VIRTUAL PRIVATE CLOUD (VPC) - QUIZ

Attempt	3	**Completed on**	Friday , 18 January 2019 , 09:32 PM
Marks Obtained	23 / 25	**Time Taken**	00 H 28 M 51 S
Your score is	92%	**Result**	Pass

Domains / Topics wise Quiz Performance Report

S.No.	Topic	Total Questions	Correct	Incorrect	Unattempted
1	Other	25	23	2	0

25 Questions	23 Correct	2 Incorrect	0 Unattempted

Show Answers: All

QUESTION 1 CORRECT

You are working as an architect in your organization. You have peered VPC A as requester and VPC B as accepter and both VPCs can communicate with each other. Now you want resources in both the VPCs to reach out to internet but anyone on internet should not be able to reach resources within VPC. Which of the following statements is true?

- **A.** Create a NAT Gateway on Requester VPC (VPC A) and configure a route in Route table with NAT Gateway. VPC B can route to internet through VPC A NAT Gateway.
- **B.** Create an Internet Gateway on Requester VPC (VPC A) and configure a route in Route table with Internet Gateway. VPC B can route to internet through VPC A Internet Gateway.
- **C.** Create NAT Gateways on both VPCs and configure routes in respective route tables with NAT Gateways. ✔

D. Create a NAT instance on Requester VPC (VPC A) . VPC B can route to internet through VPC A NAT Instance.

Explanation :

Correct Answer : C

For Option A, when NAT Gateway and configured for VPC A, the resources within VPC A can reach out to internet. But, VPC B resources cannot reach to internet through NAT Gateway created in VPC A although both VPCs are peering. This situation would cause transitive routing which is not supported in AWS routing.

Using a NAT Gateway with VPC Endpoints, VPN, AWS Direct Connect, or VPC Peering

A NAT gateway cannot send traffic over VPC endpoints, VPN connections, AWS Direct Connect, or VPC peering connections. If your instances in the private subnet must access resources over a VPC endpoint, a VPN connection, or AWS Direct Connect, use the private subnet's route table to route the traffic directly to these devices.

For example, your private subnet's route table has the following routes: internet-bound traffic (0.0.0.0/0) is routed to a NAT gateway, Amazon S3 traffic (pl-xxxxxxxx; a specific IP address range for Amazon S3) is routed to a VPC endpoint, and 10.25.0.0/16 traffic is routed to a VPC peering connection. The pl-xxxxxxxx and 10.25.0.0/16 IP address ranges are more specific than 0.0.0.0/0; when your instances send traffic to Amazon S3 or the peered VPC, the traffic is sent to the VPC endpoint or the VPC peering connection. When your instances send traffic to the internet (other than the Amazon S3 IP addresses), the traffic is sent to the NAT gateway.

You cannot route traffic to a NAT gateway through a VPC peering connection, a VPN connection, or AWS Direct Connect. A NAT gateway cannot be used by resources on the other side of these connections.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-basics (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-%20gateway-basics)

For Option B, Internet Gateways are for two way traffic. But the requirement is only for resouces to reach out to internet, inbound traffic from internet should not be allowed. So Internet Gateway is not correct choice.

For Option D, similar to Option A, this situation would cause transitive peering and hence not supported.

Note: AWS recommends using NAT Gateway over NAT Instance

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-comparison.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-comparison.html)

Ask our Experts

QUESTION 2 CORRECT

Your organization already had a VPC(10.10.0.0/16) setup with one public (10.10.1.0/24) and two private subnets – private subnet 1 (10.10.2.0/24) and private subnet 2 (10.10.3.0/24). The public subnet has the main route table and two private subnets have two different route tables respectively. AWS sysops team reports a problem stating the EC2 instance in private subnet 1 cannot communicate to RDS MySQL database which is on private subnet 2. What are the possible reasons? (choose 2 options)

- [ ] **A.** One of the private subnet route table's local route has been changed to restrict access only within the subnet IP range.
- [x] **B.** RDS security group inbound rule is incorrectly configured with 10.10.1.0/24 instead of 10.10.2.0/24. ✔
- [x] **C.** 10.10.3.0/24 subnet's NACL is modified and it does not have an inbound ALLOW rule set for ALL Traffic. ✔
- [ ] **D.** RDS Security group outboud does not contain a rule for ALL traffic or port 3306 for 10.10.2.0/24 IP range.

Explanation :

Correct Answer: B, C

For Option A, for any route table local route cannot be edited or deleted.
AWS Docs says:

"Every route table contains a local route for communication within the VPC over IPv4. If your VPC has more than one IPv4 CIDR block, your route tables contain a local route for each IPv4 CIDR block. If you've associated an IPv6 CIDR block with your VPC, your route tables contain a local route for the IPv6 CIDR block. You cannot modify or delete these routes."

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html#RouteTables
(https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html#RouteTa%20bles)

For Option B, possible because security group is configured with public subnet IP range instead of private subnet 1 IP range and EC2 is in private subnet 1. So EC2 will not be able to communicate with RDS in private subnet 2.

For Option C is correct.

Inbound					
Rule #	Type	Protocol	Port Range	Source	Allow/Deny
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#default-network- acl (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#default-network-%20acl)

Option D is not correct because Security Groups are stateful - if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSecurityGroups (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSe%20curityGrou

Ask our Experts

QUESTION 3 CORRECT

A new VPC with CIDR range 10.10.0.0/16 has been setup. Internet Gateway and new route table have been created and a new route has been added with internet gateway as target and 0.0.0.0/0 as a destination. Two subnets have been created, one for public and one for private. A new Linux EC2 instance has been launched on the public subnet with Auto-assign Public IP option enabled. But when trying to SSH to the new machine, the connection is getting failed. What could be the reason?

- **A.** Elastic IP is not assigned.
- **B.** Both the subnets are associated with Main route table, no subnet is explicitly associated with new route table which has internet gateway route. ✔
- **C.** Public IP address is not assigned.
- **D.** None of the above.

Explanation :

Answer: B

Option A, An Elastic IP address is a public IPv4 address with which you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

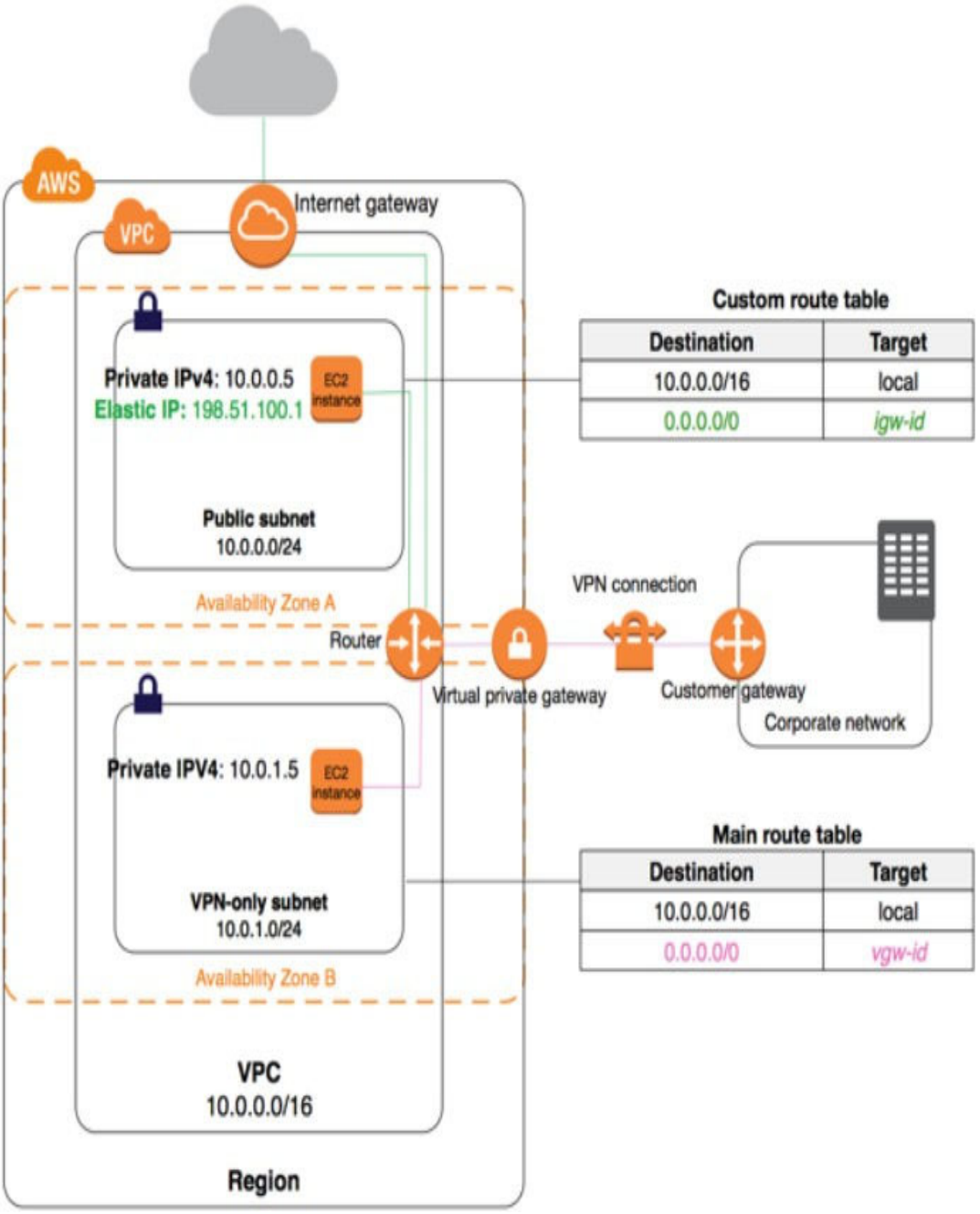
If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet; for example, to connect to your instance from your local computer.

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#eip-basics (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html#eip-%20basics)

From our problem statement, EC2 is launched with Auto-assign public IP enabled. So, since public IP is available, Elastic IP is not a necesity to connect from internet.

Option C, the problem statement clearly states that EC2 is launched with Auto-assign Public IP enabled, so this option cannot be true.

Option B, whenever a subnet is created, by default, it is associated with main route table. We need to explicitly associate the subnet to new non-main route table if different routes are required for main and non-main route tables.



- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html#RouteTables (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html#RouteTa%20bles)

Ask our Experts

QUESTION 4 CORRECT

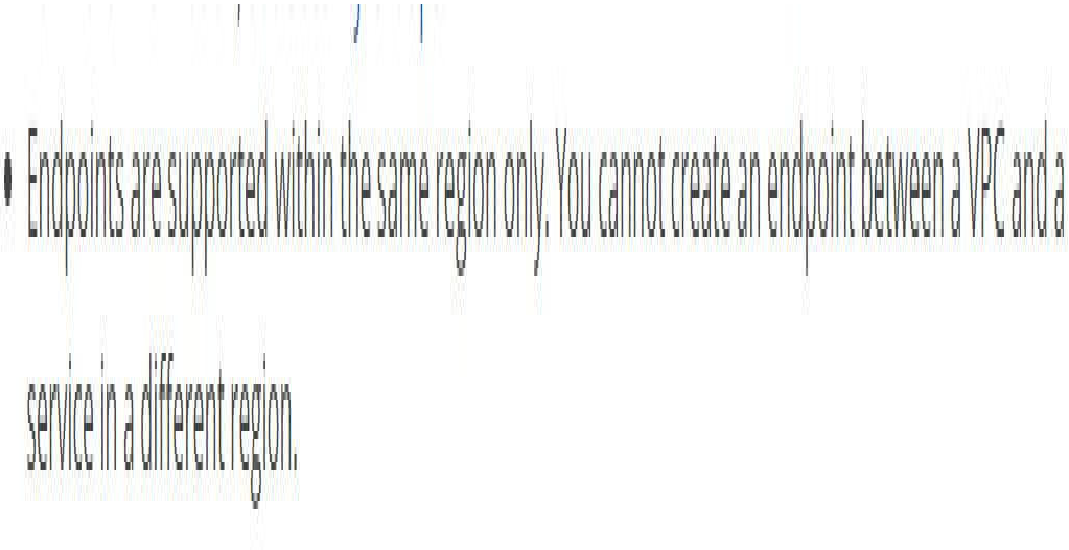
You are an architect in your organization. Your organization would want to upload files to AWS S3 bucket privately through AWS VPC. In an existing VPC, you already have a subnet and route table which contains a route to NAT Gateway. You have created VPC Endpoint for S3 and added same route table. But in AWS S3 server logs you noticed that the request to S3 bucket from an EC2 instance within the subnet associated with above route table are going to internet through NAT Gateway. What could be causing this situation?

- **A.** When NAT Gateway and VPC endpoint exist on same route table, NAT Gateway always takes precedence.
- **B.** EC2 instance is having an elastic IP address associated with it.
- **C.** AWS S3 bucket is in different region than the VPC. ✔
- **D.** AWS S3 is a managed service, all requests will always go through internet.

Explanation :

Answer: C

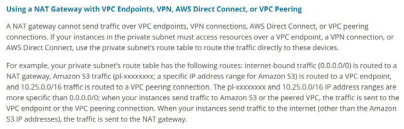
Option A, the opposite is true. VPC Endpoint always takes precedence over NAT Gateway or Internet Gateway. In the absence of VPC endpoint, requests to S3 are routed to NAT Gateway or Internet Gateway based on their existence in route table.



Option B, elastic IP address is IPv4 public address with which you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

Elastic Ips are not used for routing requests from an EC2 instance.

Option C, VPC endpoints does not support cross-region S3 requests. So, if S3 bucket is in different region than the VPC, then VPC endpoint does not take effect. In this case, requests will go through NAT Gateway as the route table has a route to it.



Using a NAT Gateway with VPC Endpoints, VPN, AWS Direct Connect, or VPC Peering

A NAT gateway cannot send traffic over VPC endpoints, VPN connections, AWS Direct Connect, or VPC peering connections. If your instances in the private subnet must access resources over a VPC endpoint, a VPN connection, or AWS Direct Connect, use the private subnet's route table to route the traffic directly to these devices.

For example, your private subnet's route table has the following routes: internet-bound traffic (0.0.0.0/0) is routed to a NAT gateway, Amazon S3 traffic (pl-xxxxxxxx; a specific IP address range for Amazon S3) is routed to a VPC endpoint, and 10.25.0.0/16 traffic is routed to a VPC peering connection. The pl-xxxxxxxx and 10.25.0.0/16 IP address ranges are more specific than 0.0.0.0/0; when your instances send traffic to Amazon S3 or the peered VPC, the traffic is sent to the VPC endpoint or the VPC peering connection. When your instances send traffic to the internet (other than the Amazon S3 IP addresses), the traffic is sent to the NAT gateway.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Appendix_Limits.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Appendix_Limits.html)

Option D is false. VPC Endpoint helps to route traffic internally within AWS network without the need to go over through internet. This makes your S3 bucket private to your network. For more information, refer VPC endpoint documentation

- https://aws.amazon.com/blogs/aws/new-vpc-endpoint-for-amazon-s3/ (https://aws.amazon.com/blogs/aws/new-vpc-endpoint-for-amazon-s3/)

Note:
As per our requirement we need to upload files to S3 bucket privately and for that requirement we have created VPC Endpoints for S3. However we noticed that the S3 server logs are indicating that the traffic is flowing through the internet rather than using VPC Endpoint connection.

As per AWS,
S3 VPC endpoints doesn't support cross region requests.
When you create a VPC endpoint for Amazon S3, any requests to an **Amazon S3 endpoint within the Region** (for example, *s3.us-west-2.amazonaws.com*) are routed to a private Amazon S3 endpoint within the Amazon network. You don't need to modify your applications running on EC2 instances in your VPC—the endpoint name remains the same, but the route to Amazon S3 stays entirely within the Amazon network, and does not access the public internet.

For more information please refer:
https://docs.aws.amazon.com/glue/latest/dg/vpc-endpoints-s3.html (https://docs.aws.amazon.com/glue/latest/dg/vpc-endpoints-s3.html)

Ask our Experts

QUESTION 5 **CORRECT**

Your organization has an existing VPC with an AWS S3 VPC endpoint created and serving certain S3 buckets. You were asked to create a new S3 bucket and reuse existing VPC endpoint to route requests to new S3 bucket. However, after creating new S3 bucket and sending requests from an EC2 instance via VPC endpoint, you found the requests are failing with "Access Denied" error. What could be the issue? (select 2 options)

- [x] **A.** VPC endpoint contains a policy, currently restricted to certain S3 buckets and does not contain new S3 bucket. ✔
- [x] **B.** AWS IAM role/user does not have access to new S3 bucket. ✔
- [ ] **C.** AWS default Deny policy restricting access to IAM role/user who is already having access to S3 bucket.
- [ ] **D.** You need to add new S3 bucket host name as destination and VPC endpoint ID as target in route table in order to route requests to new S3 bucket.

Explanation :

Answer: A, B

Option A is correct, VPC endpoint has a policy which by default allows all actions on all S3 bucket. We can restrict access to certain S3 buckets and certain actions on this policy. In such cases, for accessing any new buckets or for any new actions, VPC endpoint policy needs to be modified accordingly.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints-access.html#vpc- endpoint-policies (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints-access.html#vpc-endpoint-policies)

Option B is correct, AWS IAM role/user which is used to access S3 bucket need to have access granted via IAM policy before accessing. So if the IAM role/user is not an administrator or not having full S3 access, newly created S3 bucket must be added to the IAM policy.

- https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html (https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-access-control.html)

Option C is incorrect, by default there is no resource policy on S3 bucket. If we would like to make the bucket private, we can add a new resource policy with "Deny". Please see the documentation for more information

- https://aws.amazon.com/blogs/security/how-to-create-a-policy-that-whitelists-access-to-sensitive-amazon-s3-buckets/ (https://aws.amazon.com/blogs/security/how-to-create-a-policy-that-whitelists-access-to-sensitive-amazon-s3-buckets/)

Option D is incorrect, route for VPC endpoint is defined for all S3 requests within same region as VPC irrespective of buckets. So, we will only have one route per VPC endpoint.

Ask our Experts

QUESTION 6 CORRECT

Following are the Network ACL rules defined in your subnet which is associated with a route
table which has Internet Gateway.

Inbound Rules:

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
200	SSH (22)	TCP (6)	22	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Outbound rules:

Rule#	Type	Protocol	Port range	Destination	Allow / Deny
100	SSH (22)	TCP (6)	22	0.0.0.0/0	DENY
200	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

What would be the outcome when an administrator tries to access an EC2 instance launched into this public subnet from his corporate network using SSH?

- **A.** SSH request would fail due to specific DENY on SSH protocol.

- **B.** SSH request would fail because of DENY on all traffic with Rule # as *
- **C.** SSH request would succeed because inbound rule # 100 has ALLOW for ALL traffic. ✔
- **D.** SSH request would fail due to outbound rule # 100 has DENY for SSH traffic.

Explanation :

Answer - C

A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.
https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules)
For Option A, SSH DENY rule is 200, but rule # 100 is ALLOW ALL which would allow traffic according to above statement on NACL rule. So failure would not have occurred due to SSH DENY rule.
For Option B, Rule # * is only evaluated when the request doesn't match any of the rules listed. In our case, this rule is never evaluated because rule # 100 is ALLOW ALL traffic. So this option is incorrect.
For Option C, Network ACL is stateless, inbound rule # 100 allows incoming SSH request on port 22. When the request is outbound from the EC2 instance, NACL outbound rule # 200 allows the request to go out to ephemeral ports where the request is originated from.
For more information on ephemeral ports, refer documentation here:
https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-network-acls.html#nacl-ephemeral-ports (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-network-acls.html#nacl-ephemeral-ports)
For Option D, Network ACL is stateless, the request will not fail due to outbound rule # 100 SSH DENY because the outbound for the incoming SSH request should be open to ephemeral ports.
Refer documentation above for more on ephemeral ports.

First, You need to understand what are ephemeral ports, When a client initiates a request it chooses a random source port (ephemeral port) from a predefined range and "expects a response only at that port".
Proper definition:
An **ephemeral port** is a short-lived endpoint that is created by the operating system when a program requests any available user **port**. The operating system selects the **port** number from a predefined range, typically between 1024 and 65535, and releases the **port** after the related TCP connection terminates.

Now coming to the question ssh request originated from source would be from the ephemeral port say 48000. All traffic is allowed on all ports in the first inbound rule so request would proceed. We have denied for ssh in the first rule in outbound but for port 22(we need a rule for our ephemeral port)

therefore it does not match, we go to next rule where all traffic is allowed on all ports that's why response would succeed. if the port range would have been ephemeral in the first rule the request would have been denied.

Ask our Experts

QUESTION 7 CORRECT

Your organization was looking to download patches onto an existing EC2 instance which is inside a private subnet in existing custom VPC. You created a NAT Gateway and added a route to route table. However, when you are trying to download patches from internet onto EC2 instance, connection getting timed out. What could be the reason? (choose 2 options)

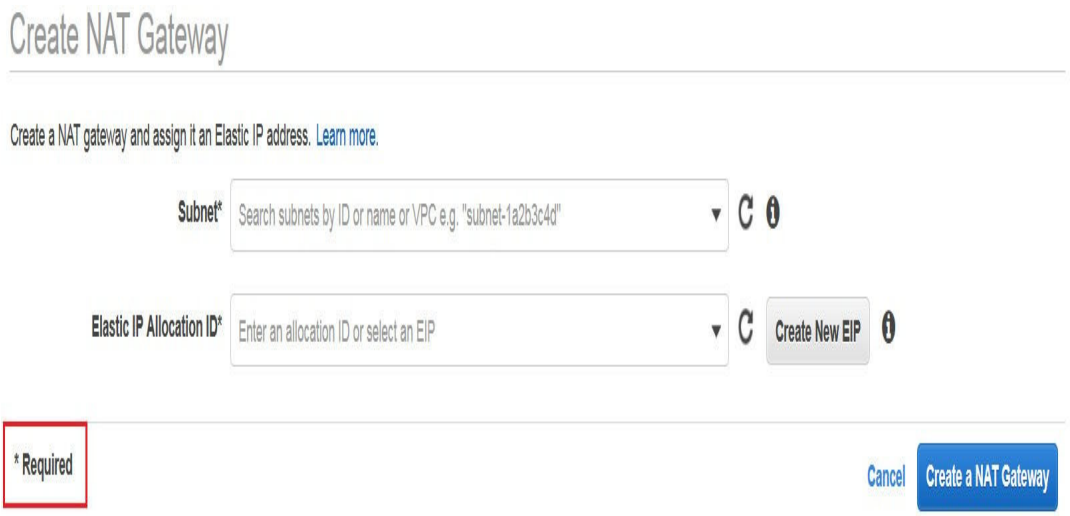
- [x] **A.** NAT Gateway created in private subnet without an Internet Gateway. ✔
- [ ] **B.** NAT Gateway is created without an Elastic IP Address.
- [x] **C.** Security Group's outbound rules of EC2 instance is restricted not to allow internet traffic. ✔
- [ ] **D.** NAT Gateway's Security Group inbound rules does not allow traffic from EC2 instance.

Explanation :

Answer A and C
For Option A, when creating NAT Gateway, there is an option to select subnet in which NAT Gateway will be created. This must be a public subnet which has a route to internet through Internet Gateway. If a private subnet is selected when creating NAT Gateway, it cannot route traffic to internet and hence the requests would fail.

- https://aws.amazon.com/premiumsupport/knowledge-center/nat-gateway-vpc-private-subnet/ (https://aws.amazon.com/premiumsupport/knowledge-center/nat-gateway-vpc-private-subnet/)

For Option B, NAT Gateway cannot be created without an elastic IP address. During the creation of NAT Gateway, Elastic IP Allocation ID is a mandatory field without which we cannot proceed to create NAT Gateway.So this option is incorrect.



For Option C, if the Security Group outbound rules does not allow internet traffic, EC2 instance cannot download patches from internet. This could be a possible reason.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSecurityGroups (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSecurityGroups)

For Option D, NAT Gateways does not have security groups.

- You cannot associate a security group with a NAT gateway. You can use security groups for your instances in the private subnets to control the traffic to and from those instances.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-basics (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-basics)

Security groups are stateful(not stateless as you mentioned below. It might be typo error). i.e if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inboundsecurity group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
AWS says that "Launch an instance in your private subnet. In the launch wizard, ensure that you select an Amazon Linux AMI. Do not assign a public IP address to your instance. Ensure that your security group rules allow inbound SSH traffic from the private IP address of your instance that you launched in the public subnet, and all outbound ICMP traffic"

Please check the below link to know more about it.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html)

Note:

Your security groups use connection tracking to track information about traffic to and from the instance. Rules are applied based on the connection state of the traffic to determine if the traffic is allowed or denied. **This allows security groups to be stateful — responses to inbound traffic are allowed to flow out of the instance regardless of outbound security group rules, and vice versa.**

As per AWS documentation,
Instances in Private Subnet Cannot Access internet

- Check that the NAT gateway is in the ***Available*** state. In the Amazon VPC console, go to the **NAT Gateways** page and view the status information in the details pane. If the NAT gateway is in a failed state, there may have been an error when it was created.
- Check that you've configured your route tables correctly:
  - The NAT gateway must be in a public subnet with a route table that routes internet traffic to an internet gateway.
  - Your instance must be in a private subnet with a route table that routes internet traffic to the NAT gateway.
  - Check that there are no other route table entries that route all or part of the internet traffic to another device instead of the NAT gateway.
- **Ensure that your security group rules for your private instance allow outbound internet traffic.**

Note
The NAT gateway itself allows all outbound traffic and traffic received in response to an outbound request (it is therefore stateful).

Please refer the following links for more information.

- https://aws.amazon.com/premiumsupport/knowledge-center/nat-gateway-vpc-private-subnet/ (https://aws.amazon.com/premiumsupport/knowledge-center/nat-gateway-vpc-private-subnet/)
- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html)

Ask our Experts

QUESTION 8 CORRECT

Your organization is planning to develop a web application containing a Web Server and an RDS Instance. This application will be accessed from internet. Your organization asked you to architect the solution on AWS. Your existing AWS

environment already has a VPC with a private subnet and public subnet which has a route to internet through Internet Gateway. What would be the best and cost effective solution you would provide?

- **A.** A bastion host in public subnet, Web Server EC2 in private subnet, RDS instance in private subnet.
- **B.** A bastion host in public subnet, Web Server EC2 in public subnet with Elastic IP, RDS instance in private subnet.
- **C.** A Bastion host in public subnet, Web Server EC2 in private subnet with NAT Gateway, RDS instance in private subnet.
- **D.** Web Server EC2 in public Subnet with Elastic IP, RDS instance in private subnet. ✔

Explanation :

Answer: D

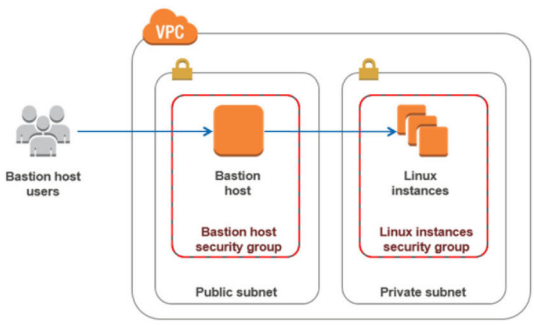
For option A, EC2 instance in private subnet cannot be reached from internet. A bastion host is a server whose purpose is to provide access to a private network from an external network, such as the Internet. It does not act as a proxy to route traffic from internet to private EC2 instance.

AWS Document says:
The solution architecture
In this section, I present the architecture of this solution and explain how you can configure the bastion host to record SSH sessions. Later in this post, I provide instructions about how to implement and test the solution.
Amazon VPC enables you to launch AWS resources on a virtual private network that you have defined. The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of your Amazon VPC. Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host. Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.

The solution architecture

In this section, I present the architecture of this solution and explain how you can configure the bastion host to record SSH sessions. Later in this post, I provide instructions about how to implement and test the solution.

Amazon VPC enables you to launch AWS resources on a virtual private network that you have defined. The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of your Amazon VPC. Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host. Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.



Option B, with EC2 instance in public subnet and Elastic IP attached, traffic from internet can reach Web Server and application works well. Although this option looks correct, this is not cost effective since there is no use of Bastion host anywhere since the EC2 instance is already in public subnet.

Option C, Same as option A. Although we have NAT Gateway attached to the subnet where Web Server EC2 resides, still the traffic from internet cannot reach the EC2 and NAT Gateway only routes traffic from AWS resources within a VPC to internet. Any traffic from internet into VPC resources is not allowed by NAT Gateway.

NAT Gateways

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances. For more information about NAT, see NAT.

Option D, The Web Server EC2 instance is in public subnet with elastic IP address attached to it and RDS in private subnet which cannot be reached from internet but only can allow traffic from EC2 in public subnet via security groups. From given answers, this looks correct in terms of cost and effectiveness.

- For more information on Elastic IP address, please refer documentation.

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html (https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html)

Ask our Experts

QUESTION 9 CORRECT

You are building a fleet of EC2 Linux Instances in the AWS environment for managing heavy workloads and writing data into AWS Redshift. The developers and administrators need to login to these EC2 machines to develop, fix, deploy, and manage workloads within your organizational network. Which of the following will be the secure and cost-effective architecture for this?

- **A.** EC2 instances on public subnet with secure SSH keys to login, RedShift in private subnet.
- **B.** A bastion host in public subnet with secure SSH key to login, EC2 instances in private subnet with secure SSH keys to login, RedShift in private subnet.
- **C.** AWS VPN connection from your organization to AWS VPC, a bastion host in VPN enabled subnet with secure SSH key to login, EC2 instances in private subnet with secure SSH keys to login, Redshift in private subnet. ✔
- **D.** AWS VPN connection from your organization to AWS VPC, EC2 instances in VPN enabled subnet with secure SSH keys to login, Redshift in private subnet.

Explanation :

Answer: C

For Option A, this is not secure because EC2 instances are in public subnet and are open to attacks such as DDoS. If you do not have a requirement to be accessed from internet, as a security best practice, try not to put AWS resources in public subnet.
For more information on DDoS attacks, refer documentation here

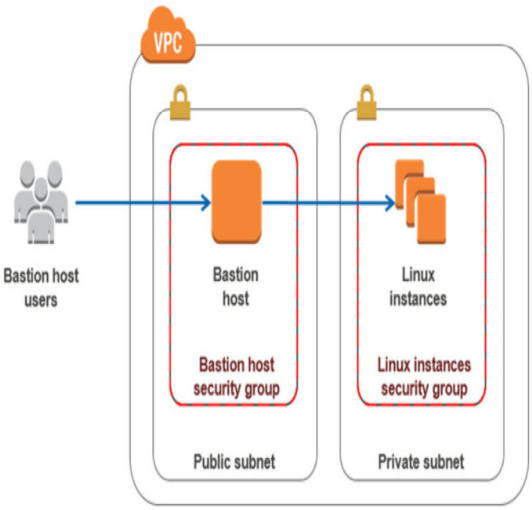
- https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/ (https://aws.amazon.com/answers/networking/aws-ddos-attack-mitigation/)

For Option B, Although EC2 instances are secured by putting them on private subnet and only enabling bastion host on public subnet looks correct, the requirement states, these instances should only be accessed via their organization network. So this option is incorrect.
A bastion host is a server whose purpose is to provide access to a private network from an external network, such as the Internet. It does not act as a proxy to route traffic from internet to private EC2 instance.
AWS Document says:
The solution architecture
In this section, I present the architecture of this solution and explain how you can configure the bastion host to record SSH sessions. Later in this post, I provide instructions about how to implement and test the solution.
Amazon VPC enables you to launch AWS resources on a virtual private network that you have defined. The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of your Amazon VPC. Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host. Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.

The solution architecture

In this section, I present the architecture of this solution and explain how you can configure the bastion host to record SSH sessions. Later in this post, I provide instructions about how to implement and test the solution.

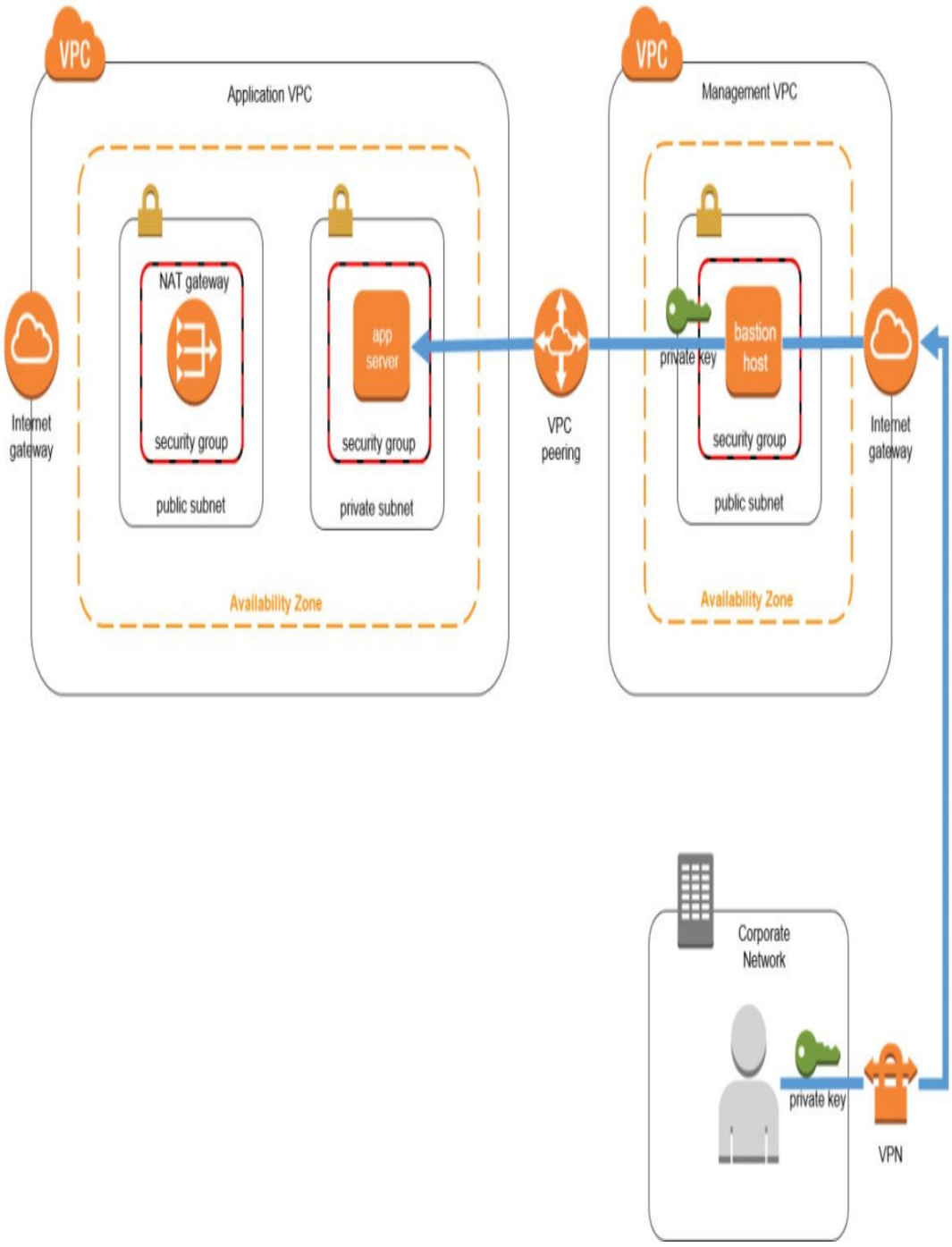
Amazon VPC enables you to launch AWS resources on a virtual private network that you have defined. The bastion host runs on an Amazon EC2 instance that is typically in a public subnet of your Amazon VPC. Linux instances are in a subnet that is not publicly accessible, and they are set up with a security group that allows SSH access from the security group attached to the underlying EC2 instance running the bastion host. Bastion host users connect to the bastion host to connect to the Linux instances, as illustrated in the following diagram.



For Option C, VPN connections are used to connect AWS VPC from your organization's network. By default, instances that you launch into an Amazon VPC can't communicate with your own (remote) network. You can enable access to your remote network from your VPC by attaching a virtual private gateway to the VPC, creating a custom route table, updating your security group rules, and creating an AWS managed VPN connection.
For more information on VPN, refer documentation here.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_VPN.html)

So, in this option, even from a VPN connection, only bastion host is exposed from AWS to VPN and you only open one connection from your organization to AWS. From bastion host, you can open connections to other resources in private subnet or other resources in peering VPCs.



- https://aws.amazon.com/blogs/mt/replacing-a-bastion-host-with-amazon-ec2-systems-manager/ (https://aws.amazon.com/blogs/mt/replacing-a-bastion-host-with-amazon-ec2-systems-manager/)

For Option D, although the connection is going through VPN and option looks correct, you need to open multiple connections to enable access to all EC2 instances. So when compared between option C and D, option C is a best practice and correct answer.

Note:

In the question, they mentioned that "Developers and Administrators need the login to the EC2 instances **Only within your organization network.**" So, they should access via their organization network.

Establish a VPN connection between your Organization network and your AWS.

Ask our Experts

QUESTION 10 CORRECT

You have a bastion host EC2 instance on AWS VPC public subnet. You would want to SSH to Bastion host EC2 instance. What would be the secure and minimal configuration you need inorder for SSH request to work? Assume route table is already setup with Internet Gateway.

- **A.** Allow SSH protocol(port 22) on Security Group Inbound and Security Group Outbound Allow Network ACL inbound and Network ACL outbound for IP range 0.0.0.0/0
- **B.** Allow SSH protocol(port 22) on Security Group Inbound and Security Group Outbound. Allow Network ACL inbound for your IP address.
- **C.** Allow SSH protocol(port 22) on Security Group Inbound and Network. ACL inbound for your IP address
- **D.** Allow SSH protocol(port 22) on Security Group Inbound. Allow Network ACL inbound and Network ACL outbound for your IP address. ✔

Explanation :

Answer D
Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules.

Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.

Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Security.html#VPC_Security_Comparison (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Security.html#VPC_Security_Comparison)

In option A, Security Group outbound is not necessary for SSH connection to work. Also, opening to 0.0.0.0/0 is insecure as it allows ALL on SSH. Although this option works, this is not secure and not a minimal configuration.
In options B and C, Network ACL outbound is not open. According to Network ACL stateless definition, this option would fail.
In option D, this is minimal and secure configuration to open only to your IP address. This is correct answer.

Ask our Experts

QUESTION 11 CORRECT

You have following Network ACL and Security Group rules. What would happen to an SSH request sent from 10.10.1.148 IP address to an EC2 instance with below security group and exists inside a subnet with below NACL rules?

Network ACL Inbound

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	SSH (22)	TCP (6)	22	10.10.1.0/24	ALLOW
200	SSH (22)	TCP (6)	22	10.10.1.148/32	DENY
300	ALL Traffic	ALL	ALL	10.10.1.148/32	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Network ACL Outbound

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Security Group Inbound

Type	Protocol	Port Range	Source	Description
SSH (22)	TCP (6)	22	10.10.1.0/24	

Security Group Outbound

Type	Protocol	Port Range	Destination	Description
SSH (22)	TCP (6)	22	172.32.1.0/24	

- **A.** SSH request succeeds due to rule # 100 in Network ACL inbound and outbound, Security Group inbound rule. ✔
- **B.** SSH request succeeds due to rule # 300 in Network ACL inbound and rule # 100 in Network ACL outbound, Security Group inbound rule.
- **C.** SSH request fails due rule # 200 in Network ACL inbound rule.
- **D.** SSH request fails due to Security Group outbound rule does not allow 10.10.1.148 IP address.

Explanation :

Answer: A

Security groups are stateful — if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules. Responses to allowed inbound traffic are allowed to flow out, regardless of outbound rules.
Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Security.html#VPC_Security_Comparison (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Security.html#VPC_Security_Comparison)

A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules)

For Option A, rule # 100 allows all traffic. So this will allow SSH request irrespective of other higher numbered rules. Security group rule allows SSH traffic for IP Range 10.10.1.0/24. IP address 10.10.1.148 falls under this IP range, so it allows SSH request. Network ACL outbound rule # 100 allows ALL traffic. So the request would succeed. This option is correct.
For Option B, when SSH request is made, rule # 300 is never evaluated because the request succeeds during rule # 300 evaluation. However, Rule # 300 gets evaluated when a non-SSH request is made. But, for this question, it is incorrect answer.
For Option C, rule # 200 is never evaluated because the request succeeds during rule # 100 evaluation. So this option is incorrect.
For option D, Security Groups are stateful. So, for an SSH request inbound to EC2 instance, security group outbound does not have an impact. So this option is incorrect.

Ask our Experts

QUESTION 12 CORRECT

Following are network ACL rules for a subnet. Which of the following statements are correct when request originating from 10.10.1.148 IP address?

Inbound rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
200	SSH (22)	TCP (6)	22	10.10.1.148/32	DENY
300	ALL Traffic	ALL	ALL	10.10.1.0/24	ALLOW
400	SSH (22)	TCP (6)	22	10.10.0.0/16	DENY
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Outbound Rules

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
200	HTTPS (443)	TCP (6)	443	10.10.1.0/24	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

- A. HTTPS(443) request would succeed.
- B. SSH(22) request would succeed.
- C. HTTP(80) request would succeed.
- D. All requests would fail. ✔

Explanation :

Answer - D
Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic (and vice versa).
A network ACL contains a numbered list of rules that we evaluate in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out of any subnet associated with the network ACL.
Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#ACLRules)

The client that initiates the request chooses the ephemeral port range. The range varies depending on the client's operating system. Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000. Requests originating from Elastic Load Balancing use ports 1024- 65535. Windows operating systems through Windows Server 2003 use ports 1025-5000. Windows Server 2008 and later versions use ports 49152-65535. A NAT gateway uses ports 1024-65535. For example, if a request comes into a web server in your VPC from a Windows XP client on the Internet, your network ACL must have an outbound rule to enable traffic destined for ports 1025-5000.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-network-acls.html#nacl-ephemeral-ports (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-network-acls.html#nacl-ephemeral-ports)

For the given rules in the question, the outbound rule * denies all the traffic outgoing except for 22 and 443 which are not part of ephemeral ports.
So all the requests would fail.
For more inför nation, please check below AWS Docs:

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#VPC_ACLs_Ephemeral_Ports (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#VPC_ACLs_Ephemeral_Ports)
- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#custom-network-acl (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#custom-network-acl)

Note:
For the HTTPS to succeed we would have to set Outbound Port Range to 1024--65535 and NOT to 443.
This is because the client that initiates the request chooses the ephemeral port range. The range varies depending on the client's operating system. Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000. So in order to allow outbound IPv4 responses to clients , for example serving web pages to people visiting the web servers in the subnet) you need to allow traffic through the ephemeral port ranges depending on the clients.

Many Linux kernels (including the Amazon Linux kernel) use ports 32768-61000. Requests originating from Elastic Load Balancing use ports 1024-65535.

Windows operating systems through Windows Server 2003 use ports 1025-5000.
Windows Server 2008 and later versions use ports 49152-65535

For example, if a request comes into a web server in your VPC from a Windows XP client on the Internet, your network ACL must have an outbound rule to enable traffic destined for ports 1025-5000.

Definition:
An **ephemeral port** is a short-lived endpoint that is created by the operating system when a program requests any available user **port**. The operating system selects the **port** number from a predefined range, typically between 1024 and 65535, and releases the **port** after the related TCP connection

terminates

Now in the question, let's say traffic from ip 10.10.1.148 originated from source port 48000(ephemeral port) to destination port 443 as inbound 443 is allowed so request would pass through NACL. Now for response to go through NACL https traffic must be allowed for port 48000(from which the request was originated) which is not allowed so the rule 3 all traffic deny applies in the outbound rule and the request fails.

Ask our Experts

QUESTION 13 CORRECT

Your organization network is connected to AWS VPC through VPN. VPC contains S3 VPC Gateway Endpoint created to access S3 through AWS internal network. You have data residing on your organization network. As an architect, you were asked to transfer the data to S3 without going to internet due to security compliance. What is the best possible way to achieve this?

- **A.** Setup an S3 proxy on EC2 instance within VPC and transfer data through VPN and S3 proxy to S3 ✔
- **B.** VPC Gateway endpoint can be used within remote(organization) network and data can be privately sent to S3 from remote(organization) network.
- **C.** Create VPN Gateway Endpoint to support this use case.
- **D.** Add a new route in VPC's VPN enabled route table with VPC endpoint to support direct transfer from remote(organization) network to S3.

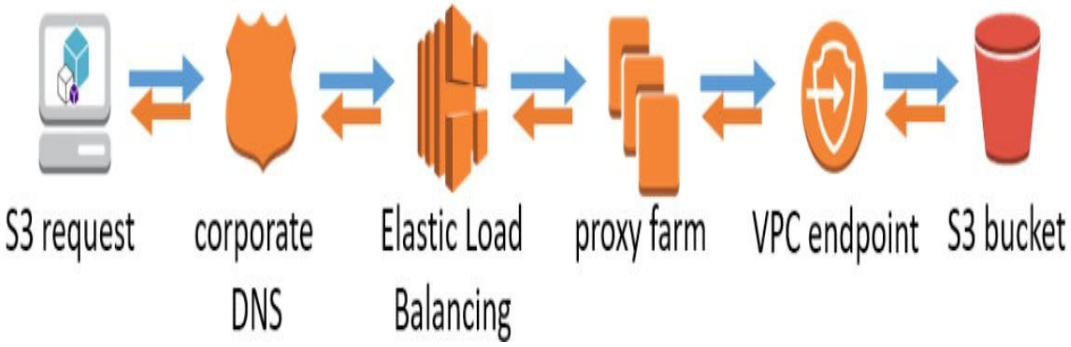
Explanation :

Answer: A

VPC Gateway endpoints are not supported outside VPC.
Endpoint connections cannot be extended out of a VPC. Resources on the other side of a VPN connection, VPC peering connection, AWS Direct Connect connection, or ClassicLink connection in your VPC cannot use the endpoint to communicate with resources in the endpoint service.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpce-gateway.html#vpc-endpoints-limitations
(https://docs.aws.amazon.com/apigateway/latest/developerguide/integrating-api-with-aws-services-s3.html)

So, to support such use cases, we can setup an S3 proxy server on AWS EC2 instance as shown below.



For more information on setting up S3 proxy, refer documentation here.

- https://aws.amazon.com/answers/networking/controlling-vpc-egress-traffic/ (https://aws.amazon.com/answers/networking/controlling-vpc-egress-traffic/)
- https://docs.aws.amazon.com/cli/latest/userguide/cli-http-proxy.html (https://docs.aws.amazon.com/cli/latest/userguide/cli-http-proxy.html)

From above explanation, only option A is correct.
Option B is not correct because VPC Gateway endpoints are not supported outside VPC. Option C, there is no AWS service called as VPN Gateway endpoint.
Option D, is not correct because VPC Gateway endpoints are not supported outside VPC and a route for VPC gateway endpoint cannot be added through route table. It can only be added by editing VPC gateway endpoint.

Ask our Experts

QUESTION 14 CORRECT

You have an existing VPC in us-east-1. You have created a VPC Endpoint for S3 and added it to the main route table. You have launched an EC2 instance inside a subnet that is associated with the main route table. From the new EC2 instance, when making a request to the S3 bucket within us-east-1, you noticed that the connection is failing. What could be the reason. (Choose 2 options)

- [x] **A.** EC2 instance security group outbound rules are restricted and does not contain prefix list. ✔
- [ ] **B.** Main route table does not have internet gateway association.
- [x] **C.** Subnet's Network ACL inbound rule does not allow traffic from S3. ✔
- [ ] **D.** Main route table does not have NAT gateway association.

Explanation :

Answer : A and C
For option A, By default, Amazon VPC security groups allow all outbound traffic, unless you've specifically restricted outbound access.
For a gateway endpoint, if your security group's outbound rules are restricted, you must add a rule that allows outbound traffic from your VPC to the service that's specified in your endpoint. To do this, you can use the service's prefix list ID as the destination in the outbound rule.

- https://aws.amazon.com/premiumsupport/knowledge-center/connect-s3-vpc-endpoint/ (https://aws.amazon.com/premiumsupport/knowledge-center/connect-s3-vpc-endpoint/)

So this option is correct.
For option B, when using VPC endpoint for S3, internet gateway is not required to route traffic to S3. VPC endpoint routes traffic internally within AWS without going out to internet.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-endpoints.html)

So this option is incorrect.

For option C, By default, network ACLs deny all inbound and outbound IPv4 traffic and, if applicable, IPv6 traffic. If your network ACL rules restrict traffic, you must specify the CIDR block (IP address range) for Amazon S3.

So this option is correct.
For option D, when using VPC endpoint for S3, NAT gateway is not required to route traffic to S3. VPC endpoint routes traffic internally within AWS without going out to internet.

So this option is incorrect.

Ask our Experts

QUESTION 15 CORRECT

Your organization had asked to be cost-efficient in designing AWS solutions. You have created three VPCs(VPA A, VPC B, VPC C), peered VPC A to VPC B and VPC B to VPC C. You have created a NAT gateway in VPC B and would like to use same NAT Gateway for resources within VPC A and VPC C. However, the resources within VPC A and VPC C cannot communicate to internet through NAT Gateway, but resources in VPC B can communicate. What could be the reason?

- **A.** Route tables in VPC A and VPC C are not configured to have VPC B's NAT gateway.
- **B.** Using another VPC's NAT Gateway is not supported in AWS. ✔
- **C.** VPC B's subnet which contains NAT gateway is not configured in VPC A and VPC C route tables.
- **D.** NAT Gateway is not created inside VPC B's public subnet.

Explanation :

Answer: B
In a VPC peering connection, using NAT Gateway of another VPC becomes transitive routing and is not supported in AWS.

Using a NAT Gateway with VPC Endpoints, VPN, AWS Direct Connect, or VPC Peering

A NAT gateway cannot send traffic over VPC endpoints, VPN connections, AWS Direct Connect, or VPC peering connections. If your instances in the private subnet must access resources over a VPC endpoint, a VPN connection, or AWS Direct Connect, use the private subnet's route table to route the traffic directly to these devices.

For example, your private subnet's route table has the following routes: internet-bound traffic (0.0.0.0/0) is routed to a NAT gateway, Amazon S3 traffic (pl-xxxxxxxx; a specific IP address range for Amazon S3) is routed to a VPC endpoint, and 10.25.0.0/16 traffic is routed to a VPC peering connection. The pl-xxxxxxxx and 10.25.0.0/16 IP address ranges are more specific than 0.0.0.0/0; when your instances send traffic to Amazon S3 or the peered VPC, the traffic is sent to the VPC endpoint or the VPC peering connection. When your instances send traffic to the internet (other than the Amazon S3 IP addresses), the traffic is sent to the NAT gateway.

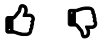
You cannot route traffic to a NAT gateway through a VPC peering connection, a VPN connection, or AWS Direct Connect. A NAT gateway cannot be used by resources on the other side of these connections.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-basics (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-nat-gateway.html#nat-gateway-basics)

For Option A, in VPC's route table, only NAT Gateway of the belonging VPC can be configured. VPC A and VPC C cannot configure VPC B's NAT Gateway in their respective route tables. This option is incorrect.

For Option B, as explained above, transitive routing is not supported. This option is correct.
For Option C, even if two VPCs are peered and configured route tables with their entire IP range, as explained above, transitive routing is not supported. This option is incorrect.
For Option D, the question says VPC B resources can communicate with internet for which NAT gateway should be on a public subnet. So this option is not valid.

Ask our Experts



QUESTION 16 CORRECT

You have setup a peering connection between two VPCs. You have launched EC2 instances in both VPCs and trying to communicate with each other through peering connections. However you found the request is getting timed out. From the following options, what could not be the reason for time out?

- **A.** Security groups of EC2 instances are not configured to allow traffic from peered VPC.
- **B.** Network ACLs have been configured not to allow traffic from peered VPC.
- **C.** Peered VPCs are in different regions. ✔
- **D.** Route tables of both VPCs only contains specific IP range for peering connection and either of the EC2 instances does not belong to the configured IP ranges.

Explanation :

Answer: C

For option A, when an EC2 instance is launched, a security group must be attached to it. By default, security group inbound does not contain any rules. We must add inbound rules to allow inbound requests to EC2 instance.

- When you create a security group, it has no inbound rules. Therefore, no inbound traffic originating from another host to your instance is allowed until you add inbound rules to the security group.
- By default, a security group includes an outbound rule that allows all outbound traffic. You can remove the rule and add outbound rules that allow specific outbound traffic only. If your security group has no outbound rules, no outbound traffic originating from your instance is allowed.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSecurityGroups (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_SecurityGroups.html#VPCSecurityGroups)

So this could be a reason for request getting timed out on either of the EC2 instances.

For option B, Network ACLs by default allows ALL traffic inbound and outbound. But, if the network ACL is modified to restrict traffic into the subnet, the EC2 instances launched inside the subnet would also the restricted according to Network ACL rules.

Default Network ACL

The default network ACL is configured to allow all traffic to flow in and out of the subnets to which it is associated. Each network ACL also includes a rule whose rule number is an asterisk. This rule ensures that if a packet doesn't match any of the other numbered rules, it's denied. You can't modify or remove this rule.

The following is an example default network ACL for a VPC that supports IPv4 only.

Inbound					
Rule #	**Type**	**Protocol**	**Port Range**	**Source**	**Allow/Deny**
100	All IPv4 traffic	All	All	0.0.0.0/0	ALLOW
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY
Outbound					
Rule #	**Type**	**Protocol**	**Port Range**	**Destination**	**Allow/Deny**
100	All IPv4 traffic	all	all	0.0.0.0/0	ALLOW
*	All IPv4 traffic	all	all	0.0.0.0/0	DENY

So this could be a reason for EC2 instances not able to communicate with each other. For Option C, AWS supports cross region VPC peering.
AWS Document says:
Inter-region VPC Peering
This approach leverages inter-region VPC peering connections to encrypt and route traffic between VPCs in different AWS Regions. A VPC peering connection uses the existing infrastructure of a VPC and private IP addresses; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.
Configuration Details
Inter-region VPC peering connections allow secure communication between VPC resources in different AWS Regions. All network traffic between regions is encrypted, stays on the AWS global network backbone, and never traverses the public internet, thereby reducing threat vectors, such as common exploits and DDoS attacks. VPC peering is appropriate for many scenarios, for example, to provide VPCs full access to each other's resources or to provide a set of VPCs partial access to resources in a central VPC. You can configure peering connections to provide access to part of a CIDR block or to an entire CIDR block of the peer VPC.
Considerations

Inter-region VPC peering is available in specific AWS Regions only (see the Amazon VPC Peering Guide for current availability). VPC peering does not support transitive routing, so if you require many-to-many connections, use a fully meshed configuration to allow communication between multiple VPCs. You can peer VPCs with overlapping CIDR blocks to the same VPC, such as a central VPC, but it will require specific adjustments to your subnet route tables (see Configurations with Specific Routes for guidance). Keep in mind that a peering relationship does not allow you to extend these other VPC connection types: VPN or AWS Direct Connect connections to a corporate network; internet connections through an internet gateway; a VPC endpoint to an AWS service; a ClassicLink connection (see Invalid VPC Peering Connection Configurations for detailed information)

Inter-region VPC Peering

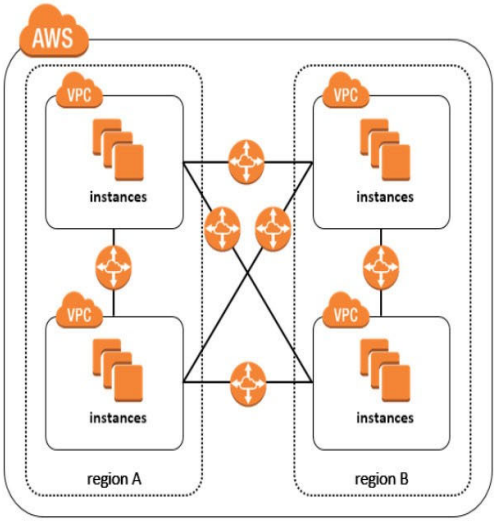
This approach leverages inter-region VPC peering connections to encrypt and route traffic between VPCs in different AWS Regions. A VPC peering connection uses the existing infrastructure of a VPC and private IP addresses; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware. There is no single point of failure for communication or a bandwidth bottleneck.

Configuration Details

Inter-region VPC peering connections allow secure communication between VPC resources in different AWS Regions. All network traffic between regions is encrypted, stays on the AWS global network backbone, and never traverses the public internet, thereby reducing threat vectors, such as common exploits and DDoS attacks. VPC peering is appropriate for many scenarios, for example, to provide VPCs full access to each other's resources or to provide a set of VPCs partial access to resources in a central VPC. You can configure peering connections to provide access to part of a CIDR block or to an entire CIDR block of the peer VPC.

Considerations

Inter-region VPC peering is available in specific AWS Regions only (see the Amazon VPC Peering Guide for current availability). VPC peering does not support transitive routing, so if you require many-to-many connections, use a fully meshed configuration to allow communication between multiple VPCs. You can peer VPCs with overlapping CIDR blocks to the same VPC, such as a central VPC, but it will require specific adjustments to your subnet route tables (see Configurations with Specific Routes for guidance). Keep in mind that a peering relationship does not allow you to extend these other VPC connection types: VPN or AWS Direct Connect connections to a corporate network; internet connections through an internet gateway; a VPC endpoint to an AWS service; a ClassicLink connection (see Invalid VPC Peering Connection Configurations for detailed information).



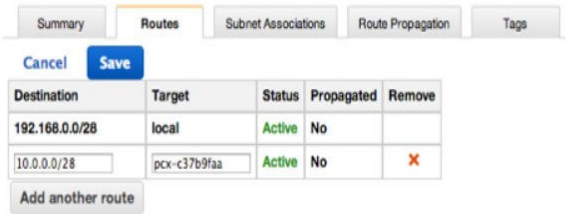
- https://aws.amazon.com/answers/networking/aws-multiple-region-multi-vpc-connectivity/ (https://aws.amazon.com/answers/networking/aws-multiple-region-multi-vpc-connectivity/)

So, this could not a reason for failure. So this option is correct.

For option D, when a VPC peering connection is created, we can configure route tables of both VPCs with entire CIDR ranges of peering VPCs or we can restrict the routing to only certain subnets or to a specific IP address.

5. For **Destination**, enter the IPv4 address range to which the network traffic in the VPC peering connection must be directed. You can specify the entire IPv4 CIDR block of the peer VPC, a specific range, or an individual IPv4 address, such as the IP address of the instance with which to communicate. For example, if the CIDR block of the peer VPC is `10.0.0.0/16`, you can specify a portion `10.0.0.0/28`, or a specific IP address `10.0.0.7/32`.

6. Select the VPC peering connection from **Target**, and then choose **Save**.



- https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/vpc-peering-routing.html (https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/vpc-peering-routing.html)

In this case, EC2 instances might have been in a subnet which is not having a peering connection route in its associated route table. So the connection will fail.
Note: The question is looking for "what could **not** the reason for time out?"

Ask our Experts

QUESTION 17 CORRECT

You created a new VPC with CIDR range 10.10.0.0/16 and a new subnet with CIDR range 10.10.1.0/24. CIDR with /24 comes with 256 IP addresses. When you go to VPC console subnets and look at the newly created subnet, you can only see 251 IP addresses. You have not launched any resources in the newly created VPC. What would have caused this?

- A. AWS reserves 5 IP addresses for every subnet. ✔
- B. AWS reserves 5 IP addresses for every VPC and are reserved from first subnet you create.
- C. AWS launches monitoring resources on behalf of you in new VPC when first subnet is created which will reserve 5 IP addresses from first subnet.
- D. None of the above.

Explanation :

Answer: A

Amazon reserves the first four (4) IP addresses and the last one (1) IP address of **every subnet** for IP networking purposes.

The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use, and cannot be assigned to an instance. For example, in a subnet with CIDR block `10.0.0.0/24`, the following five IP addresses are reserved:

- `10.0.0.0`: Network address.
- `10.0.0.1`: Reserved by AWS for the VPC router.
- `10.0.0.2`: Reserved by AWS. The IP address of the DNS server is always the base of the VPC network range plus two; however, we also reserve the base of each subnet range plus two. For VPCs with multiple CIDR blocks, the IP address of the DNS server is located in the primary CIDR. For more information, see Amazon DNS Server.
- `10.0.0.3`: Reserved by AWS for future use.
- `10.0.0.255`: Network broadcast address. We do not support broadcast in a VPC, therefore we reserve this address.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#VPC_Sizing (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Subnets.html#VPC_Sizing)

From above statement, only Option A is correct.
For Option C, AWS never launches any billable resources without notifying the account owner or administrator on behalf of you.

Ask our Experts

QUESTION 18 INCORRECT

You have created a new VPC and a private subnet. You will also be setting up VPN connection with your organization to communicate with resources within the VPC. Your organization would need DNS names for some of on-premise applications to communicate with VPC resources. You have launched a new EC2 instance with Auto-assign public IP as enable. When the instance is ready to use, you found that Public DNS name is missing. What should be done to enable it?

- **A.** Enable DNS Hostnames for VPC ✔
- **B.** Enable DNS Resolution for VPC
- **C.** Set auto-assign public IP to Use Subnet Setting
- **D.** You cannot have private DNS names for custom VPCs. Setup EC2 instance in default VPC.

Explanation :

Answer: A

By default, custom VPCs does not have DNS Hostnames enabled. So when you launch an EC2 instance in custom VPC, you do not have a public DNS name. You should go to VPC actions € Edit DNS Hostnames and enable it to have DNS hostnames for the resources within VPC.

DNS Support in Your VPC

Your VPC has attributes that determine whether your instance receives public DNS hostnames, and whether DNS resolution through the Amazon DNS server is supported.

Attribute	Description
`enableDnsHostnames`	Indicates whether the instances launched in the VPC get public DNS hostnames. If this attribute is `true`, instances in the VPC get public DNS hostnames, but only if the `enableDnsSupport` attribute is also set to `true`.
`enableDnsSupport`	Indicates whether the DNS resolution is supported for the VPC. If this attribute is `false`, the Amazon-provided DNS server in the VPC that resolves public DNS hostnames to IP addresses is not enabled. If this attribute is `true`, queries to the Amazon provided DNS server at the 169.254.169.253 IP address, or the reserved IP address at the base of the VPC IPv4 network range plus two will succeed. For more information, see Amazon DNS Server.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html)

Correct option is A.

For option B, DNS resolution is to resolve the DNS hostnames through Amazon DNS Server. For more information on Amazon DNS Server, refer documentation here.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#AmazonDNS (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_DHCP_Options.html#AmazonDNS)

Option C, auto-assign public IP defines whether you can have a public IP address for the EC2 you are launching. If you launch EC2 in a private subnet, this setting is always disabled. If you launch EC2 in public subnet, you can choose to have public IP address or not.

Option D is incorrect. Custom VPC provides an option to enable/disable DNS Hostnames as described above.

Assigning a Public IPv4 Address During Instance Launch

You can control whether your instance in a default or nondefault subnet is assigned a public IPv4 address during launch.

> **Important**
>
> You can't manually disassociate the public IPv4 address from your instance after launch. Instead, it's automatically released in certain cases, after which you cannot reuse it. If you require a persistent public IP address that you can associate or disassociate at will, associate an Elastic IP address with the instance after launch instead. For more information, see Elastic IP Addresses.

To assign a public IPv4 address to an instance during launch

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose **Launch Instance**.
3. Choose an AMI and an instance type and choose **Next: Configure Instance Details**.
4. On the **Configure Instance Details** page, select a VPC from the **Network** list. The **Auto-assign Public IP** list is displayed. Select **Enable** or **Disable** to override the default setting for the subnet.

   > **Important**
   >
   > A public IPv4 address cannot be assigned if you specify more than one network interface. Additionally, you cannot override the subnet setting using the auto-assign public IPv4 feature if you specify an existing network interface for eth0.

5. Follow the remaining steps in the wizard to launch your instance.

-
  - https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-ip-addressing.html#vpc-working-with-ip-addresses (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-ip-addressing.html#vpc-working-with-ip-addresses)

Note:
As per AWS docs "When you launch an instance into a nondefault VPC, we provide the instance with a private DNS hostname and we might provide a public DNS hostname, depending on the DNS attributes (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html#vpc-dns-support) you specify for the VPC and if your instance has a public IPv4 address."

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html#vpc-dns-support (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/vpc-dns.html#vpc-dns-support)

Ask our Experts

QUESTION 19 CORRECT

You are taking over AWS platform in your organization. You were asked to build a new application which would require a fleet of 20 EC2 instances inside a private VPC which should communicate with each other and no traffic going into the EC2 instances from internet, but should be able to receive requests from all other EC2 instances inside the VPC. When you looked at existing VPC, it was created with 10.10.0.0/24 CIDR range which contains only 256 IP addresses. You noticed that all 256 IP addresses were being consumed by 8 subnets with /27 CIDR ranges. How would you architect this solution?

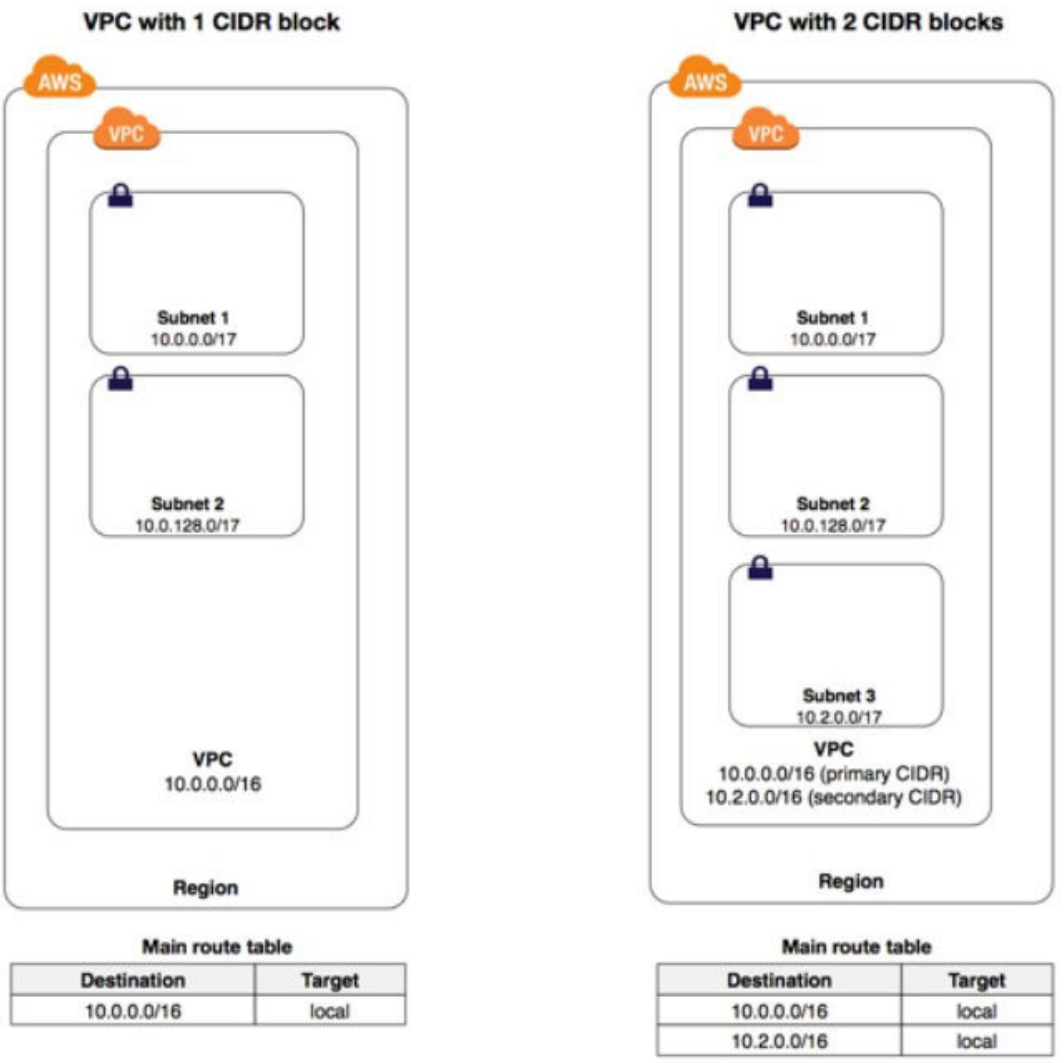
- **A.** Create a new VPC, setup 20 EC2 instances in new VPC and peer with existing VPC
- **B.** Add secondary CIDR range for the VPC, create new subnet and setup all instances in same subnet. ✔
- **C.** Edit subnet CIDR ranges to /28 and free up unused IP addresses.
- **D.** Launch EC2 instances in different subnets and setup Network ACLs and Security Groups to allow traffic between EC2 instances.

Explanation :

Answer: B

You can associate secondary IPv4 CIDR blocks with your VPC. When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is local).

In the following example, the VPC on the left has a single CIDR block (10.0.0.0/16) and two subnets. The VPC on the right represents the architecture of the same VPC after you've added a second CIDR block (10.2.0.0/16) and created a new subnet from the range of the second CIDR.



- https://aws.amazon.com/about-aws/whats-new/2017/08/amazon-virtual-private-cloud-vpc-now-allows-customers-to-expand-their-existing-vpcs/ (https://aws.amazon.com/about-aws/whats-new/2017/08/amazon-virtual-private-cloud-vpc-now-allows-customers-to-expand-their-existing-vpcs/)

For option A, although creating a new VPC, peering with existing VPC would work, it creates a lot of configuration. This solution is suited when you want to isolate certain resources within each VPC and communicate certain resources in both VPCs, or if the VPCs belong to different accounts, or if VPCs are in different region. There is a limit of 5 VPCs per region and creating VPCs without a definite need might hit the limit in long run.

For option B, adding a secondary CIDR to existing VPC is a simple configuration and can enable more IP addresses to current VPC.
For option C, a subnet's CIDR cannot be edited once created.
For option D, although this option works, this would create lot of complexity around setting up new Security Groups and netowork ACLs. This setup would be difficult to maintain and troubleshoot in case of any issues.
So, with given options, although there are multiple working solutions, option B is recommended solution.

Ask our Experts

QUESTION 20 CORRECT

Your organization has a VPC setup with custom route table having 40 routes for different use cases such as VPC peering, VPN connections, NAT gateways with different IP ranges. Main route table was having local route along with internet gateway to act for public subnet. Your VPC IP range is 10.10.0.0/16 and many teams working on this VPC to create subnets for their applications which needs to have custom route table associated with it. However, many a times, these teams forget to explicitly associate the custom route table to the subnets. This will implicitly associate with main route table which has internet gateway which is causing security concerns. This is also leading to lot troubleshooting hours when the connections to new subnet from VPN connection does not work as expected. As an architect, how would you resolve this issue?

- **A.** Create a script to create subnet and associate new subnet with custom route table. Share this with all the teams.
- **B.** Make custom route table as main route table. New subnets created will now implicitly associate with it. ✔
- **C.** Delete Internet Gateway route from main route table.
- **D.** Delete all routes from custom table and add to main route table. Delete all routes from main table and add to custom table.

Explanation :

Answer: B

A custom route table can be made as main route table so that all implicit associations of subnets will now point to newly set main route table. All the future implicitly associations of newly created subnets will point to newly set main route table.

Replacing the Main Route Table

You can change which route table is the main route table in your VPC.

To replace the main route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route Tables**.
3. Select the route table that should be the new main route table, and then choose **Set as Main Table**.
4. In the confirmation dialog box, choose **Yes, Set**.

The following procedure describes how to remove an explicit association between a subnet and the main route table. The result is an implicit association between the subnet and the main route table. The process is the same as disassociating any subnet from any route table.

For option A, although subnet creation and association can be done programatically, it may not be feasible to share access keys with all the teams (assuming the creation process is done on remote network where roles cannot be used). It is also a difficult task for organization to setup the process to run this script for new teams as they might not be aware of it. So, this option is not the best of the lot.
For option B, as described above, setting custom table as main route table is a simple configuration and all the associations would point to the new main route table implicitly.
For option C, deleting internet gateway does not solve the problem. It might create a new problem for EC2 instances using NAT gateway to cause failures in connecting to internet.
For option D, although this is an option, it is tidious and error prone.
So, with given options, although there are multiple working solutions, option B is recommended solution.

Ask our Experts

QUESTION 21 CORRECT

You are an architect in your organization. One of the application team in your organization comes to you stating recently they noticed the requests sending from an EC2 instance to an RDS in the same VPC but in another subnet are getting timed out. They claim that connections were working before. How do you troubleshoot this issue?

- **A.** Create VPC flow log for subnet where RDS instance is launched. ✔
- **B.** Check CloudWatch metrics for RDS instance.
- **C.** Check OS level logs inside RDS instance.
- **D.** Check OS level logs inside EC2 instance.

Explanation :

Answer: A

For option A, VPC Flow Logs captures IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon CloudWatch Logs. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.
You can create a flow log for a VPC, a subnet, or a network interface.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-logs-basics (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-logs-basics)

VPC Flow Logs capture following information and logs them to CloudWatch logs,
version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes start end action log-status
Find more information about each record here.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-log-records (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-log-records)

So, using VPC flow logs, we can identify if the traffic is being rejected by RDS instance when sent from the EC2 instance on a certain port. From there on, we can identify if there any overly restrictive Security Group rules or Network ACL rules.
For option B, CloudWatch metrics for RDS gives the details about RDS underlying database instance metrics. But this does not contain details about networking requests sent to RDS instance.
For more information on CloudWatch metrics for RDS, refer documentation here.

- https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/rds-metricscollected.html (https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/rds-%20metricscollected.html)

For option C, RDS underlying OS is managed by AWS and cannot be accessed by AWS customers.
For option D, enabling OS level logs at the EC2 instance where the request is being made does not provide any information on why the request is being timed out at RDS instance.
So, the correct answer is option A.

Ask our Experts

QUESTION 22 CORRECT

You have setup two VPCs VPC A – 10.10.0.0/16 and VPC B – 10.11.0.0/16. You have also setup VPC peering connection. Which of the following is correct route table configurations for the VPC peering to work?

- **A.** VPC B route table contains route with Destination as 10.10.0.0/16
- **B.** VPC A route table contains route with Destination as 10.11.0.0/16.
- **C.** VPC B route table contains route with Destination as 10.10.1.0/24 and VPC A route table contains route with Destination as 10.11.1.0/28. ✔
- **D.** VPC A route table contains route with Destination as 10.10.1.0/24 and VPC B route table contains route with Destination as 10.11.1.0/28.

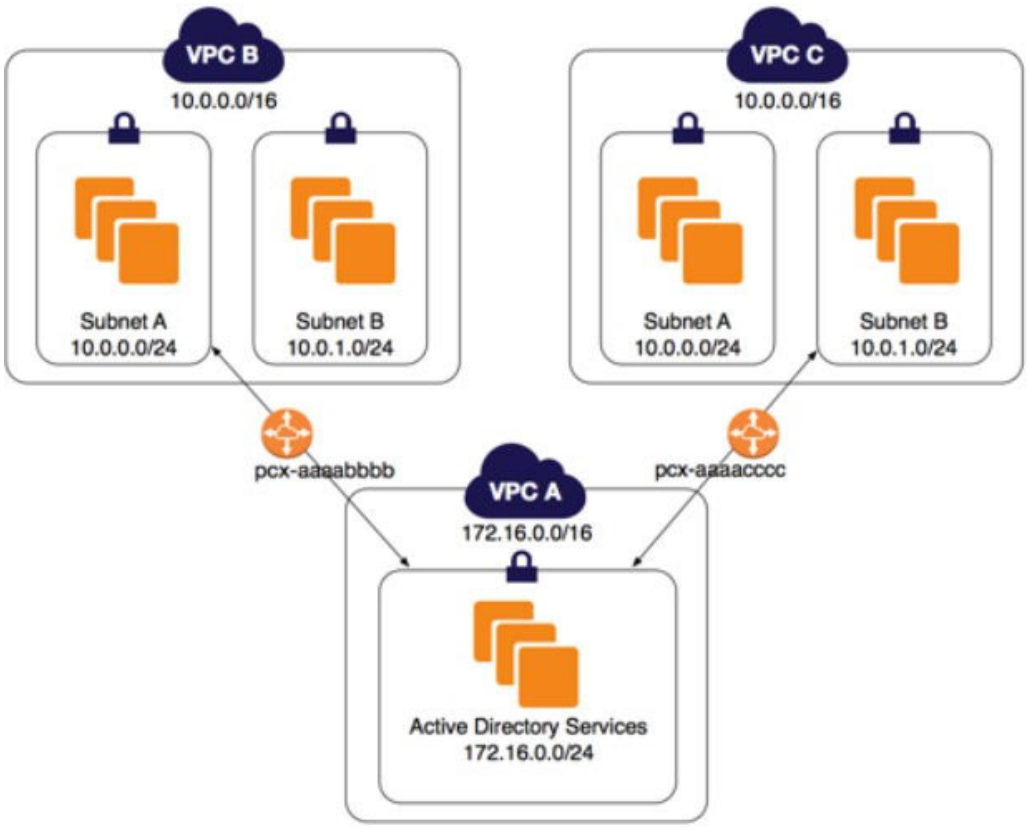
Explanation :

Answer: C

To send private IPv4 traffic from your instance to an instance in a peer VPC, you must add a route to the route table that's associated with your subnet in which your instance resides. The route points to the CIDR block (or a portion of the CIDR block) of the peer VPC in the VPC peering connection.

The owner of the other VPC in the peering connection must also add a route to their subnet's route table to direct traffic back to your VPC. For more information about supported route table configurations for VPC peering connections.

You can also peer a VPC with a specific subnet of another VPC instead of peering entire VPC.



- https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html (https://docs.aws.amazon.com/vpc/latest/peering/peering-configurations-partial-access.html)

For options A and B, they do not have second route added to return the connection back to requester VPC. So they are in correct.

For option C, as discussed above, we can configure subnets for a peering connection. So VPC A route table configured VPC B's subnet 10.11.1.0/28 and VPC B route table configured VPC A's subnet 10.10.1.0/24. This configuration is correct from given options.

For Option D, VPC A ad VPC B configured their own subnets in the respective route tables. So, this configuration will not work.

Ask our Experts

QUESTION 23 MARKED AS REVIEW CORRECT

Following are Security Group inbound rules. What is correct statement below?

Type	Protocol	Port Range	Source	Description
HTTP (80)	TCP (6)	80	10.10.1.148/32	
HTTP (80)	TCP (6)	80	10.10.1.0/28	
ALL TCP	TCP (6)	ALL	10.10.1.148/32	
SSH (22)	TCP (6)	22	10.10.1.0/28	
Custom UDP Rule	UDP (17)	3000	10.10.1.148/32	

- A. All rules are correct.
- B. HTTP port 80 for source 10.10.1.148/32 is duplicated. ✔
- C. SSH port 22 for source 10.10.1.0/28 is duplicated.
- D. Custom UDP rule port 3000 for source 10.10.1.148/32 is duplicated.

Explanation :

Answer: B

Lets take a look at the inbound rules.

- Rule # 3 defines ALL TCP allowed for 10.10.1.148 IP address
- Rule # 2 and # 4 defines port 80 and 22 are allowed for IP addresses 10.10.1.0- 10.10.1.16.
- Rule # 1 defines port 80 for 10.10.1.148 IP address.
- Rule # 5 defines custom UDP port 3000 for 10.10.1.148 IP address. Out of these rules only rule # 1 is duplicated with rule # 3.

Rest all rules are configured correctly without any duplicates. So option B is correct.

Ask our Experts

QUESTION 24 CORRECT

Following is a route table configuration inside a VPC which was initially created with 20.0.0.0/16. Which of the following statements is correct?

Destination	Target	Status	Propagated
30.0.0.0/20	local	Active	No
20.0.0.0/16	local	Active	No

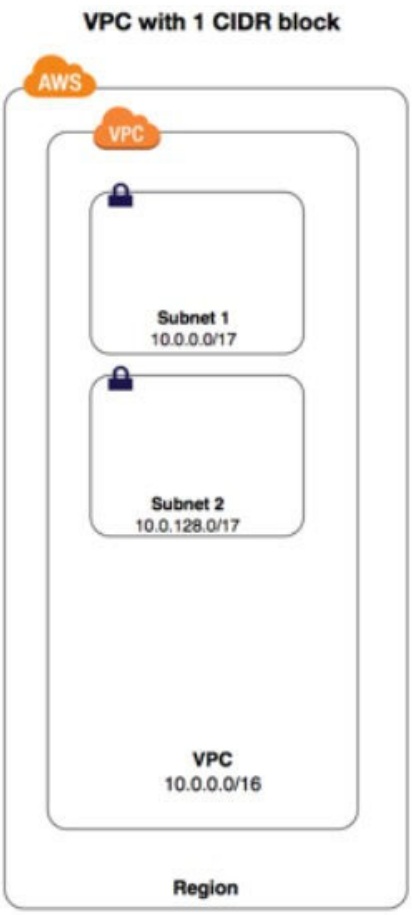
- A. VPC peering connection route for VPC with 30.0.0.0/20 IP range.
- B. VPN connection route for remote network with 30.0.0.0/20 IP range.
- C. Direct Connect connection route for remote network with 30.0.0.0/20 IP range.
- D. Secondary IP CIDR range 30.0.0.0/20 for VPC with local route. ✔

Explanation :

Answer: D

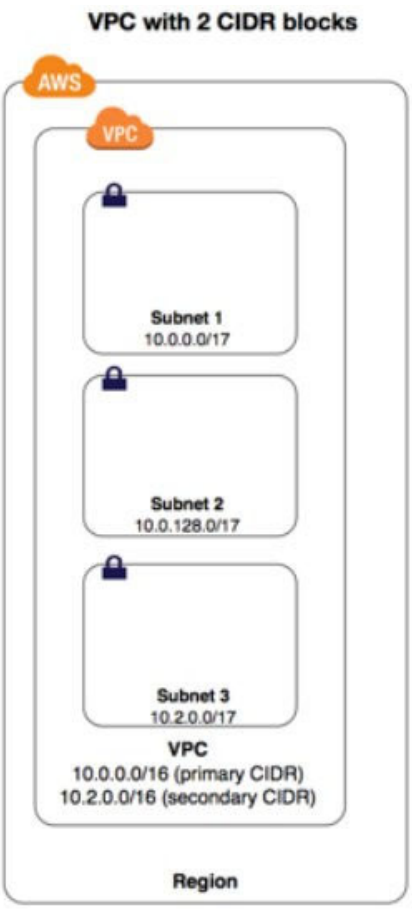
You can associate secondary IPv4 CIDR blocks with your VPC. When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is local).

In the following example, the VPC on the left has a single CIDR block (10.0.0.0/16) and two subnets. The VPC on the right represents the architecture of the same VPC after you've added a second CIDR block (10.2.0.0/16) and created a new subnet from the range of the second CIDR.



Main route table

Destination	Target
10.0.0.0/16	local



Main route table

Destination	Target
10.0.0.0/16	local
10.2.0.0/16	local

- https://aws.amazon.com/about-aws/whats-new/2017/08/amazon-virtual-private-cloud-vpc-now-allows-customers-to-expand-their-existing-vpcs/ (https://aws.amazon.com/about-aws/whats-new/2017/08/amazon-virtual-private-cloud-vpc-now-allows-customers-to-expand-their-existing-vpcs/)

From the above image, Main route table shows the routes for primary and secondary IP ranges. So correct option is D.
For option A, VPC peering connection route contains Target as pcx-xxxxxx. For option B, VPN connection route contains Target as vgw-xxxxxxx.
For option C, Direct Connect connection route too contains Target as vgw-xxxxxx.

Ask our Experts

QUESTION 25 INCORRECT

Your organization had setup a VPC with CIDR range 10.10.0.0/16. There are total 100 subnets within the VPC and are being actively used by multiple application teams. An application team who is using 50 EC2 instances in subnet 10.10.55.0/24 complains there are intermittent outgoing network connection failures for around 30 random EC2 instances in a given day. How would you troubleshoot issue with minimal configuration and minimal logs written?

- **A.** Create a flow log for the VPC and filter the logs in CloudWatch log group. ✖
- **B.** Create flow log for each EC2 instance network interface one by one and troubleshoot the connection issue.
- **C.** Create a flow log for subnet 10.10.55.0/24. ✔
- **D.** None of the above.

Explanation :

Answer: C

VPC Flow Logs captures IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon CloudWatch Logs. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

You can create a flow log for a VPC, a subnet, or a network interface.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-logs-basics (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-logs-basics)

VPC Flow Logs capture following information and logs them to CloudWatch logs,

version account-id interface-id srcaddr dstaddr srcport dstport protocol packets bytes start end action log-status

Find more information about each record here.

- https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-log-records (https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html#flow-log-records)

For option A, although creating a flow log for entire VPC would work, it captures lot of unrequired information from rest 99 subnets and finding out the affected EC2 instances from CloudWatch logs would become really troublesome.

For Option B, creating flow log at each EC2 network interface would work, but it takes log of configuration and time consuming trial and error troubleshooting.

For Option C, creating a flow log for the subnet would capture just the traffic going in and out of the subnet. This would help us identify the network trace for the affected EC2 instances and find out the root cause in timely manner.

Ask our Experts

Finish Review (https://www.whizlabs.com/learn/course/aws-csaa-practice-tests/quiz/14785)

Certification

- Cloud Certification (https://www.whizlabs.com/cloud-certification-training-courses/)
- Java Certification (https://www.whizlabs.com/oracle-java-certifications/)
- PM Certification (https://www.whizlabs.com/project-management-certifications/)
- Big Data Certification (https://www.whizlabs.com/big-data-certifications/)

Mobile App

- Android Coming Soon
- iOS Coming Soon

Company

- Support (https://help.whizlabs.com/hc/en-us)
- Discussions (http://ask.whizlabs.com/)
- Blog (https://www.whizlabs.com/blog/)

Follow us

(https://www.facebook.com/whizlabs.software/)

(https://in.linkedin.com/company/whizlabs-software)

(https://twitter.com/whizlabs?lang=en)

(https://plus.google.com/+WhizlabsSoftware)