AWS VPC Peering

Definition:

VPC Peering in AWS allows you to connect two Virtual Private Clouds (VPCs) to enable communication between them using private IP addresses, as if they were on the same network. VPC Peering is a one-to-one relationship between two VPCs.

Key Concepts:

1. **VPC (Virtual Private Cloud)**:
   A logically isolated network in AWS where users can define their IP ranges, create subnets, configure route tables, security groups, and network access control lists (NACLs).
2. **VPC Peering**:
   A networking connection between two VPCs that allows traffic to be routed between them using private IP addresses without the need for VPNs, gateways, or the internet.

Key Characteristics of VPC Peering:

- **Non-Transitive**:
  VPC Peering is non-transitive, meaning if VPC A is peered with VPC B and VPC B is peered with VPC C, VPC A cannot communicate directly with VPC C. You would need to create separate peering connections between A and C.
- **One-to-One Relationship**:
  A peering connection can only exist between two VPCs. To connect multiple VPCs, multiple peering connections need to be established.
- **Regional and Inter-Regional Peering**:
  VPCs can be peered within the same region (intra-region) or across different AWS regions (inter-region VPC peering).
- **No Overlapping IP Addresses**:
  The VPCs involved in peering must have unique, non-overlapping CIDR ranges.
- **Bidirectional Traffic**:
  After peering, traffic can flow in both directions, but routes need to be manually configured in each VPC’s route table to enable communication.

VPC Peering Use Cases:

1. **Cross-Account Peering**:
   VPC Peering allows VPCs from different AWS accounts to communicate. This is useful for companies collaborating across different AWS accounts or subsidiaries within an organization.
2. **Multi-Region Applications**:
   Inter-region VPC peering enables connecting VPCs across different AWS regions, allowing for disaster recovery, high availability, or global application access.
3. **Service Isolation**:
   Different services can be deployed in separate VPCs (e.g., one VPC for development, another for production) and connected via VPC Peering for secured, isolated communication.

How VPC Peering Works:

1. **Peering Request**:
   One VPC owner (requester) initiates a peering connection by sending a request to the owner of the other VPC (acceptor).
2. **Peering Acceptance**:
   The owner of the acceptor VPC must approve the peering request.
3. **Update Route Tables**:
   Both VPCs need to manually update their route tables to route traffic between each other over the peering connection.
4. **Security Groups and NACLs**:
   Appropriate inbound and outbound rules must be configured in both VPCs' security groups and NACLs to allow traffic over the peering connection.

Steps to Create a VPC Peering Connection:

1. **Initiate Peering Request**:
   - Go to the VPC Dashboard.
   - Navigate to "Peering Connections" and click "Create Peering Connection."
   - Specify the requester and acceptor VPCs (can be in the same or different AWS accounts).
2. **Accept Peering Request**:
   - Once the peering connection is initiated, the acceptor must go to their VPC Dashboard and accept the request.
3. **Modify Route Tables**:
   - Go to the Route Tables section.

    - Add a route to each VPC's route table, directing traffic to the other VPC's CIDR block through the peering connection.
4. **Security Groups and NACLs**:
    - Modify the security groups and NACLs to allow the desired traffic between the peered VPCs.

Cross-Account VPC Peering:

- **IAM Permissions**:
  The user initiating or accepting the peering request must have the necessary IAM permissions.
  Permissions include:
    - `CreateVpcPeeringConnection`
    - `AcceptVpcPeeringConnection`
    - `DescribeVpcPeeringConnections`
- **Account ID**:
  You'll need the AWS account ID of the other VPC owner to set up the connection.

Pricing and Performance:

- **Cost**:
  There is no hourly charge for VPC peering connections themselves. However, data transfer between VPCs is billed at the standard AWS data transfer rates.
    - Intra-region transfers are typically cheaper than inter-region transfers.
- **Bandwidth and Latency**:
  Since VPC Peering uses the AWS backbone network, the connection is fast and low-latency, with no need to route traffic through the public internet.

VPC Peering and Transit Gateway:

- **Transit Gateway**:
  AWS Transit Gateway is an alternative to VPC Peering, enabling multiple VPCs and on-premises networks to be connected via a central hub. Unlike VPC Peering, it allows transitive routing between VPCs.
- **Comparison with VPC Peering**:
    - **Scalability**: VPC Peering is point-to-point, whereas Transit Gateway supports connecting thousands of VPCs and on-premises systems.

    - **Transitive Routing**: Transit Gateway supports transitive routing, meaning VPCs connected to the Transit Gateway can communicate with each other without needing individual peering connections.

VPC Peering Limitations:

- **No Transitive Peering**:
  VPC A cannot communicate with VPC C through VPC B, even if both VPCs have a peering connection with VPC B.
- **No Overlapping CIDR Blocks**:
  VPCs involved in peering cannot have overlapping CIDR ranges. This ensures proper routing and prevents IP conflicts.
- **Routing Through NAT/IGW Not Supported**:
  VPC Peering cannot route traffic that's meant to traverse NAT gateways, VPNs, or Internet Gateways. Traffic remains strictly between private IP ranges.

VPC Peering Troubleshooting:

1. **Routing Issues**:
   Ensure both VPCs have correct routes in their route tables, allowing traffic to flow between the two VPCs over the peering connection.
2. **Security Group and NACL Configuration**:
   Verify that security groups and NACLs are set to allow inbound and outbound traffic for the necessary protocols and IP ranges.
3. **Check Overlapping CIDR Ranges**:
   Overlapping CIDR blocks between the VPCs will prevent a peering connection from being established.
4. **Peering Connection Status**:
   Check the status of the peering connection in the AWS Management Console. It should be in the "Active" state for traffic to flow.

Security Considerations:

- **Security Groups**:
  Set up fine-grained security group rules to control what traffic is allowed between peered VPCs. Only allow necessary protocols and ports to reduce exposure.
- **Network ACLs (NACLs)**:
  NACLs provide an additional layer of security. Configure them to allow or block traffic between the VPCs as needed.

- **Data Transfer Security**:
  Although VPC Peering leverages AWS's private infrastructure, consider encrypting sensitive data before transferring it across VPCs.

Best Practices:

1. **Least Privilege Access**:
   When configuring security groups and NACLs, only open necessary ports and protocols for communication between the VPCs.
2. **Monitor Traffic**:
   Use AWS CloudWatch and VPC Flow Logs to monitor the traffic flowing between the peered VPCs to ensure it adheres to your security policies.
3. **Use Tags for Organization**:
   Tag your VPC Peering connections for easier identification and organization, especially in large-scale environments.
4. **Plan CIDR Ranges Early**:
   Avoid potential IP conflicts and overlapping ranges by carefully planning the IP address space of your VPCs before establishing peering.