**1. What is AWS Lambda?**

AWS Lambda is a serverless compute service that lets you run code without provisioning or managing servers. It automatically scales and manages the infrastructure required to run your code in response to events.

**2. How does AWS Lambda work?**

You can upload your code to Lambda and define event sources that trigger the execution of your code. Lambda automatically manages the execution environment, scales it as needed, and provides monitoring and logging.

**3. What are the key benefits of using AWS Lambda?**

The benefits of AWS Lambda include automatic scaling, reduced operational overhead, cost efficiency (as you pay only for the compute time used), and the ability to build event-driven architectures.

**4. What types of events can trigger AWS Lambda functions?**

AWS Lambda functions can be triggered by various event sources, such as changes in Amazon S3 objects, updates to Amazon DynamoDB tables, HTTP requests through Amazon API Gateway, and more.

**5. How is concurrency managed in AWS Lambda?**

Lambda automatically handles concurrency by scaling out instances of your function in response to incoming requests. You can set a concurrency limit to control how many concurrent executions are allowed.

**6. What is the maximum execution duration for a single AWS Lambda invocation?**

The maximum execution duration for a single Lambda invocation is 15 minutes.

**7. How do you pass data to and from AWS Lambda functions?**

You can pass data to Lambda functions through event objects, which contain information about the triggering event. You can also return data by using the return statement or creating a response object.

**8. Can AWS Lambda functions communicate with external resources?**

Yes, Lambda functions can communicate with external resources such as databases, APIs, and other AWS services by using appropriate SDKs and APIs provided by AWS.

**9. What are AWS Lambda layers?**

AWS Lambda layers are a way to manage and share code that is common across multiple functions. Layers can include libraries, custom runtimes, and other function dependencies.

**10. How can you handle errors in AWS Lambda functions?**

You can handle errors by using try-catch blocks in your code. Lambda also provides CloudWatch Logs for monitoring, and you can set up error handling and retries for asynchronous invocations.

**11. Can AWS Lambda functions access the internet?**

Yes, Lambda functions can access the internet through the Virtual Private Cloud (VPC) or through public endpoints if your function is not configured within a VPC.

**12. What are the execution environments available for AWS Lambda functions?**

Lambda supports several runtimes, including Node.js, Python, Java, Go, Ruby, .NET Core, and custom runtimes using the Runtime API.

**13. How can you configure environment variables for AWS Lambda functions?**

You can set environment variables for Lambda functions when creating or updating the function. These variables can be accessed within your code.

**14. What is the difference between synchronous and asynchronous invocation of Lambda functions?**

Synchronous invocations wait for the function to complete and return a response, while asynchronous invocations return immediately, and the response is sent to a specified destination.

**15. What is the AWS Lambda Event Source Mapping?**

Event Source Mapping allows you to connect event sources like Amazon DynamoDB streams or Amazon Kinesis streams to Lambda functions. This enables the function to process events as they occur.

**16. How can you manage the permissions and execution roles for AWS Lambda functions?**

You can use AWS Identity and Access Management (IAM) roles to grant permissions to your Lambda functions. Execution roles define what AWS resources the function can access.

**17. What is AWS Step Functions?**

AWS Step Functions is a serverless orchestration service that lets you coordinate multiple AWS services into serverless workflows using visual workflows called state machines.

**18. How can you automate the deployment of AWS Lambda functions?**

You can use AWS Serverless Application Model (SAM) templates, AWS CloudFormation, or CI/CD tools like AWS CodePipeline to automate the deployment of Lambda functions.

**19. Can AWS Lambda functions connect to on-premises resources?**

Yes, Lambda functions can connect to on-premises resources by placing the function inside a VPC and using a VPN or Direct Connect connection to establish connectivity.

**20. What is the Cold Start issue in AWS Lambda?**

The Cold Start issue occurs when a Lambda function is invoked for the first time or after it has been idle for a while. The function needs to be initialized, causing a slight delay in response time.