**AWS Services & Cloud Computing:**

* **Hands-on experience** with AWS services: **Organizations, VPC, RDS, IAM, EC2, S3, SQS, and SNS**.
* **Strong knowledge of EC2**, including **application deployment, load balancing**, and **health monitoring**.
* **Implemented Auto Scaling** to ensure high availability based on web traffic.
* **Enabled S3 versioning** for data security and rollback in case of accidental deletion.
* **Configured CloudWatch alarms** for real-time monitoring and alerting.
* **Set up SNS notifications** for Auto Scaling and CloudWatch alerts.

**Containerization & Serverless:**

* **Pushed container images to ECR** and ran them on **ECS**.
* **Created ECS task definitions** for managing containerized applications.
* **Used EventBridge** to invoke **Lambda functions** for scheduling EC2 instance operations.

**Networking & Security:**

* **Well-versed in IAM roles**, enabling **multi-factor authentication** and managing policies.
* **Created CloudFront distributions** for content delivery, using **S3 as the origin**.
* **Hosted static websites on S3** with optimized configurations.

**Description:**

Developed a **Cloud-Native Infrastructure & Application Deployment on AWS**, ensuring scalability, high availability, security, and reliability. The system enables **efficient application hosting, automated scaling, secure transactions, and optimized resource management** while leveraging AWS cloud services for seamless performance.

|  |  |
| --- | --- |
| **User Authentication and Access Control:**   * Implemented IAM roles and policies to control access. * Enabled Multi-Factor Authentication (MFA) for security.   **Product Catalog & Storage:**   * Stored product images and data securely in Amazon S3. * Enabled S3 Versioning to prevent accidental data loss.   **Order Processing & Messaging Services:**   * Used SQS for asynchronous order processing. * Configured SNS notifications for real-time updates.   **Monitoring & Alerting:**   * Configured CloudWatch Alarms to monitor system health. * Integrated SNS notifications for failure alerts. | **Compute & Scaling:**   * Hosted the application backend on EC2 instances. * Enabled Auto Scaling to handle traffic spikes dynamically. * Deployed ELB (Elastic Load Balancer) to distribute traffic efficiently.   **Containerization & Deployment:**   * Pushed Docker images to ECR and deployed them via ECS. * Created ECS task definitions for managing microservices.   **Event-Driven Automation:**   * Used EventBridge to trigger Lambda functions for scheduled tasks. * Enabled automatic start/stop of EC2 instances based on business hours. |

“I have hands-on experience with key AWS services like EC2 for scalable compute resources, S3 for storage and backups, RDS for managed databases, and IAM for access control. I have implemented Auto Scaling to optimize performance based on traffic patterns and configured CloudWatch to monitor infrastructure health. Additionally, I’ve worked with containers using ECR and ECS, and set up EventBridge and Lambda for event-driven automation. CloudFront enhances content delivery while SQS and SNS facilitate messaging services. My expertise lies in integrating these services to create highly available, scalable, and secure cloud architectures.”

✔ **Organizations** – Manage multiple AWS accounts under a consolidated structure for billing and security policies.

✔ **VPC (Virtual Private Cloud)** – Provides isolated network environments for secure cloud deployments.

✔ **RDS (Relational Database Service)** – Fully managed relational database service supporting multiple engines like MySQL and PostgreSQL.

✔ **IAM (Identity and Access Management)** – Manages user roles, permissions, and authentication for secure access control.

✔ **EC2 (Elastic Compute Cloud)** – Provides scalable virtual machines to host applications.

✔ **S3 (Simple Storage Service)** – Object storage solution for storing files, backups, and hosting static websites.

✔ **SQS (Simple Queue Service)** – Message queuing service for decoupling application components and handling asynchronous processes.

✔ **SNS (Simple Notification Service)** – Enables event-driven notifications via email, SMS, or application-to-application messaging.

✔ **Auto Scaling** – Automatically adjusts compute capacity based on traffic demand for high availability.

✔ **CloudWatch** – Monitors AWS resources, collects logs, and sets alerts for performance and security incidents.

✔ **ECR (Elastic Container Registry)** – Stores and manages Docker container images for deployments.

✔ **ECS (Elastic Container Service)** – Runs and orchestrates Docker containers in a managed environment.

✔ **EventBridge** – Event-driven service that connects AWS services and third-party applications.

✔ **Lambda** – Serverless computing to run functions without provisioning or managing servers.

✔ **CloudFront** – Global content delivery network (CDN) that accelerates website and API performance.

1. **What is VPC Peering.**

VPC Peering is a networking connection between two Virtual Private Clouds (VPCs) that allows them to communicate as if they were on the same network. This enables resources in different VPCs to securely interact without needing to traverse the public internet.

**Key benefits of VPC Peering:**

* **Private & Secure Communication**: Traffic between peered VPCs stays within the cloud provider's network, ensuring security and reliability.
* **Low Latency & High Bandwidth**: Direct communication eliminates the need for intermediate networking solutions, improving performance.
* **Simplified Architecture**: No need for VPNs or gateways—VPCs can talk to each other natively.
* **Cross-Account Peering**: You can peer VPCs across different AWS accounts, making resource sharing easier.

1. **What is Transit Gateway.**

AWS Transit Gateway is a service that allows you to connect multiple VPCs and on-premises networks through a single gateway, simplifying network management and scaling across multiple environments.

### Key Features:

* **Centralized Network Control**: Acts as a hub-and-spoke model, eliminating complex VPC peering connections.
* **Scalability**: Easily connects thousands of VPCs and on-premises networks.
* **Simplified Routing**: Provides better control over network traffic with route tables.
* **Supports Multi-Region Deployments**: Enables seamless connectivity across AWS regions.

### How it Differs from VPC Peering:

* **VPC Peering** is a direct one-to-one connection between two VPCs.

**Transit Gateway** acts as a hub where multiple VPCs and on-premises networks can connect efficiently.

1. **What are Endpoints in AWS.**

AWS Endpoints are used to enable private connectivity between your Virtual Private Cloud (VPC) and AWS services without requiring internet access. This helps improve security and performance by keeping traffic within AWS's global network instead of going through the public internet.

### Types of AWS Endpoints:

1. **Interface Endpoints**: Uses AWS Private Link to create a private connection to AWS services through elastic network interfaces (ENIs) within your VPC.
2. **Gateway Endpoints**: Routes traffic to AWS services like S3 and DynamoDB through gateway endpoints in your VPC.
3. **What is R53 and routing policies.**

Amazon Route 53 (R53) is a **highly scalable and reliable Domain Name System (DNS) web service** offered by AWS. It helps route end users to applications hosted on AWS or elsewhere using custom domain names.

### **Key Features of Route 53**

* **Domain Registration**: You can register and manage domain names directly via AWS.
* **DNS Resolution**: Translates human-readable domain names into IP addresses.
* **Health Checks & Monitoring**: Ensures endpoints are operational and directs traffic accordingly.
* **Traffic Routing**: Supports various routing policies for optimized user experience.

### **Routing Policies in Route 53**

Route 53 provides different routing policies to control how DNS queries are resolved:

1. **Simple Routing**:
   * Basic DNS routing where all records return the same response.
   * Used when there's a single endpoint serving requests.
2. **Weighted Routing**:
   * Assigns weights to multiple resources, splitting traffic accordingly.
   * Example: 80% traffic to Server A, 20% to Server B.
3. **Latency-Based Routing**:
   * Directs users to the AWS region with the lowest latency.
   * Helps improve application performance globally.
4. **Failover Routing**:
   * Routes traffic to a backup resource if the primary endpoint fails.
   * Useful for high-availability architectures.
5. **Geolocation Routing**:
   * Routes users based on their geographic location.
   * Example: Users from Europe directed to EU servers, US users to US servers.
6. **Geo-Proximity Routing** (Using Route 53 Traffic Flow):
   * Similar to geolocation, but allows **bias adjustment** to favour certain regions.
   * Can shift traffic dynamically.
7. **Multi-Value Answer Routing**:
   * Returns multiple IP addresses for redundancy.
8. **What is Site to Site VPN and Direct Connect**

**1. Site-to-Site VPN**

A Site-to-Site VPN creates a secure and encrypted tunnel between your on-premises network and your AWS Virtual Private Cloud (VPC) over the public internet.

**Key Features:**

* Uses IPsec tunnels for encryption and security.
* Connects your data centre or corporate network to AWS via a VPN gateway.
* Quick setup without needing a dedicated physical connection.
* Lower cost, but higher latency compared to Direct Connect.

**Use Case:**

**Ideal for businesses needing secure access to AWS without high data transfer demands.**

**2. AWS Direct Connect**

AWS Direct Connect is a dedicated physical connection from your on-premises data center to AWS. It bypasses the internet, offering higher performance and lower latency.

**Key Features:**

* Provides a private, dedicated high-speed network link.
* Delivers consistent, low-latency, and high bandwidth connectivity.
* Supports multiple VPCs via AWS Transit Gateway or Virtual Interfaces.
* Typically, more expensive than VPN but better for large data transfers.

**Use Case:**

Perfect for enterprises needing reliable, high-speed, and secure access to AWS for large-scale workloads.

1. **What is Global Accelerator**

AWS **Global Accelerator** is a networking service that improves the availability, performance, and security of applications deployed across multiple AWS regions. It helps direct traffic to the optimal endpoint using AWS' global network instead of relying on the public internet.

### **Key Features of AWS Global Accelerator**

1. **Global Traffic Optimization**
   * Routes requests to the best-performing AWS endpoint based on health checks and proximity.
   * Supports **Elastic Load Balancers (ELB), EC2 instances, and Amazon EKS clusters**.
2. **Improved Latency & Performance**
   * Uses AWS' private global network to reduce **internet routing inefficiencies**.
   * Enhances application responsiveness by **reducing packet loss and jitter**.
3. **Automatic Failover & High Availability**
   * Continuously monitors endpoint health and **automatically redirects traffic** in case of failure.
   * Ensures seamless user experience even during outages.
4. **Static Anycast IP Addresses**
   * Provides **two static IPs** for the accelerator, ensuring **consistent routing**.
   * Users do not need to update DNS records when endpoints change.
5. **Multi-Region Deployment Support**
   * Ensures low-latency connections to the closest AWS region.
   * Ideal for global applications needing high performance.
6. **Use CloudFront** if you need **fast, cached delivery** of static and dynamic web content (e.g., images, videos, websites).
7. **Use Global Accelerator** if you need **low-latency, high-availability** routing for global applications (e.g., gaming servers, multi-region APIs).