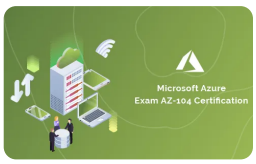


Level: Advanced

Microsoft Azure Exam AZ-104 Certification

Back to the **Course**

Azure Virtual Networks - **Practice Mode** Completed on Sun, 26 Oct 2025

1st	2/6	33.33%	FAIL
Attempt	Marks Obtained	Your Score	Result

Download Report

Domain wise Quiz Performance Report

No.	Domain	Total Question	Correct	Incorrect	Unattempted	Marked for Review
1	Implement and manage virtual networking	6	2	4	0	0
Total	All Domains	6	2	4	0	0

Review the Answers

Filter By

Question 1 Incorrect

Domain: Implement and manage virtual networking

Your organization is deploying various types of load balancers in an Azure environment, each serving different purposes. Each load balancer requires a specific type of IP address configuration. Match each load balancer scenario with the most appropriate public or private IP configuration.

Scenarios:

A public-facing load balancer with multiple backend virtual machines.

An internal load balancer managing traffic across private subnets.

A cross-region load balancer for a global application.

A load balancer requires sticky sessions for consistent client-server mapping.

Note: Match the load balancer scenarios with their correct public IP allocation method. (Check both the tables correctly)

Your Answers	Correct Answers
A. Standard SKU Static Public IP Load balancer with sticky sessions	A. Standard SKU Static Public IP Public-facing load balancer
B. Standard SKU Dynamic Public IP Cross-region load balancer	B. Standard SKU Dynamic Public IP Cross-region load balancer
C. Basic SKU Public IP Public-facing load balancer	C. Basic SKU Public IP Load balancer with sticky sessions
D. Private IP Internal load balancer	D. Private IP Internal load balancer

Explanation:

Correct Answers: 1-A, 2-C, 3-D and 4-B

Public-facing load balancer ↔ Standard SKU Static Public IP

Internal load balancer ↔ Private IP

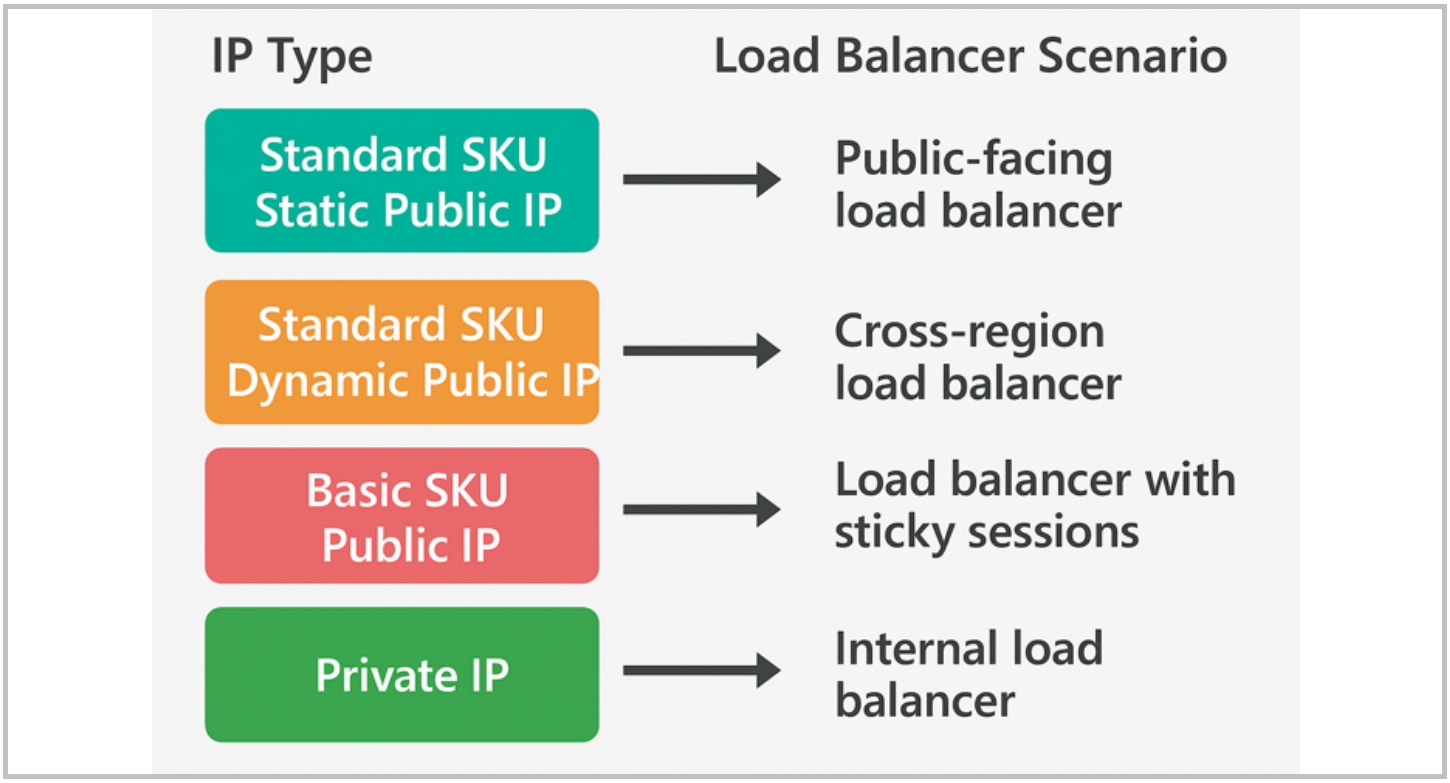
Cross-region load balancer ↔ Standard SKU Dynamic Public IP

Load balancer with sticky sessions ↔ Basic SKU Public IP

1. **Public-facing load balancer** requires a **Standard SKU Static Public IP.** This is the best choice because a public-facing load balancer distributes internet traffic, and a static IP address ensures a consistent, unchanging address. This is critical for DNS

configurations and for high-availability, high-traffic applications.

2. **Internal load balancer** requires a **Private IP**. An internal load balancer operates exclusively within a virtual network and is not exposed to the public internet. Therefore, it uses a private IP address for traffic management between resources within private subnets, enhancing security by keeping traffic isolated.
3. **Cross-region load balancer** requires a **Standard SKU Dynamic Public IP.** Cross-region load balancers are designed for globally distributed applications. The dynamic nature of the IP allows it to adapt to routing changes and efficiently redirect traffic in response to regional failures or latency shifts, which is essential for high availability and resilience across multiple regions.
4. **Load balancer with sticky sessions** requires a **Basic SKU Public IP**. Sticky sessions (or session persistence) ensure that a client's requests are consistently routed to the same backend instance. For this purpose, the Basic SKU is a cost-effective option that provides the necessary functionality without the advanced features of the Standard SKU, making it suitable for smaller-scale applications.



References:

https://learn.microsoft.com/en-us/azure/virtual-network/ip-services/public-ip-addresses#static

https://learn.microsoft.com/en-us/azure/load-balancer/load-balancer-overview#internal-load-balancer

https://learn.microsoft.com/en-us/azure/load-balancer/cross-region-overview#standard-sku

https://learn.microsoft.com/en-us/azure/load-balancer/distribution-mode-concepts#session-persistence

Ask our Experts

Did you like this **Question?**

Question 2 Correct

Domain: Implement and manage virtual networking

A web app hosted in an Azure VM is unreachable from other VNets in the same region, even though VNet peering is configured. What steps should you take to troubleshoot? (Select two options)

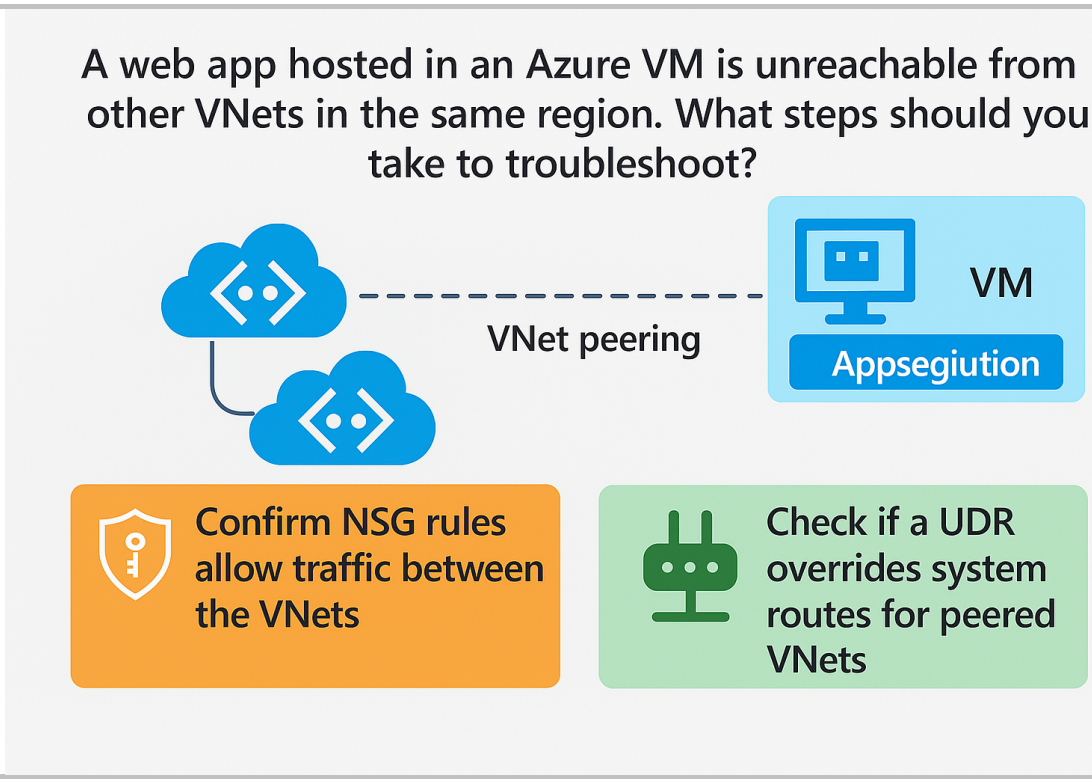
- [x] A. Confirm NSG rules allow traffic between the VNets right
- [x] B. Check if a UDR overrides system routes for peered VNets right
- [ ] C. Enable gateway transit on the peered VNets
- [ ] D. Verify DNS resolution for the web app's hostname
- [ ] E. Associate the VM's NIC with a public IP address

Explanation:

Correct Answers: A and B

Option A: Confirm NSG rules allow traffic between the VNets is correct because Network Security Groups (NSGs) control inbound and outbound traffic to subnets and virtual machines (VMs). When VNets are peered, the traffic between them must comply with NSG rules on both ends. If the NSG rules block traffic from the source or to the destination VNet, communication will fail. For example, NSG rules might allow only specific source IP ranges or block traffic on required ports. You need to ensure that the NSG allows the appropriate protocol (e.g., TCP) and port (e.g., 80 or 443 for a web app) for communication between the VNets. Additionally, NSG logs can be reviewed in Azure Monitor to confirm whether traffic is being denied. This helps identify misconfigurations.

Option B: Check if a UDR overrides system routes for peered VNets is correct because user-defined routes (UDRs) allow custom routing for traffic within a VNet, including traffic to peered VNets. However, a misconfigured UDR can override the system route that directs traffic to a peered VNet. For instance, if a UDR is configured to route traffic for the target VNet's address space to an invalid next hop (e.g., a non-existent virtual appliance), communication will fail. You need to validate the route table associated with the source subnet and ensure that the route for the peered VNet has "Virtual Network" as its next hop. Azure's effective routes feature can be used to inspect the routes applied to a VM's NIC.



Option C: Enable gateway transit on the peered VNets is incorrect because gateway transit is used to share a virtual network gateway across peered VNets for connecting to on-premises resources or other VNets. It is not required for direct communication between resources in peered VNets within the same region. Enabling gateway transit would have no impact on resolving the communication issue described in this scenario.

Option D: Verify DNS resolution for the web app's hostname is incorrect because DNS resolution is necessary for resolving hostnames to IP addresses. However, the issue described is about the inability to communicate between VNets, not an inability to resolve DNS. Even if DNS is not configured, direct communication using IP addresses should still be possible if the routing and NSG rules are correctly configured.

Option E: Associate the VM's NIC with a public IP address is incorrect because a public IP address is not required for communication between peered VNets. Peered VNets use Azure's backbone network for connectivity, which is private and does not involve the public Internet. Associating a public IP with the VM is unnecessary and does not address the root cause of the problem.

References:

https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview

https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview

Ask our Experts

Did you like this **Question?**

Question 3 Incorrect

Domain: Implement and manage virtual networking

You are configuring a complex set of rules for an Application Security Group (ASG) to restrict inbound access to multiple Azure virtual machines (VMs). The VMs reside in different subnets of the same virtual network. Each VM is configured to belong to a unique ASG based on its function (e.g., web, application, database). While testing connectivity, you notice that the web VMs cannot access the application VMs, even though the ASG rules should allow the traffic. What is the most likely issue?

A. The NSG associated with the web VMs is missing an outbound rule to permit traffic to the application VMs

B. The ASG is not correctly configured to allow cross-subnet communication between the web and application tiers wrong

C. The application VMs have conflicting DNS settings that prevent them from resolving the web VM IP addresses

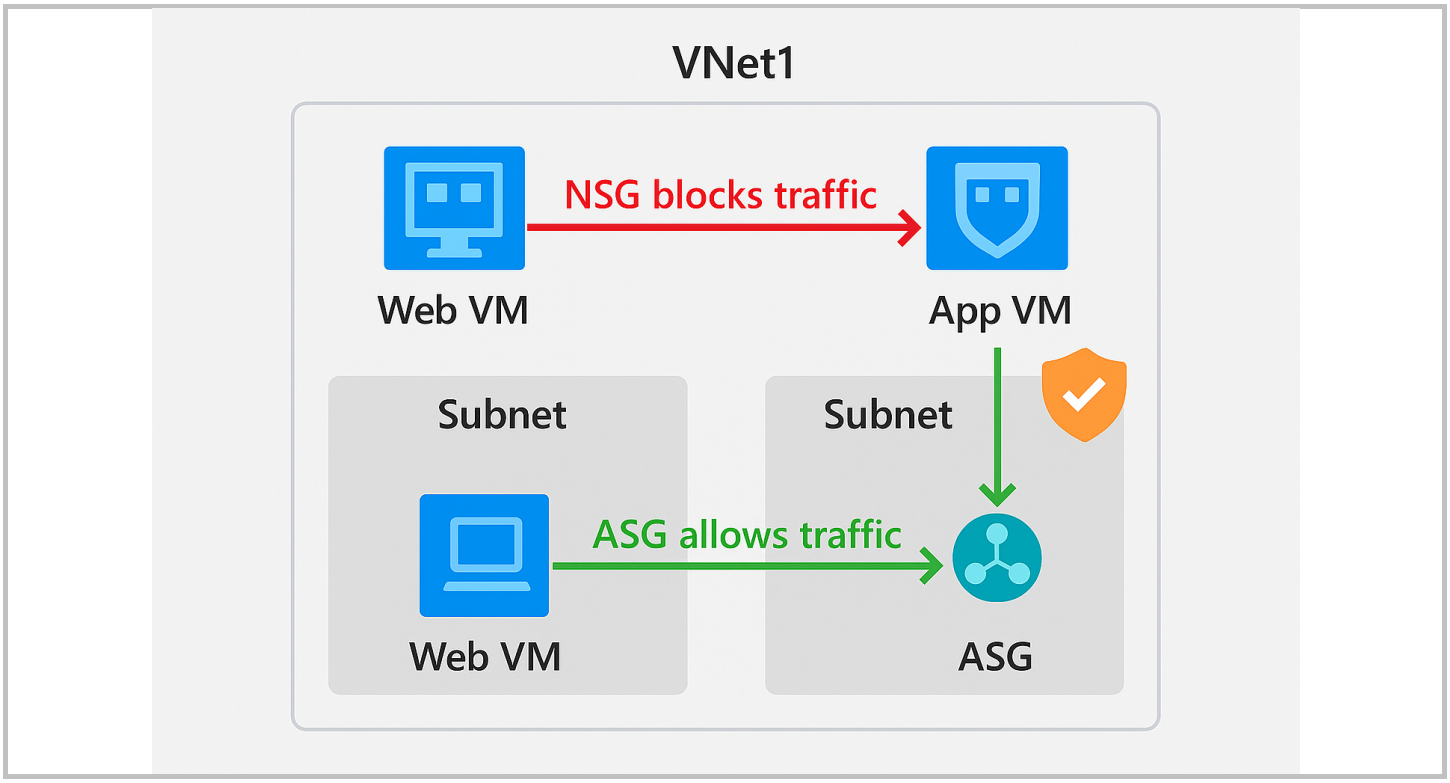
D. The NSG rules for the application VMs override the ASG rules, blocking communication right

---

Explanation:

Correct Answer: D

Option D: The NSG rules for the application VMs override the ASG rules, blocking communication is correct. Network Security Groups (NSGs) are processed before Application Security Groups (ASGs). An NSG applied to a subnet or network interface controls traffic at a granular level. If an NSG rule on the application VMs' subnet or network interface explicitly denies traffic from the web VMs, that rule will take precedence and block the traffic, regardless of any ASG rules that might be configured to allow it. This is the most common reason for this type of misconfiguration.



Option A: The NSG associated with the web VMs is missing an outbound rule to permit traffic to the application VMs is incorrect. The problem described is that web VMs cannot access the application VMs, which points to an inbound access issue on the destination

(application) VMs. The outbound rules on the source (web) VMs would only prevent them from initiating the traffic, not receiving a response, which is a different issue.

Option B: The ASG is not correctly configured to allow cross-subnet communication between the web and application tiers is incorrect. ASGs are designed to simplify security management and do not, by themselves, restrict communication across subnets. The default behavior in a virtual network is to allow communication between subnets unless an NSG explicitly blocks it.

Option C: The application VMs have conflicting DNS settings that prevent them from resolving the web VM IP addresses is incorrect. DNS issues relate to name resolution, not the blocking of network traffic itself. If the VMs were using IP addresses for communication, DNS would not be a factor, yet the traffic would still be blocked. The issue is with traffic flow, not name resolution.

References:

https://learn.microsoft.com/en-us/azure/virtual-network/network-security-groups-overview

https://learn.microsoft.com/en-us/azure/virtual-network/application-security-groups

Ask our Experts

Did you like this **Question?**

Question 4

Incorrect

Domain: Implement and manage virtual networking

You are managing a virtual network (VNet) named “ProdVNet” with three subnets: “WebSubnet”, “AppSubnet”, and “DbSubnet”. The “WebSubnet” needs to route all outbound traffic to the internet via a Network Virtual Appliance (NVA) with an IP address of 10.0.1.4. The “AppSubnet” should route traffic destined for “DbSubnet” directly without any intermediary devices. You must configure user-defined routes (UDRs) to implement this design. Which actions should you take to achieve the configuration? (Select three)

- [x] A. Create a route table named “NvaRouteTable”, add a route with the destination prefix 0.0.0.0/0, and set the next hop type to Virtual Appliance with the next hop address 10.0.1.4 right
- [x] B. Associate “NvaRouteTable” with the “WebSubnet” right
- [ ] C. Create a route table named “AppToDbRouteTable”, add a route with the destination prefix 10.0.3.0/24, and set the next hop type to VNet Peering
- [x] D. Associate “AppToDbRouteTable” with the “AppSubnet” wrong
- [ ] E. Ensure that the NVA is configured with IP forwarding enabled and placed in the same subnet as “WebSubnet” right

Explanation:

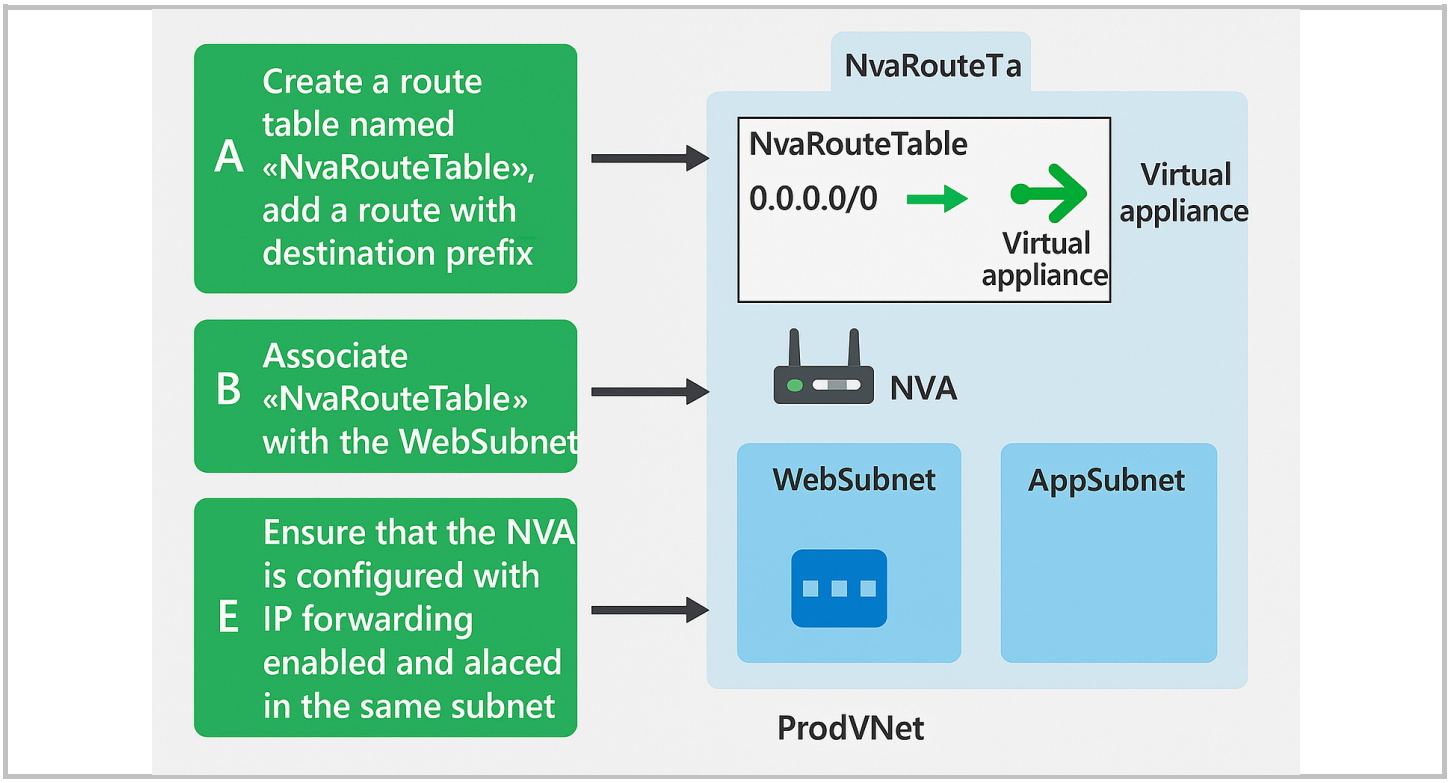
Correct Answers: A, B and E

Option A: Create a route table named “NvaRouteTable”, add a route with the destination prefix 0.0.0.0/0, and set the next hop type to Virtual Appliance with the next hop address 10.0.1.4 is correct. This step correctly defines a User-Defined Route (UDR) to force all

outbound internet traffic from the "WebSubnet" through a specific network appliance. The 0.0.0.0/0 prefix represents the default route for all traffic not covered by other routes, and specifying the next hop as a Virtual Appliance with its private IP address ensures that traffic is directed to the NVA for inspection.

Option B: Associate "NvaRouteTable" with the "WebSubnet" is correct. A route table's rules only take effect when it is associated with a specific subnet. By linking "NvaRouteTable" to the "WebSubnet," all resources within that subnet will use the defined UDR to route their internet-bound traffic through the NVA, as required.

Option E: Ensure that the NVA is configured with IP forwarding enabled and placed in the same subnet as "WebSubnet" is correct. For the NVA to function as a router, IP forwarding must be enabled. This allows it to forward traffic not addressed to its own IP. Additionally, the NVA must be in the same subnet as the "WebSubnet" to be a valid next hop for the UDR. Without this configuration, traffic would be dropped at the NVA, and the routing would fail.



Option C: Create a route table named "AppToDbRouteTable", add a route with the destination prefix 10.0.3.0/24, and set the next hop type to VNet Peering is incorrect. This is unnecessary and incorrect. Both "AppSubnet" and "DbSubnet" are in the same VNet ("ProdVNet"). Azure's default routing behavior allows direct communication between subnets within the same VNet, so no UDR is needed. VNet Peering is only used for communication between different VNets.

Option D: Associate "AppToDbRouteTable" with the "AppSubnet" is incorrect. This action is not needed because the required communication between "AppSubnet" and "DbSubnet" is handled by Azure's default routing. Creating and associating a UDR for this purpose would be redundant and potentially cause routing conflicts.

Reference:

https://learn.microsoft.com/en-us/azure/virtual-network/virtual-networks-udr-overview

Ask our Experts

Did you like this **Question?**

Question 5 Incorrect

Domain: Implement and manage virtual networking

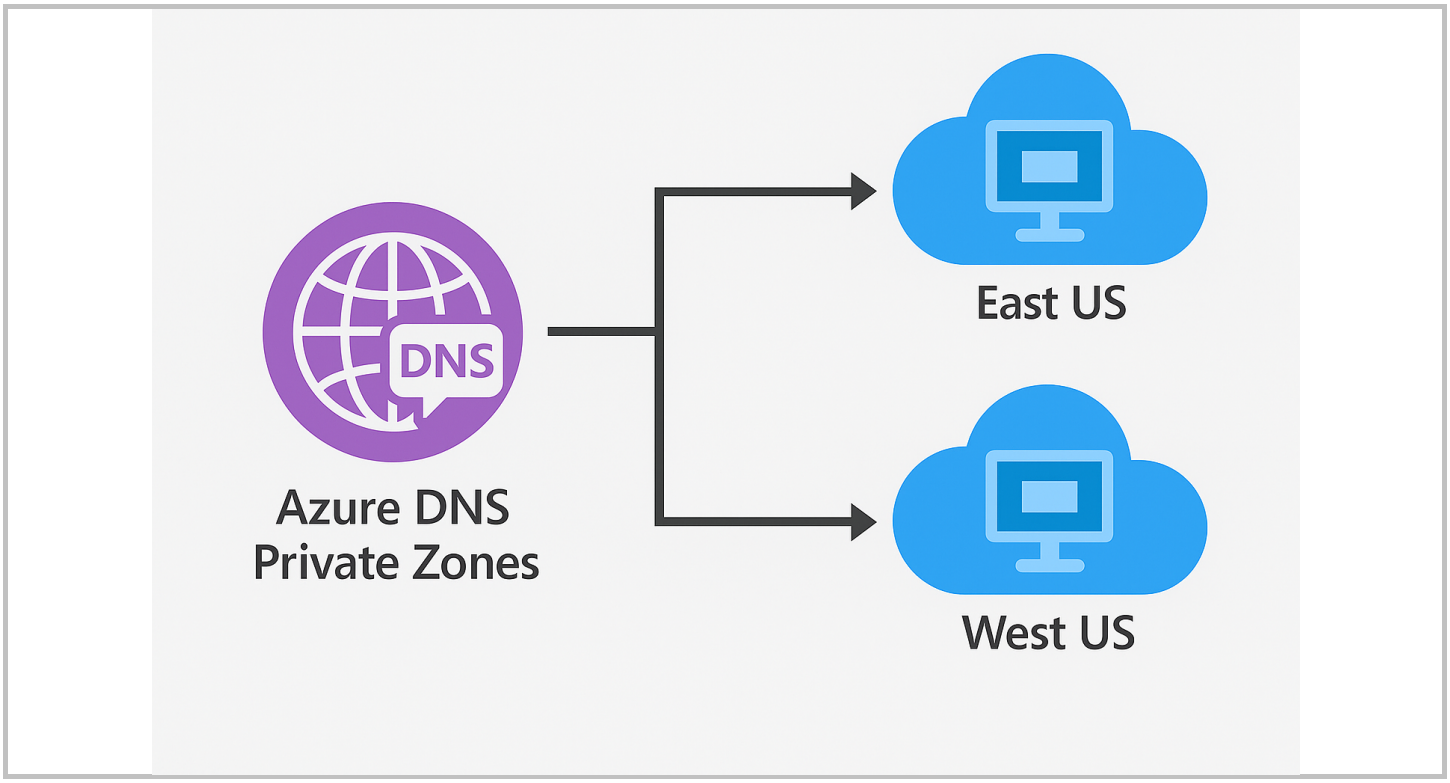
You are tasked with configuring DNS settings for a multi-region Azure deployment. The organization has two Azure regions: East US and West US. A global application with backend services spread across both regions requires internal name resolution between the regions. What feature of Azure DNS should you configure to ensure consistent and accurate name resolution across both regions?

- A. Azure DNS with Geo-location based routing wrong
- B. Azure DNS Private Zones with VNet linking right
- C. Azure DNS Public Zones for regional endpoints
- D. Azure Load Balancer with DNS resolution configuration

Explanation:

Correct Answer: B

Option B: Azure DNS Private Zones with VNet linking is correct because Azure DNS Private Zones allow you to manage DNS records for your private domains within your Azure environment. By linking Azure DNS Private Zones to multiple VNets, you enable internal DNS resolution across regions. This configuration allows resources in different Azure regions, like East US and West US, to resolve each other's names accurately and consistently, ensuring proper internal communication between services spread across regions. Linking private DNS zones to multiple VNets means that each VNet (whether in East US or West US) can use the same DNS zone for name resolution, providing a seamless experience for internal resources. This solution is ideal for multi-region deployments where you need private name resolution between services without relying on external DNS solutions.



Option A: Azure DNS with Geo-location based routing is incorrect because Azure DNS does not directly support Geo-location based routing for internal DNS resolution between regions. Geo-location routing is typically a feature provided by Azure Traffic Manager for global traffic routing, not Azure DNS. It helps route traffic based on geographic location, but it is not designed for internal name resolution across multiple Azure regions. Geo-location based routing would be relevant for public traffic that needs to be directed to the nearest region, but internal DNS resolution for resources in different regions is typically handled through Private DNS Zones in Azure. Hence, this option does not fit the requirement of internal name resolution across regions in the context of Azure DNS.

Option C: Azure DNS Public Zones for regional endpoints is incorrect because Azure DNS Public Zones are intended for public DNS resolution, meaning they handle domain name resolution for resources that are accessible from the internet. In the case of multi-region deployments, public zones are used for services exposed to the public internet, such as web apps or APIs. Since the requirement is for internal name resolution between regions (East US and West US), using public DNS zones is not appropriate. Private DNS Zones are designed for this exact purpose, providing DNS resolution within the Azure environment without exposing internal services to the public internet.

Option D: Azure Load Balancer with DNS resolution configuration is incorrect because while Azure Load Balancer can help distribute traffic between backend resources, it does not provide the DNS resolution required for internal communication between resources in different regions. Azure Load Balancer operates within a single region and does not offer DNS-based routing for multiple regions. Although Load Balancer can be used in conjunction with DNS settings for public-facing services (for example, resolving the domain to the correct Load Balancer IP), it is not the correct solution for internal DNS resolution between Azure regions. DNS resolution for multi-region internal traffic must be handled by Azure DNS Private Zones, not Azure Load Balancer.

References:

https://learn.microsoft.com/en-us/azure/dns/private-dns-overview

https://learn.microsoft.com/en-us/azure/dns/private-dns-virtual-network-links

Ask our Experts

Did you like this **Question?**

Question 6 Correct

Domain: Implement and manage virtual networking

A company deployed an Azure virtual machine scale set (VMSS) behind an Azure Load Balancer to distribute HTTP traffic. Users report intermittent connection failures when accessing the application. Upon inspection, it is found that health probes intermittently fail in some instances. The configuration shows that the probe interval is set to 5 seconds, and the unhealthy threshold is set to 2.

Proposed Solution: Increase the probe interval to 15 seconds and the unhealthy threshold to 5, ensuring longer timeouts for health checks. This will stabilize the connection and resolve the issue. Is this proposed solution correct? (Select Yes or No)

A. Yes

B. No right

Explanation:

Correct Answer: B

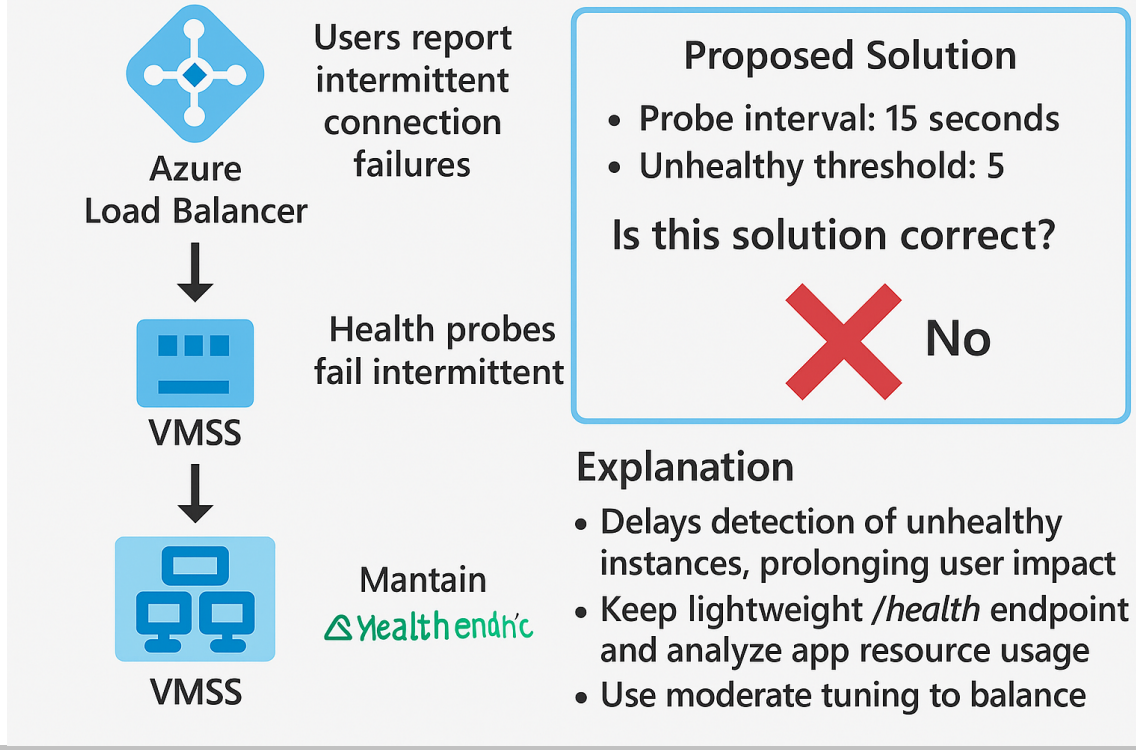
The proposed solution is incorrect because increasing the probe interval and unhealthy threshold does not address the root cause of the intermittent connection failures. Instead, it only delays the detection of unhealthy instances. This can lead to a degraded user experience as the load balancer continues to send traffic to instances that are not responding correctly.

Intermittent failures are often caused by underlying application issues such as:

Misconfigured health probe endpoints

Insufficient resource allocation

Performance bottlenecks



The correct approach is to identify and fix these underlying problems. A more effective solution would be to ensure the health probe endpoint is lightweight and reliable. If a change to the probe settings is necessary, a moderate adjustment would be better than an excessive one, which would sacrifice the load balancer's responsiveness.

Reference:

Azure Load Balancer health probes | Microsoft Learn

Ask our Experts

Did you like this **Question?**

Finish Review