**1. Amazon S3 - Data Storage and Management**

* Scenario: A company is planning to store large volumes of unstructured data such as media files, backups, and logs. They need a cost-effective solution with scalable storage that allows for easy access and management of their data. The company also wants to ensure that their data is secure, with backup policies and lifecycle management in place.
* **Question**: Explain how Amazon S3 can be used to address the company’s storage needs. Discuss the different storage classes in S3 and how they can help optimize costs. Include how you would implement security, backup policies, and lifecycle management to meet the company’s requirements.

**2. EC2 - Launching and Managing Instances**

* Scenario: A startup is launching a new web application and needs to deploy it on AWS EC2 instances. The application needs to be highly available, scalable, and easily manageable. The team is also considering utilizing EC2 Auto Scaling and Elastic Load Balancers to distribute traffic efficiently.
* **Question**: Describe the steps involved in launching an EC2 instance for the application, including choosing the appropriate instance type and configuration. How would you integrate EC2 Auto Scaling and Elastic Load Balancing to ensure high availability and scalability? Provide the advantages of using these services in the given scenario.

**3. Load Balancers - Traffic Distribution**

* Scenario: A growing e-commerce company is experiencing increased traffic to their website, leading to performance degradation. To ensure the website remains available and responsive, the company plans to use Elastic Load Balancing (ELB) to distribute traffic across multiple EC2 instances.
* **Question**: Explain how Elastic Load Balancer (ELB) works and the types of load balancers available in AWS. How would you configure an ELB to distribute traffic evenly across multiple EC2 instances? What are the benefits of using an ELB for this scenario?

**4. Auto Scaling - Ensuring High Availability**

* Scenario: A media streaming company needs to ensure that their application can handle sudden spikes in user demand, especially during peak usage hours. They want to use AWS Auto Scaling to automatically adjust the number of EC2 instances based on traffic demand.
* **Question**: Discuss how AWS Auto Scaling can be configured to automatically scale the EC2 instances in response to traffic demands. What factors would you consider when setting up Auto Scaling policies, and how can you ensure that the application remains available during high-demand periods?

**5. IAM - User Management and Access Control**

* Scenario: A software development company is working on a project that involves sensitive data. The team needs to manage user access to AWS resources, with specific roles for developers, testers, and system administrators. The company wants to ensure that each user has the least privilege necessary to perform their job.
* **Question**: Describe how AWS Identity and Access Management (IAM) can be used to manage user access to AWS resources. Discuss the steps involved in creating IAM roles for different team members (developers, testers, administrators) and implementing the principle of least privilege. How can you monitor and audit user activities in AWS?

**1. Amazon S3 - Data Storage and Management**

**Answer:**

Amazon Simple Storage Service (S3) is an object storage service that provides scalability, security, and durability for storing large amounts of data.

**How Amazon S3 Addresses Storage Needs:**

* **Scalability**: S3 automatically scales to handle increasing storage needs.
* **Cost-effectiveness**: Offers different storage classes to optimize costs.
* **Security**: Data encryption, access control policies, and bucket permissions ensure security.
* **Durability & Availability**: Ensures 99.999999999% (11 nines) durability and high availability.

**S3 Storage Classes:**

1. **S3 Standard** – Suitable for frequently accessed data.
2. **S3 Intelligent-Tiering** – Automatically moves data between two tiers based on access patterns.
3. **S3 Standard-IA (Infrequent Access)** – For less frequently accessed data.
4. **S3 One Zone-IA** – Lower cost, stored in a single Availability Zone.
5. **S3 Glacier & Glacier Deep Archive** – Used for long-term archival storage.

**Security, Backup, and Lifecycle Management:**

* **Security Measures:**
  + IAM policies to restrict access.
  + Bucket policies and ACLs to control permissions.
  + Server-side encryption (SSE) for data protection.
* **Backup Policies:**
  + Versioning to keep track of file changes.
  + Cross-region replication (CRR) for redundancy.
* **Lifecycle Management:**
  + Set lifecycle rules to transition objects to lower-cost storage classes or delete them after a set period.

**2. EC2 - Launching and Managing Instances**

**Answer:**

Amazon Elastic Compute Cloud (EC2) provides scalable virtual machines to host applications.

**Steps to Launch an EC2 Instance:**

1. **Choose an Amazon Machine Image (AMI)** – Select an OS like Ubuntu, Windows, or Amazon Linux.
2. **Choose Instance Type** – Based on CPU, memory, and network requirements (e.g., t2.micro for free tier).
3. **Configure Instance Details** – Select VPC, subnet, IAM roles, and key pairs.
4. **Add Storage** – Attach EBS volumes (General Purpose SSD, Provisioned IOPS SSD).
5. **Configure Security Group** – Allow necessary inbound/outbound rules.
6. **Review & Launch** – Connect using SSH (ssh -i key.pem ec2-user@public-ip).

**Integrating Auto Scaling & Load Balancer:**

* **Auto Scaling:**
  + Automatically adjusts EC2 instances based on traffic.
  + Uses scaling policies (CPU utilization, request count).
* **Elastic Load Balancer (ELB):**
  + Distributes traffic across instances.
  + Ensures high availability and fault tolerance.
  + Supports different load balancers (ALB, NLB, CLB).

**Advantages:**

* **Elasticity** – Instances scale dynamically based on demand.
* **High Availability** – ELB ensures no single point of failure.
* **Cost Efficiency** – Auto Scaling prevents over-provisioning.

**3. Load Balancers - Traffic Distribution**

**Answer:**

AWS Elastic Load Balancer (ELB) distributes incoming traffic across multiple EC2 instances.

**How ELB Works:**

1. Accepts client requests and routes them to available EC2 instances.
2. Performs health checks to ensure instance availability.
3. Supports SSL termination and automatic scaling.

**Types of Load Balancers:**

1. **Application Load Balancer (ALB)** – Works at the application layer (Layer 7).
2. **Network Load Balancer (NLB)** – Works at the transport layer (Layer 4), handles high-performance needs.
3. **Classic Load Balancer (CLB)** – Legacy option for basic load balancing.

**Configuring an ELB:**

* **Step 1:** Create a Load Balancer from the AWS Management Console.
* **Step 2:** Select target instances or a target group.
* **Step 3:** Configure listener rules (e.g., HTTP, HTTPS).
* **Step 4:** Set up security groups and health checks.

**Benefits of ELB:**

* **Improved Performance** – Distributes traffic evenly.
* **Fault Tolerance** – Redirects traffic if an instance fails.
* **Scalability** – Works with Auto Scaling for dynamic growth.

**4. Auto Scaling - Ensuring High Availability**

**Answer:**

AWS Auto Scaling automatically adjusts the number of EC2 instances based on demand.

**Configuring Auto Scaling:**

1. **Define Launch Template** – Specify AMI, instance type, security groups.
2. **Create Auto Scaling Group (ASG)** – Attach instances across multiple Availability Zones.
3. **Set Scaling Policies:**
   * **Target Tracking Scaling** – Maintain a specific metric (e.g., CPU utilization at 50%).
   * **Step Scaling** – Scale up/down based on predefined thresholds.
   * **Scheduled Scaling** – Scale at specific times.

**Factors for Setting Up Auto Scaling:**

* **Traffic Load** – Number of requests per second.
* **Resource Utilization** – CPU, memory, disk usage thresholds.
* **Fault Tolerance** – Deploy instances in multiple Availability Zones.

**Ensuring High Availability:**

* **Health checks** – Terminate unhealthy instances.
* **Multi-AZ deployment** – Instances spread across different zones.
* **Elastic Load Balancer (ELB) integration** – Ensures traffic is balanced.

**Advantages:**

* **Efficiency** – Saves costs by scaling down during low demand.
* **Resilience** – Automatically replaces failed instances.
* **Performance** – Handles sudden traffic spikes without manual intervention.

**5. IAM - User Management and Access Control**

**Answer:**

AWS Identity and Access Management (IAM) manages user access to AWS services.

**Managing User Access in AWS IAM:**

* **IAM Users** – Individual accounts with specific permissions.
* **IAM Groups** – Collection of users with shared permissions.
* **IAM Roles** – Assignable permissions for AWS services (e.g., EC2, Lambda).
* **IAM Policies** – JSON-based permissions defining allowed actions.

**Steps to Create IAM Roles for Developers, Testers, and Administrators:**

1. **Create IAM Users** – Assign unique credentials.
2. **Define IAM Groups:**
   * **Developers** – Read/write access to development resources.
   * **Testers** – Read-only access to test environments.
   * **Administrators** – Full access to AWS services.
3. **Attach Policies to Groups:**
   * Use AWS managed policies or create custom ones.
   * Example: AmazonS3FullAccess, EC2ReadOnlyAccess.

**Implementing the Principle of Least Privilege:**

* Grant only necessary permissions per role.
* Use IAM Conditions to restrict access by IP, time, or service.
* Enable Multi-Factor Authentication (MFA) for added security.

**Monitoring and Auditing User Activities:**

* **AWS CloudTrail** – Logs all user actions.
* **AWS Config** – Tracks configuration changes.
* **IAM Access Analyzer** – Identifies potential security risks.

**Advantages:**

* **Security** – Prevents unauthorized access.
* **Compliance** – Ensures regulatory requirements are met.
* **Granular Control** – Defines precise permissions for different users.