Serverless Architecture

A serverless architecture is a way to build and run applications and services without having to manage infrastructure. Your application still runs on servers, but all the server management is done by AWS.

Why do we need serverless architectures?

By using a serverless architecture, your developers can focus on their core product instead of worrying about managing and operating servers or runtimes, either in the cloud or on-premises. This reduced overhead lets developers reclaim time and energy that can be spent on developing great products which scale and that are reliable.

Benefits:

* Move from idea to market, faster
* Lower your costs
* Adapt at scale
* Build better applications, easier

AWS Lambda is one such serverless compute service

Lambda

Lambda is an ideal compute service for application scenarios that need to scale up rapidly, and scale down to zero when not in demand.

Lambda runs your code on a high-availability compute infrastructure and performs all of the administration of the compute resources, including server and operating system maintenance, and automatic scaling, and logging.

why do we need Lambda?

EC2-virtual sevices are limited by cpu and ram,

they will be continuoulsy running even when they are not in us

scaling is complicated because of intervention to add/remove servers

but using lambda-

we run virtual function, no servers to manage

it will run only on-demand

scaling is automated

Benefits

easy pricing- pay per request and compute time

functions get invoked only when needed

integrated with many programming languages

easy monitoring through aws cloudwatch

increasing RAM will also improve cpu and network

Use case:

Let say, we need to store an image in S3

S3 triggers AWS lambda. lambda function creates a thumbnail of the image.

and pushes back the thumbnail in to the s3 bucket

and pushes some metadata(inclues image size, image name, creation date, etc) into dynamodb

this thumbnail creation sclaes it really well

its completely serverless as we don't provision any servers here

def lambda\_handler(event, context):

#print("Received event: " + json.dumps(event, indent=2))

print("value1 = " + event['key1'])

print("value2 = " + event['key2'])

print("value3 = " + event['key3'])

return event['key2'] # Echo back the first key value

#raise Exception('Something went wrong')

Test Event Name

DemoEvent

Response

"value2"

Function Logs

START RequestId: 386dbca2-24bf-46aa-91c9-67d46ba9e558 Version: $LATEST

value1 = value1

value2 = value2

value3 = value3

END RequestId: 386dbca2-24bf-46aa-91c9-67d46ba9e558

REPORT RequestId: 386dbca2-24bf-46aa-91c9-67d46ba9e558 Duration: 16.45 ms Billed Duration: 17 ms Memory Size: 128 MB Max Memory Used: 31 MB

Request ID

386dbca2-24bf-46aa-91c9-67d46ba9e558