyze network traffic patterns to identify anomalies and potential security threats.

  - **VM connection monitor:** Monitor network connectivity and performance for Azure VMs.

Monitor and troubleshoot networks by using Azure Monitor Network Insights:

- **Azure Monitor:** This service provides a central location for collecting, analyzing, and visualizing telemetry data from various Azure resources, including virtual networks.
- **Network Insights:** This is a feature within Azure Monitor that focuses specifically on network traffic data. It offers advanced capabilities for:
  - **Traffic analysis:** Analyze traffic patterns by source, destination, protocol, and application.
  - **Performance monitoring:** Monitor network performance metrics like latency, throughput, and packet loss.
  - **Troubleshooting tools:** Leverage tools like next hop analysis and path visualization similar to Network Watcher, but with potentially deeper insights from traffic data.

Activating and Monitoring DDoS Protection

Distributed Denial-of-Service (DDoS) attacks overwhelm your resources with traffic, making applications inaccessible. Azure offers DDoS Protection Standard and Premium tiers to safeguard your virtual networks. Here's how to activate and monitor them:

- **Activate DDoS Protection:**
  1. Navigate to your virtual network in the Azure portal.
  2. Go to the "Security" section and select "DDoS protection."
  3. Choose the appropriate tier (Standard or Premium) based on your needs.
  4. Configure DDoS protection settings like attack mitigation thresholds.
- **Monitor DDoS Protection:**
  1. Access the DDoS protection overview page for your virtual network.
  2. View metrics like incoming traffic volume, attack types detected, and mitigation actions taken.
  3. Utilize Azure Monitor logs to analyze detailed DDoS activity and identify potential threats.
  4. Configure alerts to receive notifications when DDoS attacks occur or mitigation thresholds are reached.

Evaluating Network Security Recommendations from Microsoft Defender for Cloud Secure Score

Microsoft Defender for Cloud provides a "Secure Score" that assesses your Azure environment's security posture. It includes recommendations for improving network security. Here's how to evaluate them:

- **Understanding Secure Score:**

1. Access the Microsoft Defender for Cloud portal.
2. Locate the "Secure Score" section to view your overall score and breakdown by security areas.

- **Evaluating Network Security Recommendations:**
    1. Navigate to the "Recommendations" section under "Secure Score."
    2. Filter by security control to focus on network security recommendations.
    3. Each recommendation details the potential security risk and recommended mitigation steps.
    4. Analyze the impact of each recommendation and prioritize them based on severity and risk.

By implementing these recommendations, you can strengthen your network security posture and reduce vulnerabilities.

Here are some additional resources that you might find helpful:

- **Azure DDoS Protection documentation:** https://learn.microsoft.com/en-us/azure/ddos-protection/
- **Microsoft Defender for Cloud Secure Score:** https://learn.microsoft.com/en-us/training/modules/examine-microsoft-secure-score/

Evaluating Network Security Recommendations from Attack Path Analysis

- **Microsoft Defender for Cloud Attack Path Analysis:** This is a feature that analyzes your Azure environment and identifies potential attack paths an attacker could exploit. It considers vulnerabilities, misconfigurations, and resource access controls to build a security graph.
- **Evaluating Recommendations:** After running Attack Path Analysis, you'll see a list of identified attack paths. Each path details the resources involved and potential weaknesses. Defender for Cloud also provides recommendations to mitigate these risks. These might include:
    - Patching vulnerable resources
    - Hardening security configurations (e.g., restricting access controls)
    - Implementing additional security measures (e.g., firewalls)

How to Use This in AZ-700:

- You'll need to understand how to interpret the attack paths displayed by Defender for Cloud.
- You should be able to identify network security weaknesses based on the information provided.
- The exam might ask you to evaluate specific recommendations and determine their effectiveness in mitigating network security risks.

Identifying Network Resources with Security Explorer

- **Microsoft Defender for Cloud Security Explorer:** This is another tool within Defender for Cloud that allows you to investigate security posture across your Azure resources. It provides a unified view of security alerts, recommendations, and resource configurations.
- **Identifying Network Resources:** Security Explorer allows you to filter and search for specific resources based on various criteria, including:
  - Resource type (e.g., virtual networks, load balancers)
  - Location
  - Security state (e.g., compliant, misconfigured)

How to Use This in AZ-700:

- You should be familiar with the different types of Azure network resources and their security implications.
- The exam might ask you to use Security Explorer to identify specific network resources with security vulnerabilities or misconfigurations.
- Understanding how to filter and search within Security Explorer is crucial for efficient network security monitoring.

Additional Tips:

- Make sure you understand the different types of network security threats (e.g., Denial-of-Service attacks, data breaches).
- Be familiar with the various security controls available in Azure to protect your network resources (e.g., Network Security Groups, Azure Firewall).
- Practice using Microsoft Defender for Cloud in a lab environment to get hands-on experience with Attack Path Analysis and Security Explorer.

Design, implement, and manage connectivity services (20–25%)

Design, implement, and manage a site-to-site VPN connection

Designing a Site-to-Site VPN Connection:

- **Connectivity Requirements:**
  - Identify traffic volume and bandwidth needs between your on-premises network and Azure.
  - Determine if you need high availability (HA) for the connection (explained later).
- **IP Addressing:**
  - Plan non-overlapping address spaces for your on-premises network and Azure VNet.
- **Security Considerations:**
  - Choose an appropriate encryption algorithm (e.g., AES-GCM) and key strength.
  - Define firewall rules to control traffic flow between the connected networks.

- **High Availability (HA):**
  - For redundancy, configure Active-Active mode with two virtual network gateways (VNet gateways) in your Azure VNet, each paired with your on-premises VPN device. Traffic is balanced across both tunnels.
  - Alternatively, use Active-Standby mode with a primary and secondary VNet gateway. The secondary takes over if the primary fails.

Selecting a VNet Gateway SKU:

VNet gateway SKUs offer different Throughput (Mbps) capacities. Choose the one that aligns with your expected traffic volume:

- **Basic:** Suitable for low-bandwidth scenarios (up to 100 Mbps).
- **Standard:** Good for typical workloads (up to 1 Gbps).
- **High Performance:** Ideal for high-bandwidth needs (up to 10 Gbps).
- **Ultra Performance:** Supports extremely high bandwidth requirements (up to 80 Gbps).

Implementing a Site-to-Site VPN Connection:

- **Create a Resource Group:** Organize your Azure resources in a logical group.
- **Configure a Virtual Network (VNet):** Define the address space for your Azure resources.
- **Create a GatewaySubnet:** A dedicated subnet within your VNet for deploying the VNet gateway.
- **Deploy a Virtual Network Gateway (VNet Gateway):** Choose the desired SKU based on your needs.
- **Configure a Local Network Gateway:** This represents your on-premises VPN device.
- **Create a VPN Connection:** Specify the VNet gateway, local network gateway, connection type (IPSec), and authentication method (e.g., pre-shared key).
- **Configure your On-Premises VPN Device:** Establish the VPN tunnel according to your device's specific instructions.
- **Verify the Connection:** Use Azure portal tools or network monitoring tools to confirm successful connectivity.

Additional Resources:

- Microsoft documentation on S2S VPNs: https://learn.microsoft.com/en-us/azure/vpn-gateway/
- Exam AZ-700: Design and implement core networking infrastructure: https://learn.microsoft.com/en-us/credentials/certifications/resources/study-guides/az-700

AZ-700: Site-to-Site VPN Connections

Policy-Based vs. Route-Based VPN Connections:

- **Policy-Based VPN (Default):** This is the most common type and is easier to configure. It uses pre-defined security policies to determine which traffic is allowed through the VPN tunnel. Policies define source and destination address prefixes, protocols, and ports.
- **Route-Based VPN:** Offers more granular control by relying on routing protocols (like BGP) to determine which traffic goes through the VPN. This is typically used for complex network configurations with specific routing needs.

Choosing the Right Option:

- Use a policy-based VPN for most scenarios, especially for simple connectivity between your on-premises network and Azure VNet.
- Consider a route-based VPN if you need fine-grained control over traffic routing or have complex network configurations with existing BGP routing.

Create and Configure a Local Network Gateway (LNG):

An LNG represents your on-premises network in Azure. It defines the address space of your on-premises network and the public IP address of your VPN device. Here's how to create and configure it:

1. Access Azure Portal and navigate to "Virtual Wans" or "Virtual Network Gateways" (depending on your configuration).
2. Click "Create" and choose "Local network gateway".
3. Provide a name, resource group, and location for the LNG.
4. Define the "Address space" that represents your on-premises network IP range.
5. Specify the "Public IP address" of your on-premises VPN device.
6. Review and create the LNG.

Create and Configure an IPSec/IKE Policy:

IPSec (Internet Protocol Security) and IKE (Internet Key Exchange) are protocols used to establish a secure VPN tunnel. The IPSec/IKE policy defines the encryption algorithms, key exchange methods, and other security parameters for the connection. Here's the configuration process:

1. Go to your Virtual Network Gateway in Azure Portal.
2. Click on "Connections" and then "Add".
3. Choose "Site-to-Site (IPSec)" as the connection type.
4. Select the previously created Local Network Gateway.
5. Define the "Shared key" (PSK) used for authentication between Azure and your on-premises VPN device (refer to your device's documentation for configuration).
6. Choose the appropriate "IKE version" (typically IKEv2).
7. Optionally, configure advanced settings like encryption algorithms, DH group, and lifetime cycles.

8. Review and create the VPN connection. Azure will provide a configuration script that you can download and import into your on-premises VPN device to complete the setup.

Additional Resources:

- Microsoft documentation on S2S VPNs: https://learn.microsoft.com/en-us/azure/vpn-gateway/
- Video tutorial on configuring an S2S VPN: https://www.youtube.com/playlist?list=PLOyJPig7sNjyyA_yU5nLFSncz_SQx5IJ4

Create and Configure a Virtual Network Gateway (VNet Gateway):

A VNet Gateway acts as a secure tunnel endpoint that connects your on-premises network to your Azure virtual network (VNet). Here's how to create and configure one:

- **Design:**
  - Choose a gateway type: VNet Gateway supports various options like VPN (IPSec), ExpressRoute, and internet-facing. Select "VPN" for site-to-site connectivity.
  - Decide on a SKU: Different SKUs offer varying throughput and pricing options. Consider your bandwidth needs.
  - Plan the gateway subnet: Create a dedicated subnet within your VNet to deploy the VNet Gateway. It should have enough IP addresses for future scaling.
- **Configuration:**
  - Use the Azure portal, Azure PowerShell, or Azure CLI to create the VNet Gateway resource.
  - Specify the resource group, location, gateway type (VPN), SKU, and the pre-created gateway subnet.
  - Configure public IP address allocation (static or dynamic) for the VNet Gateway.

Diagnose and Resolve Virtual Network Gateway Connectivity Issues:

Troubleshooting connectivity issues with your VNet Gateway involves identifying the root cause. Here are some steps:

- **Verification:**
  - Check the VNet Gateway health using the Azure portal. Look for errors or warnings.
  - Verify the VPN connection status. Ensure it's connected and operational.
- **Connectivity Tools:**
  - Use Azure Monitor logs to diagnose issues related to routing, IKE negotiation, or IPSec tunnels.
  - Leverage tools like Azure Network Watcher to perform connectivity tests and identify bottlenecks.
- **Common Issues:**

- Incorrect configuration: Double-check settings like subnet assignments, public IP addresses, and firewall rules.
- Routing problems: Ensure proper routing is configured on both the on-premises network and the Azure VNet.
- Security group restrictions: Verify that security groups don't block necessary ports (IKE and IPSec)

Azure Extended Network:

Azure Extended Network is a deprecated service (retired in June 2023). It aimed to connect on-premises networks to Azure through private peering over the Microsoft backbone network. However, Microsoft recommends alternative solutions:

- **Site-to-Site VPN:** This is the preferred method for secure private connectivity between on-premises and Azure environments.
- **ExpressRoute:** Offers a dedicated private connection to Azure bypassing the public internet.

Design, implement, and manage a point-to-site VPN connection

Selecting a Virtual Network Gateway SKU for Point-to-Site VPN:

P2S VPN connections rely on virtual network gateways (VNet gateways) in Azure. When choosing a VNet gateway SKU, consider these factors:

- **Number of concurrent connections:** Select a SKU that supports the expected number of users simultaneously connecting via P2S VPN. Basic SKU is suitable for low connection counts, while Standard SKU caters to more users.
- **Throughput needs:** Standard SKUs offer higher bandwidth compared to Basic SKUs for data transfer over the VPN tunnel.
- **Scalability requirements:** If you anticipate growth in P2S users, choose a Standard SKU for easier scaling.

Tunnel Type Selection and Configuration:

There are two main tunnel types for P2S VPN:

- **Policy-based Routing (IKEv2):** This is the most common and recommended option. It offers dynamic routing capabilities and negotiation between the VPN client and Azure VNet gateway.
- **Route-based VPN (OpenVPN):** This offers more granular control over routing but requires manual configuration on both the client and Azure side.

Authentication Method Selection:

P2S VPN supports various authentication methods to secure connections:

- **Pre-shared Key (PSK):** A simpler method using a shared secret for authentication. However, it's less secure for large deployments.

- **RADIUS Authentication:** Provides centralized user authentication using a Remote Authentication Dial-In User Service (RADIUS) server. More secure for managing multiple users.
- **Azure Active Directory (Azure AD) Authentication with Microsoft Entra:** Integrates with Azure AD for user authentication using existing identities. Offers strong security and centralized management.

RADIUS Authentication Configuration:

To configure RADIUS authentication:

- **Deploy a RADIUS server:** This can be an on-premises server or a cloud-based solution like Azure AD RADIUS.
- **Configure the RADIUS server:** Set up user accounts, authentication protocols (e.g., PAP, MS-CHAP v2), and shared secrets.
- **Configure the Azure VNet gateway:** In the Azure portal, specify the RADIUS server details like address, shared secret, and authentication methods.

Azure AD Authentication with Microsoft Entra Configuration:

Microsoft Entra (formerly Azure AD Multi-Factor Authentication) integrates with Azure AD for P2S VPN authentication. Here's how to configure it:

- **Enable Azure AD authentication on the VNet gateway:** In the Azure portal, choose this option during VNet gateway creation.
- **Configure user groups:** Assign specific Azure AD user groups access to the P2S VPN connection.
- **Configure Microsoft Entra for multi-factor authentication (MFA):** Set up MFA policies for additional security on top of Azure AD credentials.

Additional Resources:

- Microsoft documentation on P2S VPN: https://learn.microsoft.com/en-us/azure/vpn-gateway/point-to-site-about
- Selecting a VNet gateway SKU: https://learn.microsoft.com/en-us/azure/vpn-gateway/

Implement a VPN Client Configuration File

Azure Point-to-Site (P2S) VPN connections allow individual devices to securely connect to your Azure Virtual Network (VNet). The client configuration file holds the information needed for a device to establish the VPN tunnel. You cannot directly edit this file, but Azure generates it based on your P2S configuration.

Here's how to obtain the VPN client configuration file:

- **Azure Portal:**
    1. Go to your virtual network gateway resource.
    2. Select "Point-to-Site configuration."

3. Click "Download VPN client." This generates a zip file containing the configuration files.

- **PowerShell:** Use the New-AzVpnClientConfiguration cmdlet to generate the configuration for a specific resource group and gateway name.

The configuration file typically includes:

- **Connection details:** Server address, VPN type (IKEv2 or OpenVPN).
- **Authentication settings:** Based on your chosen method (certificate, Azure AD, RADIUS).
- **Encryption settings:** Cipher suites and algorithms used for secure communication.

Diagnose and Resolve Client-Side and Authentication Issues

Troubleshooting P2S VPN connectivity issues involves checking both client-side configuration and authentication methods. Here are some common issues and solutions:

- **Client configuration errors:** Ensure the downloaded configuration file is imported correctly on the client device. Verify settings like connection name, server address, and authentication certificates (if used).
- **Connectivity issues:** Check firewall rules on the client device that might block VPN traffic. Test internet connectivity and ensure the client device can reach the Azure VPN Gateway endpoint.
- **Authentication failures:** For certificate-based authentication, verify the client certificate is valid, not expired, and trusted by the Azure VPN Gateway. For Azure AD or RADIUS authentication, check if the user has the necessary permissions and multi-factor authentication (MFA) is configured correctly (if used).

Azure Requirements for Always On VPN

Always On VPN is a Windows 10 feature that allows a device to automatically establish and maintain a VPN connection. While Azure supports P2S VPN with Always On VPN, there are specific requirements:

- **Supported Operating Systems:** Windows 10 or later versions with the Always On VPN feature enabled.
- **Configuration method:** Pushing the VPN profile using Group Policy or Mobile Device Management (MDM) is recommended for centralized control.
- **Authentication method:** Certificate authentication is the most secure option for Always On VPN with Azure P2S.

Azure Requirements for Azure Network Adapter

Azure Network Adapter is a feature that allows a Windows or Linux VM to directly connect to an Azure VNet without needing a separate VPN connection. Here are the key points for Azure Network Adapter:

- **Supported Platforms:** Windows Server 2016 or later, most Linux distributions.

- **Deployment Model:** Only available for VMs deployed in the Resource Manager deployment model.
- **Connectivity:** Provides direct, private connectivity between the VM and the VNet resources.
- **Security:** Requires proper network security group (NSG) configuration to control inbound and outbound traffic.

Additional Resources:

- Microsoft documentation on Point-to-Site VPN: https://learn.microsoft.com/en-us/azure/vpn-gateway/point-to-site-about
- Download Azure VPN Client: https://www.microsoft.com/p/azure-vpn-client/9np355qt2sqb
- Always On VPN documentation: https://learn.microsoft.com/en-us/windows-server/remote/remote-access/tutorial-aovpn-deploy-setup
- Azure Network Adapter documentation: https://learn.microsoft.com/en-us/windows-server/manage/windows-admin-center/azure/use-azure-network-adapter

Design, implement, and manage Azure ExpressRoute

Selecting an ExpressRoute Connectivity Model

ExpressRoute offers two connectivity models:

1. **Private Peering:** This establishes a private connection between your on-premises network and Microsoft Azure through a connectivity provider. Traffic stays off the public internet, improving security and performance.
2. **Microsoft Peering:** This model connects to Microsoft public cloud services (like Office 365 and Dynamics 365) through the Microsoft global network.

Choosing the Model:

- Select **Private Peering** for secure, private connectivity to Azure resources and Microsoft services within the same region.
- Choose **Microsoft Peering** if you need to connect to Microsoft public cloud services in addition to Azure resources.

Selecting an Appropriate ExpressRoute SKU and Tier

ExpressRoute comes in different SKUs (Standard and Basic) and tiers (Local, Metro, Global Reach). Here's a breakdown:

- **SKUs:**
  - **Standard:** Offers higher bandwidth options (up to 100 Gbps) and features like Global Reach (explained later).
  - **Basic:** Lower bandwidth options (up to 2 Gbps) suitable for smaller deployments.
- **Tiers:**

  - **Local:** Connects your on-premises network to Azure within the same metro area.
  - **Metro:** Connects your network to Azure across nearby metro regions.
  - **Global Reach:** Extends private connectivity across geographically distant Azure regions. (Available only with Standard SKU)

Choosing the SKU and Tier:

- Consider your bandwidth needs. Standard SKU offers higher options for demanding workloads.
- Evaluate geographic reach requirements. Local is for same-metro connections, Metro for nearby regions, and Global Reach for geographically dispersed regions (with Standard SKU).

Designing ExpressRoute for Specific Requirements

Now, let's explore how to design ExpressRoute for key functionalities:

- **Cross-Region Connectivity:** Utilize ExpressRoute Global Reach (Standard SKU only) to establish private connections between Azure resources in different regions.
- **Redundancy:** Configure redundant ExpressRoute circuits from different connectivity providers to ensure uptime if one circuit fails.
- **Disaster Recovery:** Designate a secondary Azure region for disaster recovery and establish ExpressRoute connectivity to both primary and secondary regions for seamless failover.

Additional Resources:

- Microsoft offers a comprehensive learning module on Design and implement Azure ExpressRoute: https://learn.microsoft.com/en-us/training/modules/design-implement-azure-expressroute/
- This module covers configuration exercises using Azure portal and Azure PowerShell.

ExpressRoute Options

Azure ExpressRoute offers several options to tailor your private connection to Microsoft Azure, each catering to specific use cases:

- **Global Reach:** Connects your on-premises network to Azure across different geographical regions. This is ideal for organizations with geographically dispersed resources or users who need low latency access to Azure services in multiple regions.
  - Implementation: You provision a single ExpressRoute circuit and configure Global Reach within it. This establishes private connectivity between your on-premises network and multiple Azure regions.
- **FastPath:** Optimizes the data path for specific Microsoft cloud services like Azure Storage, Azure Databricks, and Azure SQL Database. It bypasses the standard internet routing for these services, potentially reducing latency and improving performance.
  - Implementation: You enable FastPath on your ExpressRoute circuit during provisioning or later through Azure PowerShell or the Azure CLI. This activates the optimized data path for the designated services.

- **ExpressRoute Direct:** Provides a dedicated private connection to Microsoft's global network, bypassing internet routing entirely. It offers the highest level of performance and isolation but requires a physical connection at a Microsoft datacenter location.
    - Implementation: This option isn't readily available through self-service provisioning. You'll need to contact Microsoft sales for ExpressRoute Direct connectivity.

Choosing Peering Configuration

ExpressRoute supports two primary peering configurations:

- **Azure private peering:** Enables private connectivity between your on-premises network and all Azure services (including virtual networks, PaaS services, and IaaS resources). This is the most common choice for private, secure access to all Azure offerings.
- **Microsoft peering:** Allows private communication with Microsoft cloud services (like Office 365, Dynamics 365, and Azure Active Directory) that are not natively part of the Azure public cloud. This peering is helpful for hybrid deployments that integrate Azure with other Microsoft services.

You can also configure **both Azure private peering and Microsoft peering** on the same ExpressRoute circuit for scenarios where you need access to both types of resources.

Configuring Azure Private Peering

Here's a breakdown of configuring Azure private peering for your ExpressRoute circuit:

1. **Provision an ExpressRoute circuit:** Use the Azure portal, Azure PowerShell, or the Azure CLI to create a new circuit. Specify the connectivity provider (e.g., Equinix, Verizon) and location.
2. **Enable private peering:** During circuit creation or later, select "Azure private peering" in the peering configuration options.
3. **Configure BGP:** Establish Border Gateway Protocol (BGP) routing between your on-premises network and the Microsoft cloud. This involves exchanging routing information to ensure proper traffic flow. BGP configuration details will vary depending on your specific network equipment.
4. **Connect your virtual networks:** Once the ExpressRoute circuit and peering are set up, link your Azure virtual networks to the private peering connection. This enables private communication between your on-premises resources and your virtual networks in Azure.

Additional Considerations

- **High Availability:** Design your ExpressRoute circuit with redundancy to avoid a single point of failure. Consider provisioning circuits with different connectivity providers or locations.
- **Cost Optimization:** Evaluate your traffic patterns and requirements to determine the most cost-effective ExpressRoute option. Global Reach and FastPath may incur additional charges.

Configure Microsoft Peering:

Microsoft peering allows direct private connectivity between your on-premises network and various Microsoft Cloud services like Azure, Office 365, and Dynamics 365. Here's how to configure it:

- **Provision an ExpressRoute circuit:** Work with your connectivity provider to establish a circuit.
- **Enable Microsoft peering in the circuit:** Within the Azure portal or Azure PowerShell, navigate to your ExpressRoute circuit and enable Microsoft peering.
- **Configure routing:** Advertise your on-premises network prefixes to Azure via Border Gateway Protocol (BGP). This informs Azure about the traffic routes for your network.

Create and Configure an ExpressRoute Gateway:

An ExpressRoute gateway manages the connection between your virtual network and the ExpressRoute circuit. Here's the process:

- **Create an ExpressRoute gateway:** In the Azure portal or PowerShell, define a new ExpressRoute gateway resource within your desired virtual network resource group.
- **Link the ExpressRoute circuit:** Associate the previously created ExpressRoute circuit with the ExpressRoute gateway.
- **Configure routing:** Similar to Microsoft peering, advertise your virtual network prefixes to the ExpressRoute circuit using BGP.

Connect a Virtual Network to an ExpressRoute Circuit:

Connecting a virtual network to the ExpressRoute circuit establishes the private communication path. Here's how to achieve it:

- **Locate the virtual network:** Within the Azure portal or PowerShell, identify the virtual network you want to connect.
- **Subnet association:** Choose the specific subnet(s) within the virtual network that require ExpressRoute connectivity.
- **Link the ExpressRoute gateway:** Associate the ExpressRoute gateway you created earlier with the chosen subnet(s). This creates the connection.

Recommend a Route Advertisement Configuration:

BGP route advertisement dictates how traffic is routed between your on-premises network and Azure resources. Here's how to recommend a configuration:

- **Identify traffic types:** Analyze the traffic flow between your on-premises network and Azure services (Microsoft vs. internet-bound).
- **Specific vs. aggregate prefixes:** Decide whether to advertise specific prefixes for each subnet or use a more concise aggregate advertisement.
- **Redundancy:** Consider advertising prefixes to both primary and secondary connections within the ExpressRoute circuit for high availability.

Additional Resources:

- Microsoft official documentation offers in-depth explanations and step-by-step guides for each of these functionalities: https://learn.microsoft.com/en-us/training/modules/design-implement-azure-expressroute/
- You can find practice labs for these configurations to gain hands-on experience: https://github.com/microsoft/Deploy-and-Optimize-Azure-ExpressRoute-Private-Peering (search for "AZ-700 Labs")

Configuring Encryption over ExpressRoute

While ExpressRoute itself establishes a private connection, you can add an extra layer of security by encrypting the data traversing the connection. Here are two main options:

- **VPN within ExpressRoute:** You can configure a Point-to-Point (P2P) VPN tunnel within the ExpressRoute circuit. This encrypts data traffic between your on-premises network and Azure using protocols like IPSec.
- **Azure Virtual WAN with ExpressRoute:** Azure Virtual WAN allows centralizing and managing network connectivity across various branches and cloud environments. You can integrate ExpressRoute circuits with Virtual WAN and leverage its built-in encryption capabilities.

Implementing Bidirectional Forwarding Detection (BFD)

BFD is a fast convergence protocol that helps detect and recover from path failures within the ExpressRoute circuit. It continuously sends probes between your network and Microsoft's edge and quickly identifies any outages. By implementing BFD:

- You achieve faster failover times to secondary connections in case of path disruptions.
- BFD helps maintain high availability and resiliency for your Azure deployments.

Diagnosing and Resolving ExpressRoute Connection Issues

Troubleshooting ExpressRoute connectivity issues requires a systematic approach. Here's a breakdown:

- **Verification:** First, verify basic configurations like circuit provisioning, peering setup, and routing policies. Tools like Azure portal or Azure Resource Manager can help with this.
- **Connectivity Checks:** Use tools like ping and traceroute to test connectivity between your on-premises network and Azure resources. Identify any bottlenecks or connectivity drops.
- **Azure ExpressRoute Diagnostics:** Leverage built-in diagnostics tools within Azure to analyze connectivity issues. These tools provide insights into BGP route propagation, peering status, and potential configuration problems.
- **Service Provider Involvement:** If the issue persists, it might be related to the connectivity provided by your service provider. Collaborate with your provider to isolate and address the problem within their network.

Additional Resources:

- Microsoft offers a comprehensive learning module on designing, implementing, and managing Azure ExpressRoute: https://learn.microsoft.com/en-us/training/modules/design-implement-azure-expressroute/

- This module covers additional functionalities like ExpressRoute global reach and FastPath.

Design and implement an Azure Virtual WAN architecture

Selecting a Virtual WAN SKU

Azure Virtual WAN offers two SKUs:

- **Standard:** Provides full mesh connectivity between virtual WAN hubs within a region. Ideal for scenarios with frequent communication between geographically distributed locations.
- **Basic:** Offers limited connectivity options. Best suited for simple, hub-and-spoke architectures with no need for inter-hub communication.

Choosing the right SKU depends on your needs:

- **Number of connected locations:** Standard offers full mesh for better scalability across regions.
- **Inter-hub communication:** Standard allows communication between hubs, while Basic doesn't.
- **Cost:** Standard has a higher cost than Basic.

Designing a Virtual WAN Architecture

Once you've chosen the SKU, define your architecture by considering these elements:

- **Connectivity types:**
  - **Branch connectivity:** Connect branch offices using Site-to-Site VPN (S2S VPN) or ExpressRoute for private connectivity.
  - **Remote user access:** Allow remote users to connect securely with Point-to-Site VPN (P2S VPN).
  - **Cloud connectivity:** Integrate Azure Virtual Networks (VNets) using VNet connections.
- **Security services:** Leverage Azure Firewall within the Virtual WAN hub for centralized network security policies.
- **Routing:** Define routing policies to control traffic flow within the Virtual WAN.

Creating a Hub in Virtual WAN

Here's how to create a virtual WAN hub:

1. Log in to the Azure portal and navigate to the Virtual WAN service.
2. Click "Create" and choose a name and resource group for your Virtual WAN.
3. Select the desired location for the hub (consider factors like latency and compliance).
4. Choose the SKU (Standard or Basic) based on your design requirements.
5. Review and create the virtual WAN hub.

Additional Resources:

- Microsoft documentation provides in-depth information on Virtual WAN architecture design and implementation: https://learn.microsoft.com/en-us/azure/virtual-wan/virtual-wan-about
- You can find a video tutorial demonstrating Virtual WAN hub creation: https://www.youtube.com/watch?v=Ef4qeKrQ4r4

Choosing Appropriate Scale Units for Gateways:

- **Site-to-Site VPN Gateway:**
  - **Basic:** Suitable for low-bandwidth scenarios (up to 1 Gbps) with limited connections.
  - **Standard:** Ideal for most deployments, offering up to 10 Gbps throughput and supporting more connections.
  - **High Performance:** Best for high-bandwidth requirements (up to 20 Gbps) and critical connections.
- **ExpressRoute Gateway:**
  - **Standard:** Supports up to 2 Gbps throughput for private connectivity to Azure.
  - **ExpressRoute Global Reach:** Extends connectivity across different Azure regions for geographically dispersed resources.

Deploying a Gateway into a Virtual WAN Hub:

1. Access the Azure portal and navigate to **Virtual WANs**.
2. Select your Virtual WAN and then go to **Gateways**.
3. Choose the desired gateway type (Site-to-Site VPN or ExpressRoute) and size (scale unit).
4. Configure the gateway settings based on your specific needs (e.g., VPN device IP address for Site-to-Site).
5. Click **Create** to deploy the gateway into your Virtual WAN hub.

Configuring Virtual Hub Routing:

Azure Virtual WAN offers two routing methods:

- **Basic:** Automatic routing for basic scenarios with limited spokes.
- **Custom:** Manual configuration for more granular control over traffic flow.

Custom Routing with User-Defined Routes (UDRs):

1. Navigate to your Virtual WAN hub in the Azure portal.
2. Go to **Routing** and select **User-defined routes**.
3. Click **Add** to create a new route.
4. Specify the destination prefix (traffic target), next hop type (internet, VNet, etc.), and next hop address (where to send traffic).

5. Define the priority for this route (higher values take precedence).
6. Click **Save** to create the UDR.

Integrating a Third-Party NVA for Cloud Connectivity:

Azure Virtual WAN allows integrating a third-party Network Virtual Appliance (NVA) for advanced security or functionality within the Virtual WAN hub. Here's a general process:

1. Deploy a VNet in your Virtual WAN hub where you want the NVA.
2. Deploy your desired third-party NVA solution in the VNet.
3. Configure peering between the VNet with the NVA and the VNets requiring NVA services.
4. Set up routing to direct traffic through the NVA for specific security policies or functionalities.

Additional Resources:

- Microsoft Documentation on Azure Virtual WAN: https://learn.microsoft.com/en-us/training/modules/introduction-azure-virtual-wan/
- Hub-Spoke Network Topology with Azure Virtual WAN: https://learn.microsoft.com/en-us/azure/architecture/networking/architecture/hub-spoke-vwan-architecture
- SD-WAN connectivity architecture with Azure Virtual WAN: https://learn.microsoft.com/en-us/azure/virtual-wan/sd-wan-connectivity-architecture

Design and implement application delivery services (15–20%)

Design and implement Azure Load Balancer and Azure Traffic Manager

Mapping Requirements to Azure Load Balancer Features:

- **High Availability:** Load Balancer distributes traffic across healthy VMs in a pool, ensuring service remains available even if one VM fails.
- **Scalability:** It automatically scales resources up or down based on traffic volume.
- **Health Monitoring:** Continuously monitors VM health and removes unhealthy ones from the pool.
- **Load Distribution:** Balances traffic across VMs using various methods (round robin, weighted, etc.).
- **Security:** Supports inbound and outbound traffic rules for granular control.

Identifying Use Cases for Azure Load Balancer:

- **Web applications:** Distributes traffic across web servers for high availability and performance.
- **Multi-tier applications:** Balances traffic between application tiers (web, database) for scalability.
- **Stateful applications:** Manages sessions for applications requiring session persistence.

- **Backend workloads:** Balances traffic to services like VMs, container instances, or Azure App Service.

Choosing an Azure Load Balancer SKU and Tier:

- **Basic SKU:** Cost-effective option for simple scenarios with limited throughput needs.
- **Standard SKU:** Offers higher throughput, availability zones for redundancy, and outbound rules.
- **Global Standard SKU (Public Preview):** Extends Standard SKU's capabilities across global regions for geographically distributed applications.

Tier Selection:

- **Basic Tier:** Ideal for development and testing environments.
- **General Purpose Tier:** Suitable for most production deployments.

Public vs. Internal Load Balancers:

- **Public Load Balancer:** Exposes your application to the public internet via a public IP address.
- **Internal Load Balancer:** Only accessible within the Azure virtual network, not exposed publicly.

Choose a public load balancer for applications users access from the internet. Use internal load balancers for applications within your virtual network that don't require external access.

Regional vs. Global Load Balancers:

- **Regional Load Balancer:** Distributes traffic within a single Azure region.
- **Global Load Balancer (Public Preview):** Distributes traffic across multiple Azure regions based on routing methods (geographic, performance).

Use a regional load balancer for applications primarily used within a specific region. Consider a global load balancer for geographically distributed deployments or to improve user experience by routing requests to the closest healthy endpoint.

Additional Resources:

- Microsoft Azure documentation: https://learn.microsoft.com/en-us/azure/load-balancer/
- Microsoft Azure Traffic Manager: https://learn.microsoft.com/en-us/azure/traffic-manager/traffic-manager-overview
- Load-balancing options in Azure: https://learn.microsoft.com/en-us/azure/architecture/guide/technology-choices/load-balancing-overview

Azure Load Balancer

- **What it is:** A layer 4 load balancer that distributes traffic across healthy backend servers (VMs, App Services, etc.) based on pre-defined rules.
- **Benefits:**
  - High availability: Ensures application uptime even if a server fails.

  - Scalability: Distributes traffic for increased capacity.
  - Performance: Optimizes traffic flow for faster response times.
- **Configuration:**
  - Define a load balancing pool containing backend servers.
  - Choose a health probe to monitor server health (e.g., ping, HTTP).
  - Select a load distribution method (e.g., round robin, least connections).
  - Create inbound NAT rules to map public ports to backend ports.

Azure Traffic Manager (ATM)

- **What it is:** A DNS-based traffic routing service that directs users to the most optimal backend service based on your chosen policy.
- **Benefits:**
  - Global load balancing: Routes traffic to geographically closest endpoints.
  - High availability: Fails over to healthy endpoints if a region becomes unavailable.
  - Performance optimization: Improves user experience by minimizing latency.
- **Routing methods:**
  - Priority: Directs traffic to the highest priority endpoint.
  - Weighted round robin: Distributes traffic with weight assigned to each endpoint.
  - Geographic: Routes users to the closest endpoint based on their location.
  - Performance: Routes users to the endpoint with the fastest response times.
  - Multi-value: Returns multiple healthy endpoints for further client-side selection.

Gateway Load Balancer (Azure Front Door)

While not explicitly mentioned in the AZ-700 objectives, it's important to understand the difference between Application Gateway and Azure Front Door (AFD):

- **Application Gateway:** A layer 7 load balancer that operates at the application layer (HTTP/HTTPS). Offers advanced features like content routing, SSL offloading, and web application firewall (WAF) capabilities.
- **Azure Front Door (AFD):** A global content delivery network (CDN) that provides layer 7 load balancing, web application acceleration, and security features. Ideal for highly scalable and geographically distributed web applications.

Azure Load Balancer

- **Layer 4 Load Balancing:** Operates at the transport layer, distributing traffic based on IP addresses and ports.
- **Implementation:**

  - **Load Balancing Rule:** Defines how traffic is distributed to backend resources (VMs, App Services). You can choose distribution methods like round robin or based on health probes.
  - **Inbound NAT Rules:** Allow external traffic to reach backend resources with private IP addresses using public ports. You map a public port on the load balancer to a specific port on the backend VM.
  - **Outbound NAT Rules:** Configure outbound connections initiated from backend resources. You can use:
    - **Dynamic Outbound NAT (DONAT):** Assigns a public IP address from a pool to backend VMs for outbound connections.
    - **Static Outbound NAT (SNAT):** Maps a public IP address to a specific backend VM for consistent outbound communication.

Azure Traffic Manager

- **DNS-based Load Balancing:** Operates at the DNS layer, directing traffic to geographically distributed endpoints based on user location or other routing methods.
- **Implementation:**
  - Create a Traffic Manager profile and specify the endpoints (web apps, VMs, etc.) you want to route traffic to.
  - Choose a traffic routing method:
    - **Priority:** Route traffic to the highest priority endpoint.
    - **Performance:** Route traffic to the endpoint with the fastest response time.
    - **Geographic:** Route traffic to the endpoint closest to the user's location.
    - **Weighted Round Robin:** Distribute traffic across endpoints with weight assignments.
    - **Subnet:** Route traffic based on the user's IP subnet.
    - **Multi-value:** Return multiple healthy endpoints for applications with high availability needs.

Resources for further learning:

- Microsoft Documentation:
  - Azure Load Balancer: https://learn.microsoft.com/en-us/azure/load-balancer/
  - Azure Traffic Manager: https://learn.microsoft.com/en-us/azure/traffic-manager/

Design and implement Azure Application Gateway

Mapping Requirements to Features

Azure Application Gateway (AG) is a Layer 7 load balancer that manages traffic for your web applications in Azure. Here's how to map your requirements to its features:

- **High Availability & Scalability:** AG distributes traffic across multiple backend servers (pools) for redundancy and handles high traffic volumes.
- **Application Routing:** Route traffic based on URL paths, host headers, or other HTTP(S) request attributes for intelligent traffic management.
- **Web Application Firewall (WAF):** (Available in specific SKUs) Protects your applications from common web attacks.
- **SSL/TLS Termination:** Offloads SSL processing from backend servers, improving performance and security.
- **Performance Optimization:** Features like caching and compression can improve application responsiveness.

Identify your needs:

- Do you need Layer 7 routing for your web application?
- Do you require high availability and scalability?
- Is web application security a concern?
- Would SSL termination benefit your application architecture?

Match needs with features: If your requirements align with these features, AG is a good choice.

Use Cases for Azure Application Gateway

Here are some common use cases for AG:

- **Load Balancing:** Distributing traffic across multiple web servers for high availability and scalability.
- **Web Application Traffic Management:** Routing traffic based on complex rules for different application versions or functionalities.
- **Web Application Firewall (WAF):** Protecting your web applications from common attacks (with specific SKUs).
- **SSL Offloading:** Improving performance and security by terminating SSL/TLS connections at the gateway.
- **Microservices Architecture:** Managing traffic flow between microservices based on specific routing rules.

Consider AG if you need any of these functionalities in your Azure environment.

Manual vs. Autoscale

AG offers two backend pool capacity options:

- **Manual Scaling:** You define the number of backend servers in the pool and manage scaling manually.

- **Autoscaling:** AG automatically scales the backend pool based on predefined metrics like CPU or memory usage.

Choose manual scaling for predictable traffic patterns.

Use autoscaling for dynamic workloads with fluctuating traffic to optimize resource utilization and cost.

Creating a Backend Pool

A backend pool is a group of backend servers (VMs, App Services, etc.) that AG distributes traffic across. Here's how to create one:

1. **Navigate to your Application Gateway resource in the Azure portal.**
2. Go to **Backend pools** section.
3. Click **Add**.
4. Provide a name for the pool.
5. Choose the **Add a target** option and select the type of backend resource (VM, App Service, etc.).
6. Specify the target resources (IP addresses or URIs) you want to add to the pool.
7. Click **OK** to create the pool.

You can add/remove backend servers from the pool as needed.

Configuring Health Probes

Health probes monitor the health of backend servers in a pool. Unhealthy servers are removed from traffic distribution. Here's how to configure probes:

1. **Navigate to your Application Gateway in the Azure portal.**
2. Go to **Health probes** section.
3. Click **Add**.
4. Choose the probe type (e.g., HTTP, TCP).
5. Define the probe target (path, port) and interval.
6. Set the unhealthy threshold (consecutive failures before marking unhealthy).
7. Click **OK** to create the probe.

Listeners:

- Listeners are virtual entities that accept incoming traffic on specific ports and protocols (HTTP/HTTPS).
- You define a listener with a name, frontend IP configuration (public/private), port, and protocol.
- An Application Gateway can have multiple listeners to handle different types of traffic.

Routing Rules:

- Routing rules determine how incoming requests are forwarded to backend resources based on conditions.
- You define conditions using a combination of factors like:
    - Path (URL): Route traffic based on specific paths within the URL.
    - Host Header: Route traffic based on the hostname in the request header.
    - Headers: Route based on custom headers present in the request.
- Routing rules point to a specific backend pool containing the target resources.

HTTP Settings:

- HTTP settings define how Application Gateway handles the HTTP aspects of communication.
- You can configure:
    - **Pick-for-sticky sessions:** Maintain user sessions on a specific backend server for a consistent experience.
    - **Caching:** Enable caching of static content to improve performance.
    - **URL rewrite:** Modify request or response URLs on the fly (more on rewrite sets later).
    - **Timeouts:** Set timeouts for backend server responses to prevent hanging requests.

Transport Layer Security (TLS):

- TLS encrypts communication between the internet and your Application Gateway for secure access.
- You can configure:
    - **TLS/SSL certificates:** Upload your public SSL certificate to terminate TLS at the Gateway.
    - **Client certificates:** Require client certificates for mutual authentication (optional).
    - **Cipher suites:** Choose the supported TLS cipher suites for secure connections.

Rewrite Sets:

- Rewrite sets allow you to modify request or response URLs based on predefined rules.
- You can define rewrite rules with:
    - Search string: The part of the URL you want to modify.
    - Replace string: The value to replace the search string with.
    - Regular expressions: Use advanced patterns for complex URL manipulations.
- Rewrite sets are referenced in HTTP settings to achieve URL modifications.

Resources for further learning:

- Microsoft Documentation: https://learn.microsoft.com/en-us/training/modules/configure-azure-application-gateway/
- Tutorial: Create application gateway with multiple websites: https://learn.microsoft.com/en-us/azure/application-gateway/create-multiple-sites-portal

Design and implement Azure Front Door

Mapping Requirements to Features:

- **Performance:** AFD utilizes Microsoft's global edge network, bringing content closer to users for faster load times. Features like:
  - **Anycast Routing:** Directs traffic to the nearest edge location.
  - **Health Probing:** Monitors backend health and automatically fails over to healthy origins.
- **Security:** AFD bolsters your application's security with:
  - **Web Application Firewall (WAF):** Filters out malicious traffic.
  - **HTTPS Offloading:** Improves performance by handling SSL encryption/decryption at the edge.
- **Scalability:** AFD seamlessly scales to handle traffic spikes.
- **Global Reach:** Delivers content globally with numerous Points of Presence (PoPs).

Identifying Use Cases:

AFD shines in scenarios like:

- **Globally distributed web applications:** Ensures optimal user experience across regions.
- **Static content delivery:** Delivers static content like images and scripts efficiently.
- **API Gateway:** Provides a single entry point for APIs and enforces access control.

Choosing an Appropriate Tier:

AFD offers two tiers:

- **Standard:** Cost-effective option for moderate traffic and basic features.
- **Premium:** Ideal for high-traffic scenarios and advanced features like custom routing rules.

Configuring Azure Front Door:

Here's a breakdown of the configuration process:

- **Front Door Profile:** Create a profile specifying the resource group and location.
- **Frontend Hosts:** Define custom domain names for users to access your application.
- **Routing Rules:** Specify how AFD routes traffic to backend origins. Options include:
  - **Weighted Round Robin:** Distributes traffic evenly across healthy origins.

  - **Priority Routing:** Directs traffic to higher-priority origins first.
  - **Path-Based Routing:** Routes traffic based on URL paths.
- **Origins:** Add your backend servers (App Services, Azure Functions, etc.) as origins.
- **Health Probes:** Configure probes to monitor origin health and trigger failover.

Configuring SSL Termination and Encryption:

AFD can handle SSL termination, decrypting HTTPS traffic at the edge and improving performance. You can also configure end-to-end encryption by installing certificates on your backend servers and AFD.

Additional Resources:

- Microsoft Documentation: https://learn.microsoft.com/en-us/azure/frontdoor/
- Quickstart: Create an Azure Front Door profile: https://learn.microsoft.com/en-us/azure/frontdoor/create-front-door-portal

Azure Front Door

Azure Front Door (AFD) is a cloud-based content delivery network (CDN) and global load balancer service that accelerates content delivery and distribution for your web applications and APIs. It improves performance, scalability, and security by strategically caching content at geographically distributed edge locations (Points of Presence or PoPs) closer to your users.

Configuring Caching

AFD offers various caching options to optimize content delivery:

- **Caching Rules:** Define which content types (e.g., HTML, images, JavaScript) and URLs should be cached for a specific duration.
- **TTL (Time-to-Live):** Set the expiration time for cached content, controlling how often it's refreshed from the origin server.
- **Cache Behavior:** Specify cache behavior for specific scenarios (e.g., query strings, dynamic content).

Benefits of Caching:

- **Reduced Latency:** Users receive content from the nearest PoP, minimizing travel time.
- **Reduced Origin Load:** Origins are relieved of serving frequently accessed content.
- **Improved Scalability:** AFD handles increased traffic efficiently.

Configuring Traffic Acceleration

AFD employs several techniques to accelerate traffic:

- **HTTP/2:** Uses this faster protocol compared to HTTP/1.1 for efficient communication.
- **TCP Offloading:** Handles TCP connections on the edge servers, freeing up origin resources.

- **TLS Offloading:** Performs encryption/decryption at the edge, further optimizing origin performance.
- **Origin Connection Multiplexing:** Establishes multiple connections to origins for faster data transfer.

Benefits of Traffic Acceleration:

- **Faster Page Load Times:** Users experience quicker content delivery.
- **Improved User Experience:** Enhances website responsiveness and engagement.

Implementing Rules, URL Rewrite, and URL Redirect

AFD's routing engine allows you to define rules for managing traffic flow and content delivery:

- **Routing Rules:** Based on factors like HTTP method, URL path, or hostname, determine which backend to route a request to.
- **URL Rewrite:** Modify the request URL path before forwarding it to the origin server (e.g., SEO-friendly URLs).
- **URL Redirect:** Permanently (301) or temporarily (302) redirect users to a different URL based on conditions.

Benefits of Rules, URL Rewrite, and URL Redirect:

- **Traffic Management:** Granular control over how traffic is directed to backends.
- **SEO Optimization:** Enhance search engine ranking with URL rewrites.
- **User Experience:** Provide seamless redirects for content updates or maintenance.

Securing an Origin with Azure Private Link

Azure Private Link enables private, secure connections between your web app or service in a virtual network (vNet) and AFD without traversing the public internet. This isolation protects against unauthorized access.

Benefits of Azure Private Link:

- **Enhanced Security:** Traffic flows over a secure, dedicated Azure network connection.
- **Improved Control:** Limits exposure of application endpoints to only authorized resources.
- **Compliance:** Adheres to stricter data security regulations.

Additional Considerations

- **Health Probes:** AFD monitors backend health using health probes (e.g., HTTP/TCP) to ensure traffic is directed to healthy origins.
- **WAF Integration:** Integrate Azure Web Application Firewall (WAF) with AFD for additional protection against malware and common attacks.
- **Monitoring and Logging:** Monitor AFD performance metrics and logs for troubleshooting and optimization.

Design and implement private access to Azure services (10–15%)

Design and implement Azure Private Link service and Azure private endpoints

Planning Private Endpoints

Planning is vital for effective private endpoint deployment. Here's what to consider:

- **Services:** Identify Azure PaaS services (Storage, SQL Database) or Private Link Services you want to connect to.
- **Virtual Networks (VNets):** Determine the VNets where you'll create private endpoints for secure access.
- **Connectivity:** Decide if connections will be within the same region, across peered VNets, or use ExpressRoute/VPN for on-premises access.
- **Security:** Define access controls using Resource Based Access Control (RBAC) to restrict who can access the private endpoint.

Creating Private Endpoints

Once you have a plan, here's how to create private endpoints:

1. **Access Azure portal or Azure CLI.**
2. **Navigate to the desired VNet.**
3. **Select "Private Endpoints" under the "Networking" section.**
4. **Click "Create" and choose the target service or Private Link service.**
5. **Specify a name, resource group, and the desired subnet within the VNet.**
   - Subnet selection influences outbound traffic routing.
6. **Configure private link service connection (if applicable).**
7. **Review and create the private endpoint.**

Note: You can create multiple private endpoints to access different subresources within an Azure service (e.g., separate endpoints for Azure Storage blobs and files).

Configuring Access to Private Endpoints

Here's how to configure access to private endpoints:

1. **For Azure PaaS services:** No additional configuration is needed for authorized services.
2. **For Private Link services:** The service owner needs to approve connection requests from your private endpoint in their Private Link service configuration.

RBAC plays a crucial role:

- Assign appropriate RBAC roles to users or service identities to grant access to the private endpoint within your VNet.

- By default, public internet access to the service remains possible unless explicitly restricted on the service side.

Creating a Private Link Service

A Private Link service acts as a wrapper for your service running behind a Standard Load Balancer. It allows authorized consumers to access your service privately using private endpoints within their VNets. Here's how to create one:

- **Deploy a Standard Load Balancer**: This balancer distributes traffic to your service instances.
- **Enable Private Link on the Load Balancer**: This exposes your service for private access through Private Link.
- **Configure Access (Optional)**: You can restrict access to specific VNets or subscriptions using Azure Role-Based Access Control (RBAC).

Integrating Private Link and Private Endpoints with DNS

Private endpoints use private IP addresses from your VNet. To simplify service discovery, you can integrate Private Link with DNS:

- **Create a Private DNS Zone**: This zone resides within your VNet and holds private DNS records.
- **Link the Private DNS Zone with the Private Link Service**: This creates a private DNS record mapping a custom domain name to the service's private endpoint.
- **Configure VM DNS Settings**: Point your VMs to use the private DNS server for resolving service names.

Integrating a Private Link Service with On-Premises Clients

Private Link inherently works within Azure. To connect on-premises clients to your Private Link service, you need to establish a private connection between your on-premises network and Azure:

- **ExpressRoute**: This dedicated private connection offers high bandwidth and low latency.
- **VPN Gateway**: A more cost-effective option for occasional access, but with lower performance compared to ExpressRoute.

Additional Considerations:

- **Service Approval**: Granting access to your Private Link service requires approval for each connection request.
- **Subresource Access**: A single private endpoint connects to one service. If your service has subresources (e.g., storage accounts with blobs and files), you need separate private endpoints for each.

For comprehensive learning, refer to Microsoft's documentation:

- **What is Azure Private Link?**: https://learn.microsoft.com/en-us/training/modules/introduction-azure-private-link/
- **Create a Private Link service**: https://learn.microsoft.com/en-us/training/modules/introduction-azure-private-link/

- **Integrate Azure Private Link with DNS**: There's no official documentation for this specific configuration, but you can combine the concepts from these resources:
    - https://learn.microsoft.com/en-us/azure/dns/private-dns-privatednszone
    - https://learn.microsoft.com/en-us/training/modules/introduction-azure-private-link/
- **Connect Azure to on-premises networks**: https://learn.microsoft.com/en-us/azure/vpn-gateway/

Design and implement service endpoints

Choosing When to Use a Service Endpoint

- **Enhanced Security:** Service endpoints restrict data traffic to the Azure internal network, reducing the attack surface and potential breaches compared to public internet access.
- **Improved Connectivity:** Traffic stays within the Azure backbone, potentially lowering latency and improving overall performance.
- **Compliance Needs:** Certain regulations might require data to remain within a specific geographic boundary. Service endpoints can help achieve such compliance.

Creating Service Endpoints

There are several ways to create service endpoints depending on your preference:

- **Azure Portal:** Use the intuitive interface to configure service endpoints for supported Azure services.
- **Azure Resource Manager (ARM) Templates:** Define service endpoints declaratively for infrastructure as code deployments.
- **Azure PowerShell or Azure CLI:** Leverage these command-line tools for scripting and automation purposes.

Configuring Service Endpoint Policies

Service endpoint policies allow granular control over traffic flow. You can define:

- **Allowed source addresses:** Specify which subnets within your virtual network can access the service endpoint.
- **Service alias:** Assign a custom DNS name within your virtual network for the Azure service endpoint, simplifying resource access.

Configuring Access to Service Endpoints

Once a service endpoint is created, resources within your virtual network can access the Azure service using the configured private IP address or service alias (if defined). Here are some key points:

- **No Public IP Required:** Resources don't need public IP addresses to communicate with the service through the service endpoint.

- **Route Table Configuration:** Ensure proper route tables are configured to direct traffic towards the service endpoint within your virtual network.
- **DNS Resolution:** Update DNS settings within your virtual network to point to the service alias (if used) for seamless resource access.

Additional Considerations:

- Service endpoints are currently not supported for all Azure services. Refer to Microsoft's documentation for a comprehensive list of supported services.
- You can integrate service endpoints with Azure Private Link for even more granular access control and service discovery.

Design and implement Azure network security services (15–20%)

Implement and manage network security groups

Network Security Groups (NSGs) in Azure

NSGs are stateful firewalls that control inbound and outbound network traffic to Azure resources (virtual machines, web apps, etc.) within a virtual network (VNet). They function like security filters, allowing or denying traffic based on pre-defined rules.

Creating a Network Security Group (NSG)

1. Access the Azure portal and navigate to your resource group.
2. Click on "Network security groups" or search for it in the services blade.
3. Select "Add" to create a new NSG.
4. Provide a name and location for your NSG.
5. (Optional) Add tags for organization purposes.
6. Click "Create."

Associating an NSG to a Resource

There are two primary methods for associating an NSG with a resource:

- **During Resource Creation:**
  1. When creating a new resource (e.g., virtual machine), locate the "Networking" section.
  2. Under "Security group," select the NSG you want to associate.
  3. Proceed with creating the resource.
- **After Resource Creation:**
  1. Go to the existing resource (e.g., virtual machine) in the Azure portal.
  2. Navigate to the "Networking" section.
  3. Under "Security group," select the desired NSG.

4. Click "Save" to apply the association.

Application Security Groups (ASGs are deprecated)

Important Note: Microsoft has deprecated Application Security Groups (ASGs) in favor of NSGs with advanced security features. While existing ASGs continue to function, new deployments should utilize NSGs for enhanced capabilities.

Alternative Approach with NSGs

Since ASGs are no longer recommended, here's how to achieve similar functionality with NSGs:

1. Create an NSG with appropriate security rules for your application's specific needs.
2. Associate this NSG with the network interface cards (NICs) attached to the resources requiring application-level security.

Security Rules in NSGs

NSGs control traffic flow using security rules. Each rule specifies:

- **Direction:** Inbound (traffic entering the resource) or Outbound (traffic leaving the resource)
- **Priority:** Lower numbers execute first (important for allowing rules to take precedence over blocking rules)
- **Source:** IP address/range, Azure service tag (e.g., 'Internet'), or another NSG
- **Destination:** Port or port range, IP address/range, or another NSG
- **Protocol:** TCP, UDP, ICMP, or any
- **Action:** Allow or Deny traffic matching the rule's criteria

Best Practices for NSGs

- **Start with Deny-All Default Rule:** Create an initial rule with lowest priority (highest number) that denies all traffic. Subsequent rules can then allow specific traffic based on your application's needs.
- **Least Privilege:** Grant only the minimum access required for your resources to function correctly.
- **Separate NSGs for Different Security Needs:** Consider creating distinct NSGs for various resource groups or tiers (e.g., web servers, databases) to enhance organization and control.
- **Document Your Rules:** Maintain clear documentation of your NSG rules to simplify troubleshooting and future modifications.

Create and Configure NSG Rules:

- **Rules:** An NSG consists of security rules that define what traffic is allowed (permit) or denied (deny) to reach your resources. Each rule specifies:
  - **Direction:** Inbound (to) or outbound (from) the resource.
  - **Priority:** Lower numbered rules are evaluated first (important for allowing specific traffic before a deny-all rule).

- **Protocol:** TCP, UDP, ICMP, or any.
- **Source:** IP address/range, Azure service tag (e.g., 'Internet'), or virtual network.
- **Destination Port:** Specific port (e.g., 80 for HTTP) or a range.
- **Destination:** IP address/range or virtual network.

- **Creating Rules:** You can create NSG rules through the Azure portal, Azure CLI, or PowerShell. The specifics might differ slightly depending on the chosen method.

Interpret NSG Flow Logs:

- **Flow Logs:** NSGs can generate flow logs that provide detailed information about network traffic flow. These logs include:
  - Rule that matched the traffic flow.
  - Source and destination IP addresses.
  - Ports used.
  - Bytes transferred.
  - Flow direction (inbound/outbound).
- **Interpreting Logs:** Flow logs help you troubleshoot connectivity issues, identify suspicious activity, and validate your NSG rules' effectiveness. You can analyze logs in Azure Monitor or export them for further analysis.

Validate NSG Flow Rules:

- **Testing Rules:** After creating NSG rules, it's essential to test them to ensure they function as intended. Here are some methods:
  - **NSG Simulator:** Azure provides an NSG simulator tool to test rules without impacting live traffic.
  - **Test VMs:** Create temporary VMs with specific configurations to test if traffic flows as expected.
  - **Flow Logs:** Analyze flow logs to see if allowed traffic is flowing and denied traffic is blocked.

Verify IP Flow:

- **Verifying Connectivity:** Once you've configured NSG rules, verify that communication between resources is working as expected. You can use tools like ping or remote desktop to test connectivity.
- **Troubleshooting Issues:** If communication fails, analyze NSG rules, flow logs, and resource configurations to identify the cause.

Configure an NSG for Remote Server Administration:

- **Remote Access Scenarios:** There might be situations where you need remote access (RDP/SSH) to VMs secured by NSGs. Here are two options:

  - **Temporary Rule:** Create a temporary NSG rule allowing RDP/SSH traffic from your public IP for a limited duration. This approach is less secure and should be used cautiously.
  - **Azure Bastion:** A more secure approach is to use Azure Bastion. It's a managed service that provides secure RDP/SSH access to VMs without exposing them to the public internet. Bastion uses jump boxes within the virtual network, eliminating the need for public IP addresses on VMs.

Design and implement Azure Firewall and Azure Firewall Manager

Mapping Requirements to Azure Firewall Features:

- **Identify Security Needs:** Before diving into Azure Firewall, clearly define your security requirements. Are you looking to control inbound/outbound traffic, filter specific protocols/ports, or integrate with advanced threat protection solutions?
- **Matching Features:** Once you have your needs, map them to Azure Firewall features. Here are some key capabilities:
  - **Network Traffic Filtering:** Allow or deny traffic based on source/destination IP addresses, ports, protocols, and application security groups.
  - **Next Generation Firewall (NGFW):** Inspect traffic for deep packet inspection (DPI) to identify malware and other threats.
  - **Stateful Inspection:** Track connections to ensure authorized communication.
  - **Dynamic Routing:** Integrate with routing protocols (BGP) for secure communication between on-premises and Azure environments.
  - **High Availability:** Deploy firewalls in an active/passive configuration for redundancy.

Selecting an Azure Firewall SKU:

- **WAF (Web Application Firewall):** Protects web applications from common attacks like SQL injection and cross-site scripting (XSS). Choose this if web application security is a primary concern.
- **Standard:** Offers all core firewall functionalities like traffic filtering, stateful inspection, and basic threat protection. This is a good choice for general-purpose network security.
- **Basic:** Provides basic traffic filtering and stateful inspection. Suitable for simpler deployments with limited security requirements.

Consider these factors when choosing an SKU:

- **Security Needs:** The level of threat protection required (WAF vs Standard vs Basic).
- **Performance:** Throughput and connection capacity needed for your network traffic.
- **Cost:** Pricing varies between SKUs.

Designing an Azure Firewall Deployment:

Here's what to consider when designing your deployment:

- **Network Topology:** Decide whether to use a hub-and-spoke architecture with Azure Virtual WAN or individual virtual networks.
- **Deployment Location:** Place firewalls strategically near your resources for optimal performance.
- **High Availability:** Configure firewalls in an active/passive configuration for redundancy.
- **Network Security Groups (NSGs):** Use NSGs to define granular security policies at the subnet or individual resource level.
- **Azure Firewall Manager (Optional):** For centralized management of multiple firewalls across subscriptions and regions.

Creating and Implementing an Azure Firewall Deployment (Hands-on Labs):

Resources:

- Microsoft provides hands-on labs to practice deploying Azure Firewall. These labs walk you through the creation and configuration process in the Azure portal: "https://learn.microsoft.com/en-us/azure/firewall-manager/"
- Exam prep resources like "https://m.youtube.com/watch?v=WAkOkuaonBI" can also be helpful for understanding the configuration steps.

Additional Tips:

- Leverage Azure Firewall policies to define traffic filtering rules.
- Use Azure Monitor for logging and analyzing network traffic for security insights.
- Integrate Azure Firewall with Azure Security Center for advanced threat protection and vulnerability management.

Configuring Azure Firewall Rules

Azure Firewall acts as a stateful firewall, inspecting incoming and outgoing traffic based on a defined rule set. Here's how to configure these rules:

- **Service Rules:** Define allowed or denied traffic based on Azure services (e.g., allowing access to Storage but denying access to SQL Database).
- **Application Rules:** Granular control by specifying protocols, ports, and source/destination IP addresses or ranges.
- **NAT Rules:** Configure Network Address Translation (NAT) to manage outbound traffic visibility and source IP address presentation.

Creating and Implementing Azure Firewall Manager Policies

Azure Firewall Manager simplifies managing multiple Azure Firewalls across subscriptions and regions. Here's how to create and implement policies:

- **Centralized Policy Management:** Define firewall policies with allowed/denied rules once and apply them to multiple firewalls for consistency.

- **Hierarchical Policies:** Create global policies for organization-wide security baselines and local policies for specific needs of individual teams or workloads.
- **Policy Inheritance:** Local policies inherit rules from global policies, allowing for customization while maintaining a core security posture.

Creating a Secure Hub with Azure Firewall

A secure hub centralizes security policies for a Virtual WAN environment. Here's how to deploy Azure Firewall within a secure hub:

- **Deploy Azure Firewall in a Secured Virtual Hub:** Create a secured virtual hub by deploying Azure Firewall within an Azure Virtual WAN hub. This simplifies traffic routing and policy application.
- **Route Automation:** Firewall Manager automates route configuration for secured virtual hubs, ensuring traffic flows through the firewall for inspection.
- **Centralized Security Management:** Manage security policies and settings for the Azure Firewall within the secure hub from a central location using Firewall Manager.

Additional Resources:

- Microsoft Documentation:
    - What is Azure Firewall?
    - Azure Firewall Manager [invalid URL removed]
- Microsoft Azure Pricing:
    - Azure Firewall Manager Pricing

Design and implement a Web Application Firewall (WAF) deployment

Understanding WAF Features and Capabilities

Before deploying a WAF, it's crucial to understand its features and how they map to your security requirements. Here's a breakdown of key aspects:

- **Rule sets:** These pre-configured rules identify and block common web application attacks like SQL injection, cross-site scripting (XSS), and common web vulnerabilities. Microsoft provides managed rule sets and allows customization for specific needs.
- **Pattern matching:** WAF can identify malicious patterns within HTTP requests and headers, helping to block attacks that exploit specific vulnerabilities.
- **IP reputation blocking:** Based on known malicious IP addresses, WAF can automatically block traffic from these sources.
- **Anomaly detection:** This feature analyzes traffic patterns and identifies deviations from normal behavior, potentially signaling attacks.
- **Rate limiting:** WAF can limit the number of requests from a single IP address to prevent denial-of-service (DoS) attacks.

- **Logging and alerting:** WAF logs all activity and can be configured to generate alerts for suspicious behavior, allowing for investigation and response.

Designing a WAF Deployment

Here's what to consider when designing your WAF deployment:

- **Integration point:** Choose where to integrate WAF. Azure offers options like Azure Application Gateway (AG) and Azure Front Door (AFD). AG provides advanced routing capabilities, while AFD focuses on global performance optimization.
- **High availability:** Ensure your WAF solution is highly available by deploying it across multiple regions or availability zones.
- **Scalability:** Design for future growth. WAF should scale to handle increasing traffic volumes.
- **Logging and monitoring:** Plan how you'll collect, analyze, and store WAF logs. Integrate with Security Information and Event Management (SIEM) tools for centralized visibility.

Detection vs. Prevention Mode

WAF offers two primary modes:

- **Detection mode:** Logs suspicious activity but doesn't block traffic. This allows for analysis before taking action. Useful for initial deployment or for monitoring established rules.
- **Prevention mode:** Actively blocks traffic that violates WAF rules. This is the recommended mode for production environments, but requires careful configuration to avoid unintended blocking of legitimate traffic.

Configuring Rule Sets for WAF on Azure Front Door

Azure Front Door (AFD) integrates natively with WAF. Here's how to configure rule sets:

1. Access your AFD profile in the Azure portal.
2. Navigate to the "WebApplicationFirewall" blade.
3. Select "Managed Rules" and choose the desired rule set (e.g., OWASP 3.x CRS).
4. Optionally, configure custom rules for specific scenarios.
5. Define the detection or prevention mode for the rules.
6. Save your configuration.

Additional Resources

- Microsoft documentation: https://learn.microsoft.com/en-us/azure/web-application-firewall/
- Design and Implement a Web Application Firewall WAF deployment - Part 1 YouTube

Configuring Rule Sets for WAF on Application Gateway:

- **Rule Sets:** These are pre-defined collections of rules that identify and block malicious traffic targeting web applications. Microsoft offers built-in rule sets like the OWASP ModSecurity Core Rule Set 3.3.0.
- **Custom Rules:** You can create custom rules to address specific threats relevant to your application. These rules can be based on patterns in request headers, body content, or cookies.
- **Configuring Rules:**
  - **Match Conditions:** Define what triggers the rule (e.g., specific HTTP methods, URL paths, or header values).
  - **Action:** Specify the action to take when a rule matches (e.g., Block the request, Log the request, or Redirect).

Implementing a WAF Policy:

- A WAF policy is a container for your configured rule sets. It defines how these rules are applied to your web applications.
- You can create different policies with varying rule sets for different applications depending on their security needs.
- Within a policy, you can define the order in which rule sets are evaluated. This helps prioritize certain checks and ensure effective protection.

Associating a WAF Policy:

- Once you have a WAF policy with your desired rule sets, you need to associate it with your Application Gateway.
- This association tells the Application Gateway to enforce the rules defined in the policy on incoming traffic directed to your web applications.

Additional Considerations:

- **Logging:** Enable logging for your WAF policy to analyze attacks and identify potential security breaches.
- **Monitoring:** Monitor the effectiveness of your WAF by reviewing logs and application health metrics.
- **Testing:** Regularly test your WAF rules to ensure they don't block legitimate traffic.

Resources for further learning:

- Microsoft documentation on Azure WAF: https://learn.microsoft.com/en-us/azure/web-application-firewall/
- Microsoft Cloud Academy course on AZ-700: Microsoft Cloud Academy (Subscription required)
- Video tutorial on WAF deployment for AZ-700: YouTube

- For a full set of 600 questions. Go to https://skillcertpro.com/product/microsoft-azure-network-engineer-az-700-exam-questions/
- SkillCertPro offers detailed explanations to each question which helps to understand the concepts better.
- It is recommended to score above 85% in SkillCertPro exams before attempting a real exam.
- SkillCertPro updates exam questions every 2 weeks.
- You will get life time access and life time free updates

SkillCertPro assures 100% pass guarantee in first attempt.

Disclaimer: All data and information provided on this site is for informational purposes only. This site makes no representations as to accuracy, completeness, correctness, suitability, or validity of any information on this site & will not be liable for any errors, omissions, or delays in this information or any losses, injuries, or damages arising from its display or use. All information is provided on an as-is basis.