**What is lambda?**

***AWS lambda is a serverless compute service offered by Amazon Web Services (AWS) that enables you to run code without provisioning or managing servers. With lambda, you can execute your code in response to various events, such as HTTP requests, file uploads to Amazon S3, updates to DynamoDB tables, or messages from Amazon SQS. AWS automatically handles the underlying infrastructure, including server maintenance, scaling, and monitoring, allowing you to focus solely on your application logic.***

**Key features of AWS lambda**

***.Event-Driven Execution***

Lambda runs code in response to events, like:

.file uploads to S3

.API calls via API Gateway

.Table changes in DynamoDB

**.Automatic Scaling :**

.Lambda scales automatically with the number of requests.

.Whether it’s 1 request or 1million , Lambda adjusts instantly.

**.Flexible Language Support :**

.You can write function in python, Node.js, Java, Go, C#, Ruby ,and more.

**.Cost Efficiency :**

.You only pay for the time your code runs (in milliseconds).

.No changes when your code is not running – great for infrequently workloads.

**.Integration with AWS Services:**

.S3, DynamoDB, SNS, SQS

.ECS/EKS (for container-based application)

.CloudWatch for logs and monitoring

**Common Use Cases**

***. Web and Mobile Backends:***

***. Real-Time file Processing :***

***. Data transformation and ETL:***

***. IoT Data Processing:***

***. Scheduled Tasks:***

**How AWS Lambda Works**

**1. Function Creation : Develop your code and package it as a lambda function, specifying the runtime environment and necessary permissions.**

**2. Event Source Configuration: Configure triggers from AWS services or external sources that will invoke your Lambda function when specific events occur.**

**3. Execution and Scaling: When triggered, Lambda runs your function in a secure and isolated environment, automatically managing the compute resources and scaling as needed.**

**4. Monitoring and logging: AWS provides built-in monitoring through Amazon CloudWatch, allowing you to track function execution metrics and logs for debugging and performance optimization.**

**Basic question-**

**What is AWS Lambda?**

**What languages does AWS Lambda support?**

**How is lambda priced?**

**What is the maximum execution time for a Lambda function?**

**What are the Lambda layers and how are they used?**

**How do you manage environment variables security in Lambda?**

**What are some best practices for developing Lambda function in production?**

**How can you trigger an AWS Lambda function?**

**What is a cold start in AWS Lambda? How can it be mitigated?**

**Steps to Use AWS Lambda**

1. ***Login to AWS Console***

**2.Create a Lambda function**

**1. Go to Lambda service.**

**2. Click on “Create function”.**

**3. Choose Author from scratch:**

. Function name : My Lambda function

.Runtime: Select your language (e.g. Python, js, Java, Node)

**4.Click Create function.**

**3. Write or upload your code**

**You can :**

.Write inline in the Lambda editor.

.Upload a.zip file with your code.

.Use S3 or Container images for longer apps.

def lambda\_handler(event, context):

return {

'statusCode': 200,

'body': 'Hello from Lambda!'

}

**4. Set a Trigger**

You can trigger Lambda with:

.API Gateway (for web apps)

.S3 (when a file is uploaded)

.CloudWatch (for scheduled tasks)

.DynamoDB, SNS, SQS, etc.

**5. Test the function**

.Use the TEST button in Lambda service

.Configure a test event

.Click Invoke to see output

**6. Monitor execution**

**.Use Amazon CloudWatch Logs to:**

.View logs

.Monitor performance

.Debug error