

AWS Lambda

Tessema Mengistu (Ph.D.)

mengistu@vt.edu

Outline

- Overview of AWS Lambda
- Basic Components of AWS Lambda

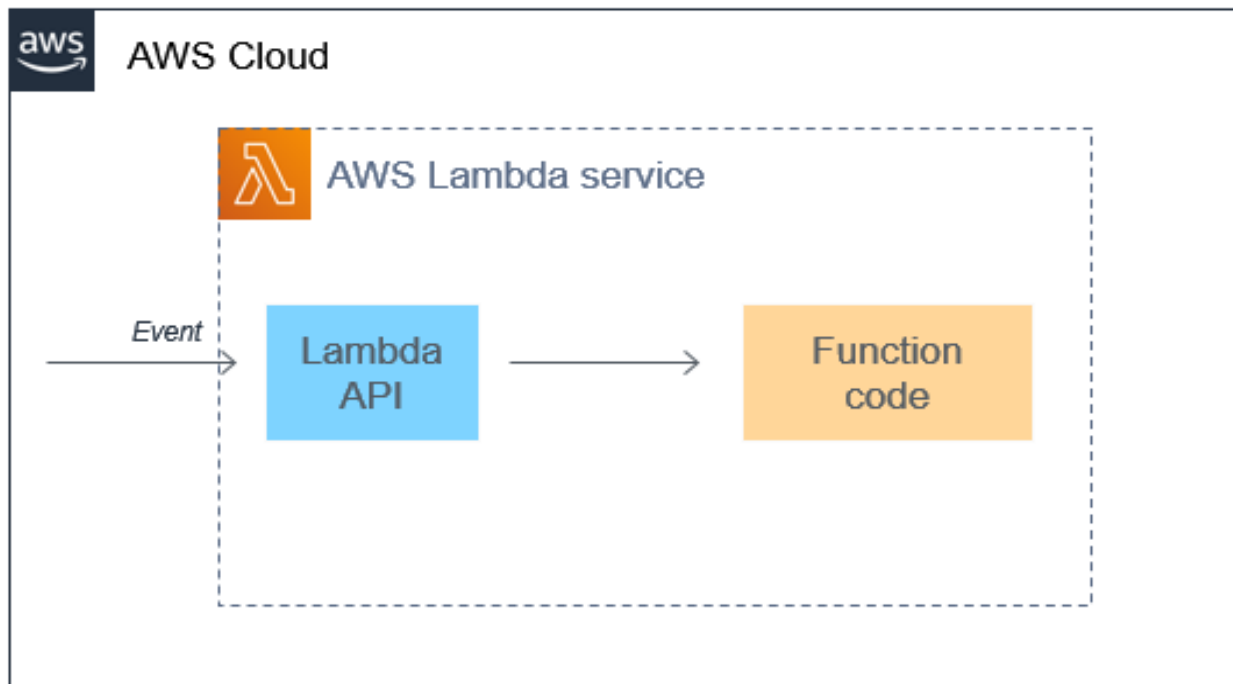
Overview of AWS Lambda

- AWS compute services can be:
 - IaaS
 - Container based
 - Serverless
- Different services
 - Amazon Elastic Compute Cloud (EC2)
 - Amazon Elastic Container Service (ECS)
 - Amazon EC2 Auto Scaling
 - AWS Fargate
 - AWS Elastic Beanstalk
 - AWS Lambda
 - ...

Overview of AWS Lambda

- AWS Lambda
 - Lambda is an on-demand compute service that runs custom code in response to events
 - Fully managed, **event driven**, serverless compute service
 - FaaS
 - Uses Lambda functions
 - Can be invoked using SDK, Lambda API, and console
 - Supports multiple programming languages through **runtimes**
 - Java, Go, Python, C#, JavaScript, Node.js, .NET, Ruby
 - Built-in fault tolerance

Overview of AWS Lambda

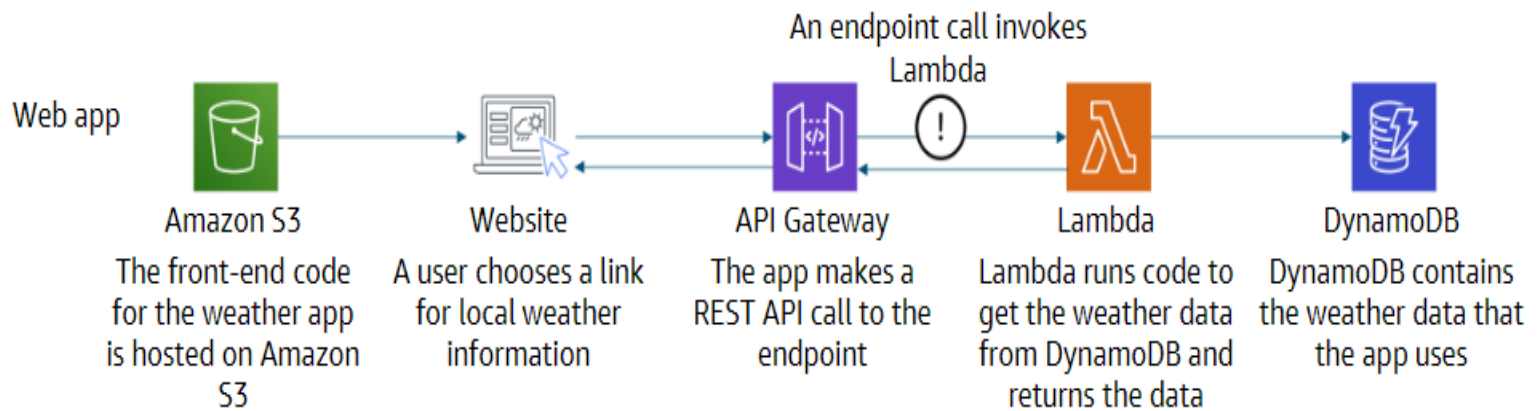


Overview of AWS Lambda

- AWS lambda provides a functionality :
 - To integrate user written code
 - To integrate and extend other AWS services
 - AWS SDK
 - Resource and concurrency model
 - Permission model
 - Built-in fault tolerance and HA
- Some useful use cases include:
 - Web applications
 - Backend
 - Data processing
 - Automating repetitive tasks
 - ...

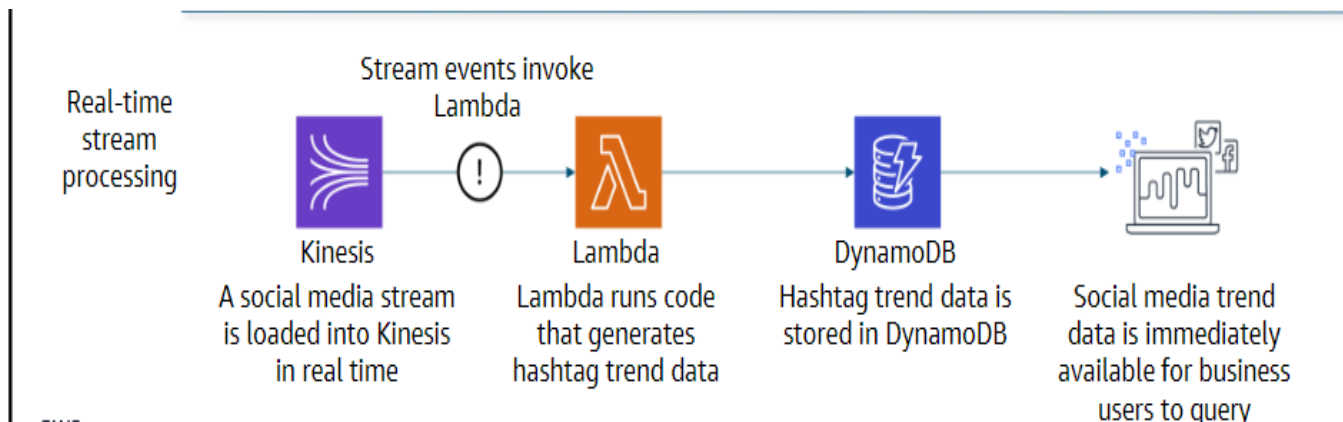
Overview of AWS Lambda

- Web Application use case



Overview of AWS Lambda

- Stream Processing use case



Basic components of AWS Lambda

- Basic components of AWS Lambda – Lambda programming model
 - Handler
 - Lambda function
 - Event and Event source
 - Context
 - Passed as second parameter to the Handler
 - Exception
 - Logging

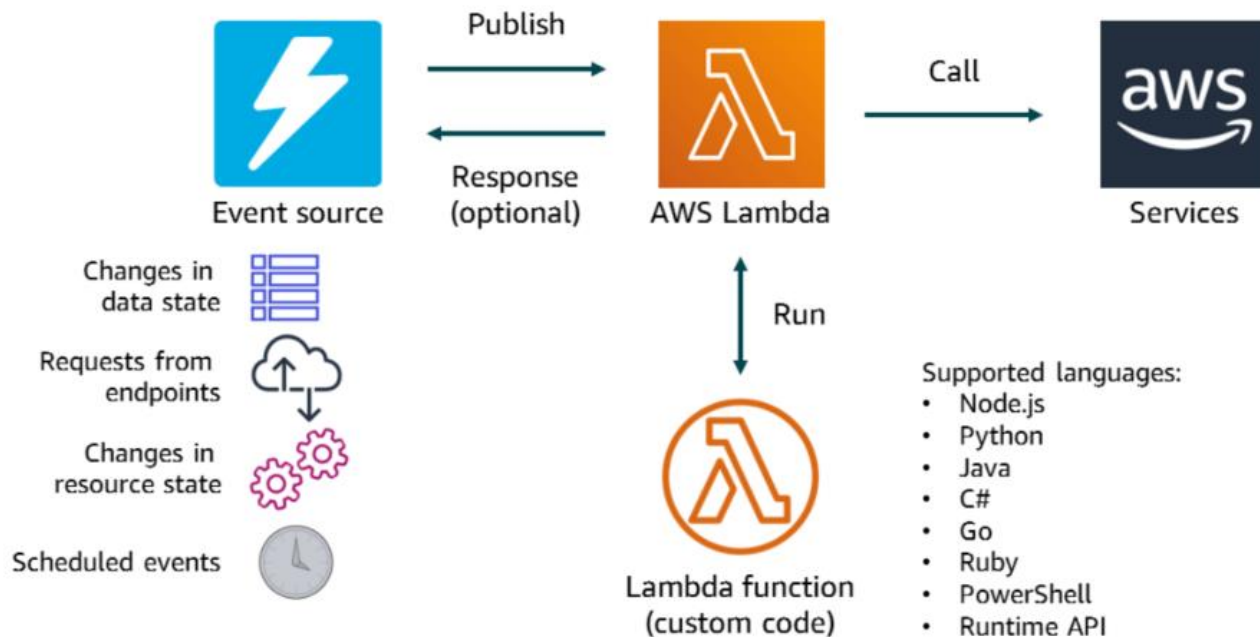
Basic components of AWS Lambda

- Events
 - Facts about a change in the system state
 - They are immutable JSON objects
 - Trigger lambda functions
 - Example: an HTTP request through API Gateway, a schedule managed by an EventBridge rule, an IOT event, or an S3 event

Basic components of AWS Lambda

- Event sources
 - Entities that published events to lambda
 - Triggers AWS Lambda functions
 - AWS services or developer-created applications
 - Can be:
 - Push model – an event source directly invokes lambda function when an event occurs
 - Synchronous
 - Asynchronous
 - Pull model – lambda polls the event stream or queue and invokes the lambda function when it detects an event

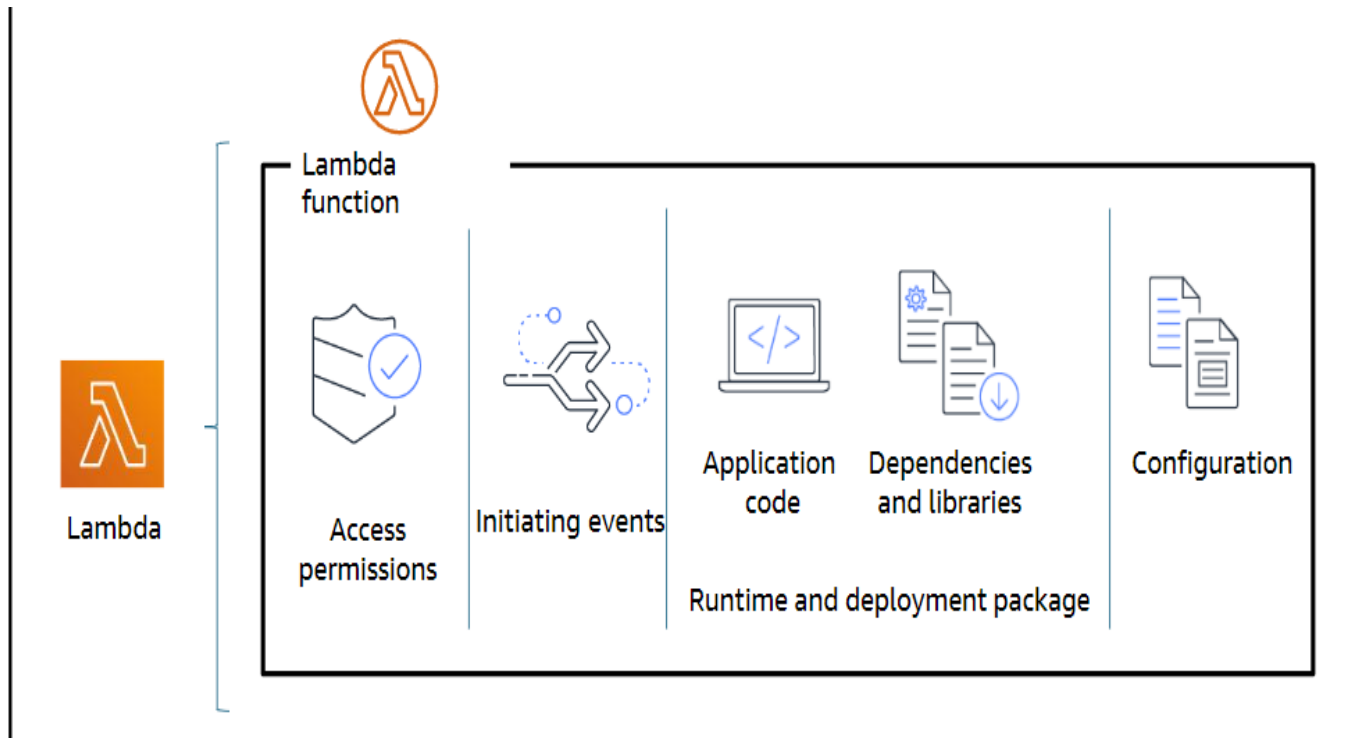
Basic components of AWS Lambda



Basic components of AWS Lambda

- Lambda Functions
 - A custom code that processes events
 - The lambda system manages and runs functions
 - When activated - Triggers

Basic components of AWS Lambda



Basic components of AWS Lambda

- Steps to create lambda functions:
 1. Write the code
 2. Give permissions
 - Event sources need permission to trigger lambda functions
 - Lambda functions need permission to interact with other AWS services
 - Handled through IAM

Basic components of AWS Lambda

Example resource (function) policy

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "lambda-fd269e28-988b-4d2b-96ae-eabcd7dc399c",
      "Effect": "Allow",
      "Principal": {
        "Service": "s3.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:function:myFirstFunction",
      "Condition": {
        "ArnLike": {
          "AWS:SourceARN": "arn:aws:s3:::myBucket1"
        }
      }
    }
  ]
}
```


Basic components of AWS Lambda

Execution role example

IAM policy

```
{
  "Effect": "Allow",
  "Action": [
    "logs:CreateLogGroup",
    "logs:CreateLogStream",
    "logs:PutLogEvents"
  ],
  "Resource": "arn:aws:logs:*:*:*"
}
```

Basic components of AWS Lambda

3. Specify the runtime environment - libraries and dependencies
 - E.g. version of Node.js
4. Give function parameter configuration
 - Trigger – event source
 - Specify memory
 - Timeout
 - Concurrency
 - Optional: Environment variables, VPC, tags, etc.

Basic components of AWS Lambda

Performance-related configurations

Configuration	Description
Memory	Set the amount of memory and proportional CPU that is allocated to the function. Lambda allocates CPU power linearly in proportion to the memory that you set. Values from 128 MB–10,240 MB.
Timeout	Choose the maximum amount of time to let a function run before ending it if it has not completed. Values from 1 second up to 15 minutes.
Concurrency	Number of invocations of a function that can run at the same time. By default, you can have 1,000 concurrency invocations per Region for an account. This limit is a soft limit. Per function, you can set a limit to prevent overwhelming a downstream system or to reserve capacity from within the account pool for the function.
Provisioned concurrency	Number of Lambda environments to <i>keep warm</i> . When Lambda creates the temporary environment for your function, it will attempt to reuse it for additional invocations. This approach saves startup time. When Lambda must initialize new environments to run a function, an invocation might get a <i>cold start</i> , which creates startup latency. Use provisioned concurrency to avoid cold starts. Provisioned concurrency is priced separately.
Monitoring and operations	Settings to enable X-Ray (active tracing) and CloudWatch Lambda Insights (collect and aggregate Lambda function runtime performance metrics and logs).



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

38

Basic components of AWS Lambda

Resource-related configurations

Configuration	Description
Triggers	Event sources that invoke a function.
Permissions	The resources that have permissions to invoke the function and the permissions that the function has to interact with other resources.
Destinations	An SNS topic, SQS queue, other Lambda function, or EventBridge event bus. It receives invocation records from a function when it is successful (on success) or when it fails (on failure).
Asynchronous invocation	Settings for number of retry attempts on async invocations (0–2) and how long to keep an event waiting to be invoked (up to 6 hours). Also, configuration of a dead-letter queue for functions that continue to fail.
VPC	Settings to enable your Lambda function to access resources in a custom VPC. A custom VPC defines a private network of resources, such as databases, cache instances, or internal services.
State machines	Step Functions state machines that can invoke your function directly in at least one of the steps.
Database proxies	Settings that are used when connecting to an Amazon Relational Database Service (Amazon RDS) instance. These settings are used to set up a database proxy to manage connection pooling for database connections.
File systems	Settings to connect an Amazon Elastic File System (Amazon EFS) file system so that the function can access the file system at runtime.



©2023, Amazon Web Services, Inc. or its affiliates. All rights reserved.

39

Basic components of AWS Lambda

Code-related configurations

Configuration	Description
Runtime	Runtime that the function will use or language that the code will be written in. Choose from supported runtimes that are listed in the AWS Lambda Developer Guide or use a custom runtime.
Environment variables	Key-value pairs that are accessible from your function code. Environment variables are useful to store configuration settings without the need to change function code.
Tags	Labels that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Code signing	Option to ensure that code has been signed by an approved source and has not been altered.

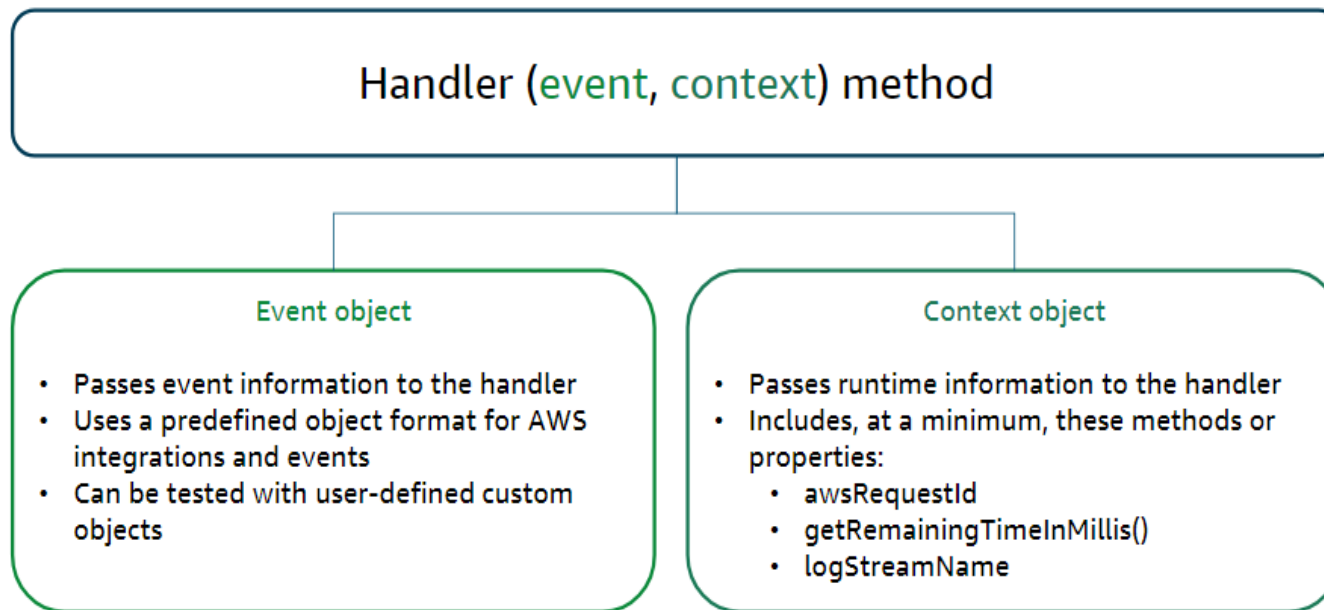
Basic components of AWS Lambda

- **Concurrency**
 - The number of lambda function invocations running at a time
 - Factors that influence concurrency:
 - Reserved concurrency in a function
 - Regional quota
 - Burst quota
 - Request rate and function duration
 - Event source

Basic components of AWS Lambda

- Function handler
 - The entry point the lambda calls to start running the lambda function
 - Defined in the handler file
 - Takes two objects
 - The event object – contains information about the event that triggers the lambda function
 - AWS built-in
 - Custom built
 - The context object – provides runtime information
 - AWS generated
 - Depends on the programming language used

Basic components of AWS Lambda



Basic components of AWS Lambda



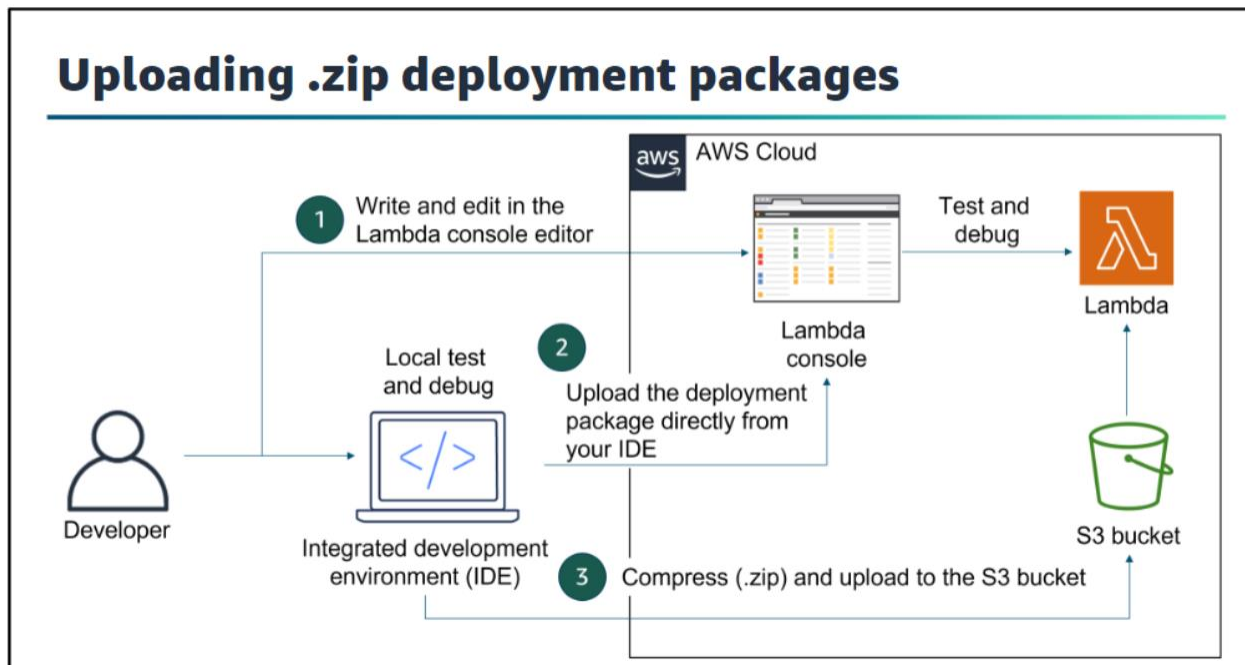
Basic components of AWS Lambda

```
~
3  import com.amazonaws.services.lambda.runtime.Context;
4  import com.amazonaws.services.lambda.runtime.LambdaLogger;
5  import com.amazonaws.services.lambda.runtime.RequestHandler;
6
7  // Handler value: example.HandlerInteger
8  ✓ public class HandlerIntegerJava17 implements RequestHandler<IntegerRecord, Integer>{
9
10     @Override
11     /*
12      * Takes in an InputRecord, which contains two integers and a String.
13      * Logs the String, then returns the sum of the two Integers.
14      */
15     ✓ public Integer handleRequest(IntegerRecord event, Context context)
16     {
17         LambdaLogger logger = context.getLogger();
18         logger.log("String found: " + event.message());
19         return event.x() + event.y();
20     }
21 }
22
23 record IntegerRecord(int x, int y, String message) {
24 }
```

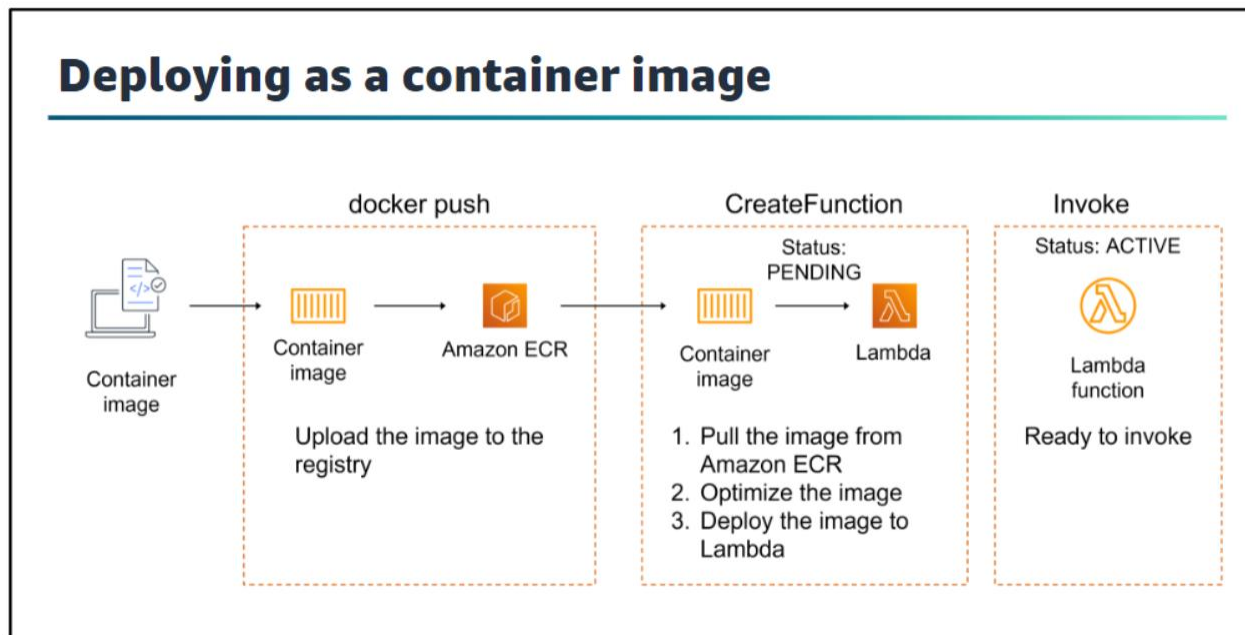
Basic components of AWS Lambda

- Lambda function Deployment
 - Using Command line tool, AWS SDK, or lambda API
 - Lambda function deployment package
 - A ZIP archive containing code and dependencies
 - A container image in an OCI-compatible format
 - A layer
 - A .zip archive that contains libraries or dependencies
 - Usually shared among multiple functions
 - Makes deployment packages small
 - A function can use up to 5 layers

Basic components of AWS Lambda



Basic components of AWS Lambda



Basic components of AWS Lambda

- Best practices
 - Treat functions as stateless
 - Include only what is needed
 - Size and complexity of dependencies
 - Separate the core business code from the function handler code
 - Avoid recursive calls
 - Use environmental variables

Basic components of AWS Lambda

- Pricing model
 - Pay-as-you-go
 - Based on the amount of time a function runs and the memory it consumes

References

- <https://aws.amazon.com/lambda/>