Overview of VPC (Virtual Private Cloud)

Importance for AWS Certification:

- Essential for AWS Certified Solutions Architect Associate and AWS Certified SysOps Administrator Associate exams.
- For AWS Certified Developer Associate level, a high-level understanding is sufficient, with potential for 1-3 exam questions.

Key Concepts of VPC:

1. **VPC (Virtual Private Cloud):**
    - A virtual network dedicated to your AWS account.
    - Isolated from other virtual networks in the AWS cloud.
    - Enables you to launch AWS resources into a virtual network that you've defined.
2. **Subnets:**
    - Subdivisions within a VPC.
    - Can be designated as public or private.
    - Public subnets have direct access to the internet, while private subnets do not.
3. **Internet Gateways:**
    - A horizontally scaled, redundant, and highly available VPC component.
    - Allows communication between instances in your VPC and the internet.
4. **NAT Gateways:**
    - Network Address Translation (NAT) service.
    - Enables instances in a private subnet to connect to the internet or other AWS services, but prevents the internet from initiating a connection with those instances.
5. **Security Groups:**
    - Virtual firewalls for your instances to control inbound and outbound traffic.
    - Operate at the instance level.
6. **Network ACLs (NACLs):**
    - Operate at the subnet level.
    - Stateless filters that control inbound and outbound traffic.
7. **VPC Flow Logs:**
    - Capture information about the IP traffic going to and from network interfaces in your VPC.
    - Useful for monitoring and troubleshooting connectivity issues.

8. **VPC Peering:**
    - Network connection between two VPCs that enables you to route traffic between them using private IP addresses.
    - VPCs can be in different AWS accounts or regions.
9. **VPC Endpoints:**
    - Enable private connections between your VPC and supported AWS services and VPC endpoint services powered by AWS PrivateLink.
    - Two types: Interface endpoints and Gateway endpoints.
10. **Site-to-Site VPN:**
    - Connection between your VPC and your remote network, such as your corporate data center.
    - Uses IPsec protocol to encrypt the data as it travels across the internet.
11. **Direct Connect:**
    - Dedicated network connection from your premises to AWS.
    - Provides a more consistent network experience compared to internet-based connections.

Exam Relevance:

- These concepts are vital for understanding and working with AWS networking.
- Focus on understanding the high-level purpose and functionality of each component.
- Expect to encounter questions related to VPC and its components in the AWS Certified Developer Associate exam, though not in extensive detail.

This overview should serve as a foundation for understanding VPCs in AWS. More detailed concepts and use cases will be revisited throughout the course as needed.

Introduction to VPC and Subnets

Virtual Private Cloud (VPC):

- **Definition**: A VPC is a private network within the AWS cloud, allowing you to deploy resources in a logically isolated environment.
- **Regional Resource**: A VPC is tied to a specific AWS region. Each region will have its own VPC.

Subnets:

- **Purpose**: Subnets partition your VPC into smaller network segments.
- **Level**: Defined at the availability zone (AZ) level.
- **Types**:

    - **Public Subnet**: Accessible from the internet. Instances in a public subnet can communicate with the internet.
    - **Private Subnet**: Not accessible from the internet. Instances in a private subnet cannot be directly reached from the internet for added security.

Route Tables:

- **Function**: Define how network traffic is directed within your VPC, specifying routes between subnets and other network destinations.
- **Application**: Route tables control the traffic between subnets and determine internet access for instances.

Instances:

- **Public Subnet Instances**: Have direct access to the internet.
- **Private Subnet Instances**: Do not have direct internet access, enhancing security.

VPC Components:

- **IP Range (CIDR)**: Specifies the range of IP addresses within your VPC.
- **AZ Configuration**: Each VPC spans multiple AZs, and you can create public and private subnets in each AZ.

Default VPC:

- **Configuration**: AWS provides a default VPC with a public subnet in each AZ.
- **Usage**: Useful for quickly deploying resources without extensive network configuration.

Internet Gateways and NAT Gateways

Internet Gateway:

- **Purpose**: Allows instances in public subnets to connect to the internet.
- **Location**: Resides in the VPC.
- **Routing**: Public subnets have routes to the internet gateway, enabling internet access for instances within these subnets.

NAT Gateways and NAT Instances:

- **Purpose**: Enable instances in private subnets to access the internet for tasks like software updates while keeping them inaccessible from the internet.
- **Types**:
    - **NAT Gateway**: Managed by AWS, providing scalability and maintenance-free operation.
    - **NAT Instance**: Managed by the user, requiring manual scaling and maintenance.
- **Configuration**:
    - **Placement**: Deployed in public subnets.
    - **Routing**: Private subnets have routes to the NAT gateway or instance, which in turn routes traffic to the internet gateway.

Typical AWS Infrastructure Setup

- **Public Subnet**: Contains instances that need internet access.
- **Private Subnet**: Contains instances that do not need to be accessible from the internet but may need outbound internet access for updates and other functions.
- **Internet Gateway**: Facilitates internet access for public subnet instances.
- **NAT Gateway/Instance**: Facilitates internet access for private subnet instances without exposing them to inbound internet traffic.

This overview of VPC and subnet concepts is fundamental for understanding AWS networking and effectively designing and managing resources within your AWS environment.

Network Security in VPC

Network ACL (NACL):

- **Definition**: A Network ACL (NACL) is a firewall that controls traffic to and from subnets in your VPC.
- **Rules**: NACLs can have both allow and deny rules.
- **Application**: Attached at the subnet level, NACLs filter traffic based on IP addresses.
- **Function**: Acts as the first line of defense for your subnet.
- **Traffic Flow**: Incoming and outgoing internet traffic passes through the NACL before reaching the subnet's resources.

Security Groups:

- **Definition**: Security Groups are firewalls that control traffic to and from EC2 instances or Elastic Network Interfaces (ENIs).
- **Rules**: Only allow rules can be specified in Security Groups.
- **Application**: Attached to individual EC2 instances or ENIs.
- **Function**: Provides a second layer of defense after NACLs.
- **Traffic Flow**: Filters traffic based on IP addresses or other security groups and allows return traffic automatically (stateful).

Differences Between Network ACLs and Security Groups

- **Attachment**:
  - NACLs: Attached at the subnet level.
  - Security Groups: Attached to individual instances or ENIs.
- **Rule Types**:
  - NACLs: Can specify both allow and deny rules.
  - Security Groups: Can specify only allow rules.
- **Statefulness**:
  - NACLs: Stateless, meaning incoming and outgoing rules need to be explicitly defined.

- Security Groups: Stateful, meaning return traffic is automatically allowed.

VPC Flow Logs:

- **Purpose**: Capture and log information about IP traffic going to and from network interfaces in your VPC.
- **Levels**: Can be enabled at the VPC, subnet, or ENI level.
- **Usage**: Useful for monitoring and troubleshooting connectivity issues.
- **Details Logged**: Information about allowed and denied traffic, and network traffic related to AWS-managed resources like ELBs, ElastiCache, RDS, Aurora, etc.
- **Data Storage**: Flow log data can be sent to Amazon S3, CloudWatch Logs, or Kinesis Data Firehose for further analysis and storage.

Summary

1. **Network ACL (NACL)**:
    - Subnet-level firewall.
    - Allows both allow and deny rules.
    - Stateless, requiring explicit rules for both inbound and outbound traffic.
2. **Security Groups**:
    - Instance or ENI-level firewall.
    - Only allows rules.
    - Stateful, allowing return traffic automatically.
3. **VPC Flow Logs**:
    - Logs all IP traffic in the VPC.
    - Helps in monitoring and troubleshooting network issues.
    - Logs can be sent to S3, CloudWatch Logs, or Kinesis Data Firehose.

Understanding these concepts is crucial for managing network security in AWS and ensuring your VPCs and subnets are protected and functioning correctly.

Connectivity in VPC

VPC Peering:

- **Purpose**: Connect two VPCs privately using AWS's network to make them behave as if they were part of the same network.
- **Setup**: Establish a VPC peering connection between VPC A and VPC B.
- **IP Range**: Ensure IP ranges of the VPCs do not overlap to avoid routing conflicts.

- **Non-Transitive**: VPC peering is not transitive, meaning a direct peering connection is needed between each pair of VPCs that need to communicate. For instance, if VPC A is peered with VPC B and VPC A is also peered with VPC C, VPC B cannot communicate with VPC C unless a separate peering connection is established between B and C.

VPC Endpoints:

- **Purpose**: Connect to AWS services using a private network instead of the public internet, providing enhanced security and lower latency.
- **Types**:
  - **Gateway Endpoints**: Specifically for S3 and DynamoDB. Allows private subnet EC2 instances to access S3 and DynamoDB without going through the internet.
  - **Interface Endpoints**: Used for other AWS services. Creates an ENI (Elastic Network Interface) in the private subnet, allowing private access to services like CloudWatch.

Site-to-Site VPN:

- **Purpose**: Connect an on-premises data center to a VPC over an encrypted public internet connection.
- **Setup**: Establish a VPN connection between an on-premises VPN appliance and AWS, creating an encrypted connection over the public internet. This is quick and easy to set up, providing secure connectivity.

Direct Connect:

- **Purpose**: Establish a private, physical connection between an on-premises data center and a VPC, ensuring secure and fast connectivity without using the public internet.
- **Setup**: Create a private line to AWS, which takes longer to establish (at least a month) due to the need for physical infrastructure setup. This provides a dedicated, secure, and reliable connection.

Summary

1. **VPC Peering**:
   - Connects two VPCs as if they were on the same network.
   - Ensure non-overlapping IP ranges.
   - Direct peering connections needed for each pair of VPCs that need to communicate.
2. **VPC Endpoints**:
   - **Gateway Endpoints**: Private access to S3 and DynamoDB.
   - **Interface Endpoints**: Private access to other AWS services via ENIs in the VPC.
3. **Site-to-Site VPN**:
   - Connects an on-premises data center to a VPC over an encrypted public internet connection.
   - Quick and easy to set up, providing secure connectivity.
4. **Direct Connect**:

- Private, physical connection between an on-premises data center and a VPC.
- Secure and fast, bypassing the public internet.
- Takes longer to establish due to the need for physical setup.

These connectivity options allow you to establish secure and efficient communication between your VPC and other networks, whether they are other VPCs, AWS services, or on-premises data centers.

Summary of VPC Concepts

1. **VPC (Virtual Private Cloud)**:
    - A private network within the AWS cloud.
    - Each AWS region has one default VPC.
2. **Subnets**:
    - Tied to specific Availability Zones.
    - Represent a network partition within your VPC.
    - Used for launching EC2 instances.
3. **Internet Gateway**:
    - Provides internet access to instances in public subnets.
    - Defined at the VPC level.
4. **NAT Gateways and NAT Instances**:
    - Provide internet access to instances in private subnets.
5. **NACLs (Network ACLs)**:
    - Stateless firewalls for subnets.
    - Control inbound and outbound traffic using allow and deny rules.
6. **Security Groups**:
    - Stateful firewalls operating at the EC2 instance or ENI level.
    - Can reference other security groups.
    - Only contain allow rules.
7. **VPC Peering**:
    - Connects two VPCs as long as their IP ranges do not overlap.
    - Non-transitive, requiring a direct peering connection for each pair of VPCs.
8. **VPC Endpoints**:
    - Provide private access to AWS services from within your VPC.

- Two types:
  - Gateway Endpoints: For S3 and DynamoDB.
  - Interface Endpoints: For other AWS services.

9. **VPC Flow Logs**:
   - Capture network traffic logs for monitoring and debugging purposes.
10. **Connectivity from On-Premises Data Centers**:
    - **Site-to-Site VPN**:
      - Encrypted connection over the public internet.
      - Quick and easy to set up.
    - **Direct Connect**:
      - Private, physical connection.
      - Secure and fast, but takes longer to establish due to the need for physical setup.

Key Points to Remember:

- **VPC and Subnets**: Understand their roles and how they are tied to Availability Zones.
- **Internet Gateway vs. NAT Gateway**: Know their differences and purposes.
- **NACLs vs. Security Groups**: Understand the differences in how they control traffic.
- **VPC Peering**: Non-transitive nature and IP range requirements.
- **VPC Endpoints**: Purpose and types (Gateway and Interface).
- **Flow Logs**: Importance for network traffic monitoring.
- **Site-to-Site VPN and Direct Connect**: Different methods for connecting on-premises data centers to AWS.

By focusing on these core concepts, you'll be well-prepared for VPC-related questions on the AWS Certified Developer exam. Don't worry if everything isn't immediately clear—revisiting these concepts as needed throughout the course will reinforce your understanding.

Three-Tier Solution Architecture on AWS

Overview

A three-tier architecture separates the application into three layers:

1. **Presentation Layer**: Handles user interaction.
2. **Logic Layer**: Processes data and business logic.
3. **Data Layer**: Manages data storage.

Implementation on AWS

1. **Elastic Load Balancer (ELB)**:

- Deployed in public subnets.
- Distributes incoming traffic across multiple EC2 instances across different availability zones (AZs).

2. **Route 53**:
    - DNS service to resolve the domain name of the ELB.
3. **EC2 Instances**:
    - Launched in an Auto Scaling group to ensure high availability and scalability.
    - Deployed in private subnets since they only need to be accessed by the ELB, not directly from the internet.
4. **Private Subnets**:
    - Host the compute resources (EC2 instances) and database resources.
    - Enhance security by isolating the compute and data layers from direct internet access.
5. **RDS (Relational Database Service)**:
    - Deployed in a separate private subnet, often referred to as a data subnet.
    - Provides a managed database service for storing application data.
6. **ElastiCache**:
    - Also deployed in the data subnet.
    - Used for caching data to improve read performance and reduce database load.
    - Can store session data for web applications.

LAMP Stack on EC2

1. **Linux**: Operating system for the EC2 instances.
2. **Apache**: Web server running on Linux.
3. **MySQL**: Database, can be run on RDS for managed service.
4. **PHP**: Application logic running on EC2 instances.
5. **ElastiCache**: Optionally added for caching (Redis or Memcached).
6. **EBS (Elastic Block Store)**: Attached to EC2 instances for persistent storage.

WordPress Architecture on AWS

1. **Load Balancer Tier**:
    - ELB to distribute traffic to the web servers.
2. **Application Tier**:
    - EC2 instances running WordPress.
    - Need to share user-uploaded content (e.g., images).
3. **EFS (Elastic File System)**:

- Provides a shared file system accessible by all EC2 instances.
- Stores shared files such as images uploaded by users.

Advanced Architecture Features

1. **NAT Gateways**:
    - Allow instances in private subnets to access the internet for updates and patches while maintaining their private IP addresses.
2. **Internet Gateway**:
    - Provides internet access to instances in public subnets.
3. **Auto Scaling Groups**:
    - Automatically adjust the number of EC2 instances based on traffic patterns.
4. **Aurora**:
    - A highly available and scalable database service for MySQL and PostgreSQL.
5. **EFS**:
    - A managed network file system that can be mounted on multiple EC2 instances simultaneously.
6. **CloudFront and S3 (Coming Soon)**:
    - CloudFront is a content delivery network (CDN) for fast content delivery.
    - S3 is an object storage service for storing large amounts of data.

Summary

- **Presentation Layer**: ELB in public subnets.
- **Logic Layer**: EC2 instances in private subnets managed by Auto Scaling groups.
- **Data Layer**: RDS and ElastiCache in data subnets.
- **LAMP Stack**: Linux, Apache, MySQL, PHP on EC2, with optional ElastiCache and EBS for storage.
- **WordPress on AWS**: Uses ELB, EC2 instances, EFS for shared storage.

This architecture provides a scalable, secure, and highly available setup for web applications, commonly used in real-world scenarios and likely to be tested in AWS certification exams.