**1. .NET Core Fundamentals**

**General Architecture**

1. What are the key differences between .NET Framework and .NET Core?
2. How does the middleware pipeline work in ASP.NET Core?
3. Explain Dependency Injection (DI) in .NET Core and its benefits.
4. How do you handle configuration management in a .NET Core application?
5. What are the different hosting models available in .NET Core?

**Web API & Microservices**

1. How do you implement **RESTful APIs** using .NET Core?
2. Explain **gRPC** and when you would use it over REST APIs.
3. How do you handle versioning in ASP.NET Core Web API?
4. How would you implement **caching** in a .NET Core microservice?
5. What are the different ways to implement authentication and authorization in ASP.NET Core?

**2. Microservices & Event-Driven Architecture**

**Design & Best Practices**

1. What are the benefits and challenges of **microservices** architecture?
2. How do you handle **inter-service communication** between microservices?
3. Explain the **Saga Pattern** and how you would implement it in .NET Core.
4. What strategies can be used to manage database consistency across microservices?
5. How do you ensure security in a microservices-based architecture?

**Event-Driven Systems**

1. How would you implement **asynchronous messaging** in microservices using AWS services?
2. What is **event sourcing**, and how can it be implemented in a .NET Core application?
3. How do you implement **distributed tracing** in microservices?
4. What is the **Circuit Breaker** pattern, and how do you implement it in .NET Core?
5. What is **rate limiting**, and how can it be applied in API design?

**3. AWS Cloud Architecture**

**Compute & Containers**

1. What are the differences between **AWS Lambda, ECS, and EKS**, and when would you use each?
2. How do you deploy a .NET Core application using **AWS Elastic Beanstalk**?
3. What is **AWS Fargate**, and how does it differ from ECS?
4. How do you configure **auto-scaling** for a .NET Core microservice on AWS?
5. How do you set up **CI/CD pipelines** for deploying .NET Core applications on AWS?

**Networking & Security**

1. What is **Amazon API Gateway**, and how does it work with .NET Core microservices?
2. How would you implement **JWT authentication** with AWS Cognito in a .NET Core API?
3. How do you configure **AWS VPC** for secure microservices deployment?
4. What are **IAM roles and policies**, and how do they apply to .NET Core applications on AWS?
5. How do you implement **cross-region deployment** for high availability?

**4. Data & Storage**

**Databases & Persistence**

1. How would you decide between using **RDS (SQL Server) and DynamoDB** in a .NET Core project?
2. What are **DynamoDB best practices** for performance optimization?
3. How do you handle **database migrations** in a .NET Core application using EF Core?
4. What is **Amazon Aurora**, and how does it differ from RDS?
5. How do you implement **data replication** across AWS regions?

**Caching & Performance**

1. How would you implement **Redis caching** in a .NET Core microservice on AWS?
2. What is **AWS ElastiCache**, and how does it work with .NET Core?
3. How do you optimize performance in a **serverless** .NET Core application?
4. How would you set up **AWS S3** for storing application data?
5. What strategies do you use for **scaling database reads and writes** in a cloud-based system?

**5. DevOps & Monitoring**

**CI/CD & Automation**

1. What are the key components of an **AWS DevOps pipeline** for .NET Core?
2. How do you configure **GitHub Actions / AWS CodePipeline** for automatic deployments?
3. What is **Blue-Green Deployment**, and how would you implement it for a .NET Core application?
4. How do you manage **infrastructure as code** using AWS CloudFormation or Terraform?
5. What is **AWS App Mesh**, and how does it support microservices?

**Logging & Monitoring**

1. How do you set up **centralized logging** for .NET Core microservices on AWS?
2. What are the key differences between **AWS CloudWatch and AWS X-Ray**?
3. How would you implement **application health checks** in .NET Core?
4. What are some common **performance bottlenecks** in .NET Core applications, and how do you debug them?
5. How do you use **Amazon GuardDuty or AWS WAF** for security monitoring?

**Scenario-Based Questions**

1. **You need to migrate a monolithic .NET Core application to AWS using microservices. How would you design the migration strategy?**
2. **A .NET Core microservice deployed on AWS ECS is experiencing slow response times. How would you diagnose and fix the issue?**
3. **Your API Gateway is receiving excessive traffic, and some requests are failing. How do you implement rate limiting and load balancing?**
4. **You need to integrate an on-premise SQL Server database with AWS-based .NET Core microservices. What approach would you take?**
5. **A production .NET Core service crashed due to an unhandled exception. How do you prevent similar failures in the future?**

**How to Prepare?**

1. **Hands-on Practice**: Build and deploy .NET Core microservices on AWS (ECS, Lambda, API Gateway).
2. **AWS Certifications**: Consider **AWS Certified Solutions Architect - Associate** for cloud architecture knowledge.
3. **Performance Tuning**: Learn .NET Core optimization techniques, caching, and logging.
4. **Security Best Practices**: Implement **IAM policies, JWT authentication, and VPC security**.
5. **DevOps & Automation**: Work with **Terraform, Docker, Kubernetes, and CI/CD pipelines**.