Rakshita Belwal

Advanced AWS Scenario-Based Interview Questions & Answers

1. Designing a Scalable Web Application with Auto Scaling

Scenario:

You're building a web application that will experience variable traffic (e.g., seasonal e-commerce). How do you ensure scalability and availability while keeping costs low?

Answer:

Use EC2 Auto Scaling Group behind an Application Load Balancer (ALB). Configure scaling policies based on CPU or request count. Use On-Demand + Spot Instances mix. Store static assets on S3 with CloudFront, and use RDS Multi-AZ for database reliability.

---

2. Securing Access to AWS Services in a Multi-Team Setup

Scenario:

Your company has multiple teams working in the same AWS account. How do you ensure they have access only to what they need?

Answer:
Use IAM roles and policies with least privilege. Group users via IAM Groups. Use AWS Organizations to create SCPs and isolate workloads per team using AWS accounts and consolidated billing.

---

3. Failover Between Regions for High Availability

Scenario:
You need to design an architecture that is resilient even if an entire AWS region goes down.

Answer:
Deploy app in two regions. Use Route 53 with failover routing. S3 cross-region replication, RDS Read Replicas or Aurora Global DBs, and multi-region DynamoDB tables can ensure data availability.

---

4. Cost Optimization for Serverless App

Scenario:
Your Lambda-based app is seeing high invocation costs. How do you reduce it?

Answer:
Analyze usage in CloudWatch Logs. Optimize function memory and runtime duration. Use asynchronous invocations where possible

and batch processing with SQS. Use Compute Savings Plans for Lambda.

---

5. Sudden Traffic Spike on EC2 Instance

Scenario:

Your application hosted on an EC2 instance is experiencing sudden traffic spikes and going down. How would you handle this using AWS services?

Answer:

Use Auto Scaling Groups with a Load Balancer (ELB). The ELB will distribute traffic across instances. Auto Scaling automatically launches new EC2 instances to meet the demand and terminates them when demand decreases. Also, use CloudWatch alarms to trigger scaling based on CPU utilization or other metrics.

---

6. Managing Secrets Securely

Scenario:

Your application needs access to database credentials. How will you manage secrets securely in AWS?

Answer:

Use AWS Secrets Manager or AWS Systems Manager Parameter Store. These services allow secure storage, automatic rotation, and access control using IAM policies. Secrets Manager supports fine-grained access and versioning for credentials.

7. High Availability for RDS

Scenario:
Your application relies heavily on a MySQL database. How do you ensure high availability?

Answer:
Use Amazon RDS Multi-AZ deployment, which automatically creates a synchronous standby replica in a different AZ. In case of failure, it automatically fails over to the standby. For better read performance, Read Replicas can be used (though they are not for HA, just performance).

---

8. Data Backup and Restore Strategy

Scenario:
How would you implement a robust backup strategy for EC2 and RDS instances?

Answer:
For EC2, use Amazon Data Lifecycle Manager to automate EBS snapshot creation and retention. For RDS, enable automated backups and create manual snapshots before critical updates. Store backups in Amazon S3 Glacier for long-term archival.

---

9. Handling S3 Bucket Security

Scenario:
You discover that your S3 bucket is publicly accessible. What steps do you take?

Answer:

- Immediately block public access using **S3 Block Public Access settings**.
- Review and remove any overly permissive **bucket policies** and **ACLs**.
- Use **IAM policies** for controlled access.
- Enable **CloudTrail** to track who accessed the data.
- Enable **S3 encryption** for sensitive data.

---

10. Cost Optimization for Idle Resources

Scenario:
How do you identify and reduce the cost of unused AWS resources?

Answer:
Use AWS Cost Explorer and Trusted Advisor to identify underutilized resources like idle EC2s, EBS volumes, unattached IPs, and unused Load Balancers. Terminate or right-size them. Use Savings Plans and Reserved Instances for predictable workloads.

11. Zero Downtime Deployment

Scenario:

How can you deploy a new version of your app with zero downtime?

Answer:

Use Blue/Green deployments with AWS CodeDeploy. Set up two environments—Blue (current) and Green (new). Route traffic to the Green once it's tested. Roll back to Blue if needed. Combine with Elastic Load Balancer and Auto Scaling for seamless deployment.

12. Detecting and Mitigating DDoS Attacks

Scenario:

Your application is under a DDoS attack. What AWS services will help mitigate it?

Answer:

Use AWS Shield (Basic for free, Advanced for enterprise). Deploy AWS WAF for filtering requests. Place your application behind a CloudFront CDN and ALB for absorbing traffic. Enable rate limiting rules and IP whitelisting/blacklisting in WAF.

13. Compliance and Auditing

Scenario:

How do you ensure compliance and auditing for resource changes?

Answer:

Use AWS CloudTrail to log all API calls. AWS Config tracks resource configurations over time. Enable AWS Audit Manager to continuously assess and monitor compliance posture. Integrate with Security Hub for centralized findings.

---

14. Infrastructure as Code

Scenario:

How do you manage and deploy your infrastructure repeatedly and reliably?

Answer:

Use AWS CloudFormation or Terraform. Write templates to define infrastructure. These can be version-controlled, reused, and deployed consistently across environments. Automate deployments through CI/CD pipelines (CodePipeline or Jenkins).

---

15. Infrastructure as Code (IaC) using CloudFormation

Scenario:
You've been asked to replicate a production environment in staging using CloudFormation. What steps will you take?

Answer:

- Review the production stack's CloudFormation template.
- Parameterize the template (like instance type, AMI, etc.) for reusability.
- Validate the template using the **`aws cloudformation validate-template`** CLI.
- Launch the template in the staging environment with appropriate parameters.
- Use change sets to preview changes before deployment.
- Ensure all IAM permissions, VPC, and security groups are appropriate for staging.

---

16. Cost Optimization

Scenario:
Your AWS bill has spiked unexpectedly. How will you investigate and optimize it?

Answer:

- Use **AWS Cost Explorer** to identify high-cost services.

- Check for unused EC2 instances, EBS volumes, and Elastic IPs.
- Use **Trusted Advisor** for cost-saving recommendations.
- Consider Reserved Instances or Savings Plans for predictable workloads.
- Set up billing alerts with **AWS Budgets**.

---

17. Auto Scaling Misconfiguration

Scenario:

Your application is not scaling properly during peak traffic hours. How will you troubleshoot Auto Scaling?

Answer:

- Review scaling policies and CloudWatch alarms.
- Ensure metrics like CPUUtilization are correctly defined.
- Check EC2 instance limits (soft limits) in that region.
- Validate the health check settings (ELB or EC2).
- Look into launch configurations and templates.

---

18. Disaster Recovery

Scenario:
Your primary region has gone down. How will you ensure business continuity using AWS?

Answer:

- Use Route 53 with health checks and DNS failover to redirect traffic.
- Maintain infrastructure in a secondary region using CloudFormation or Terraform.
- Enable cross-region replication for S3 and backups for RDS.
- Use Multi-AZ and Multi-Region architecture for high availability.

---

19. Deploying Microservices

Scenario:
You need to deploy a microservices-based application on AWS. How would you design it?

Answer:

- Use ECS or EKS for container orchestration.
- Store container images in ECR.
- Use API Gateway and AWS Lambda for serverless parts.
- Set up communication via service mesh or internal ALBs.
- Monitor using CloudWatch and X-Ray.

20. CI/CD in AWS

Scenario:

How would you implement a CI/CD pipeline using AWS services?

Answer:

- Use CodeCommit for source code.
- Set up CodeBuild for build/test stages.
- Use CodeDeploy for deployment to EC2, Lambda, or ECS.
- Manage the entire flow with CodePipeline.
- Integrate manual approvals and rollback steps.

---

21. S3 Data Protection

Scenario:

Your client requires that no one should accidentally delete data from S3. What would you do?

Answer:

- Enable versioning on the S3 bucket.
- Set up MFA delete for extra protection.
- Apply bucket policies and IAM roles with least privileges.
- Enable Object Lock (compliance mode) for regulatory data.

---

22. Real-Time Log Monitoring

Scenario:

You need to analyze logs in real-time and trigger alerts. How would you do it?

Answer:

- Stream logs from EC2 or Lambda to CloudWatch Logs.
- Set up metric filters and create CloudWatch Alarms.
- Use AWS Lambda or SNS to send alerts.
- Optionally use Kinesis Firehose to deliver logs to S3/Elasticsearch for advanced analytics.

---

23. Securing a Web Application

Scenario:

What AWS services and practices would you use to secure a public web application?

Answer:

- Deploy behind Application Load Balancer (ALB).
- Use WAF to filter common threats.
- Enable Shield Standard for DDoS protection.
- Secure data at rest with KMS and in transit with TLS.
- Enable logging with CloudTrail and monitor with GuardDuty.