|  |
| --- |
| **Name:** WASIMA QAYYUMUDDIN SHAIKH  **RollNo:**6220071  **Class:** T.E.I.T  **Sem:** V  **Subject:** ADVACE DEVOPS LAB **(Addevops**)  **EXPERIMENT NO: 05**  **1)WHAT IS AWS LAMBDA?**  AWS Lambda Reviews | The Iron.io Blog  AWS Lambda Is An Event-Driven, Serverless Computing Platform Provided By Amazon As A Part Of Amazon Web Services. It Is A Computing Service That Runs Code In Response To Events And Automatically Manages The Computing Resources Required By That Code. It Was Introduced In November 2014.  The Concept Of “Serverless” Computing Refers To Not Needing To Maintain Your Own Servers To Run These Functions. Aws Lambda Is A Fully Managed Service That Takes Care Of All The Infrastructure For You. And So “Serverless” Doesn’t Mean That There Are No Servers Involved: It Just Means That The Servers, The Operating Systems, The Network Layer And The Rest Of The Infrastructure Have Already Been Taken Care Of, So That You Can Focus On Writing Application Code.  **2)WHAT IS SERVERLESS COMPUTING?**  Serverless Computing. The wave of digital transformation has… | by  SuccessiveTech | SuccessiveTech | Medium  The Wave Of Digital Transformation Has Left Companies Racing To Improve End-User Experiences, Part Of Which Involves Improving Developer Operations To Achieve Greater Business Outcomes. The Introduction Of Cloud Computing Changed The Way Companies ‒ Across Industries ‒ Think About IT Infrastructure, Eventually Prompting The Rise Of Containers, Like Docker And Kubernetes, That Enabled Developers To Break Down Monolithic Programs To Run More Efficiently.  Serverless Computing, Or Functions-As-A-Service (Faas), Isn’t New, But It’s Becoming Clear That It’s The Next Evolutionary Step Towards Leveraging The Full Potential Of Cloud Technology And Achieving Greater Organizational Agility.  Serverless Computing Is A [Cloud Computing](https://en.wikipedia.org/wiki/Cloud_computing) [Execution Model](https://en.wikipedia.org/wiki/Execution_model) In Which The Cloud Provider Allocates Machine Resources On Demand, Taking Care Of The [Servers](https://en.wikipedia.org/wiki/Server_(computing)) On Behalf Of Their Customers.  Serverless Computing Does Not Hold Resources In Volatile Memory; Computing Is Rather Done In Short Bursts With The Results Persisted To Storage.  When An App Is Not In Use, There Are No Computing Resources Allocated To The App. Pricing Is Based On The Actual Amount Of Resources Consumed By An Application.  It Can Be A Form Of [Utility Computing.](https://en.wikipedia.org/wiki/Utility_computing) "Serverless" Is A [Misnomer](https://en.wikipedia.org/wiki/Misnomer) In The Sense That Servers Are Still Used By Cloud Service Providers To Execute Code For Developers.  However, Developers Of Serverless Applications Are Not Concerned With [Capacity Planning,](https://en.wikipedia.org/wiki/Capacity_planning) Configuration, Management, Maintenance,Fault Toleranc[e,](https://en.wikipedia.org/wiki/Fault_tolerance) Or Scaling Of Containers, [Vms,](https://en.wikipedia.org/wiki/Virtual_machine) Or Physical Servers.  Serverless Computing Can Simplify The Process Of [Deploying Code](https://en.wikipedia.org/wiki/Software_deployment) Into Production. Serverless Code Can Be Used In Conjunction With Code Deployed In Traditional Styles, Such As [Microservices](https://en.wikipedia.org/wiki/Microservices) Or [Monoliths.](https://en.wikipedia.org/wiki/Monolithic_application)  Alternatively, Applications Can Be Written To Be Purely Serverless And Use No Provisioned Servers At All. This Should Not Be Confused With Computing Or Networking Models That Do Not Require An Actual Server To Function, Such As [Peer-To-Peer](https://en.wikipedia.org/wiki/Peer-to-peer).  Serverless computing is a method of providing backend services on an as-used basis. Servers are still used, but a company that gets backend services from a serverless vendor is charged based on usage, not a fixed amount of bandwidth or number of servers.  The State of AWS Lambda Supported Languages &amp; Runtimes (Updated November  2019) **3)WHAT LANGUAGES DOES AWS LAMBDA SUPPORT?**  In the last few years a lot has changed with AWS Lambda supported languages and runtimes. With all of these changes there are some older runtimes reaching the end of their supported life and several new runtimes and new features to think about taking advantage of.  By a supported language  like Node.js, Python, or Java. By a runtime  mean a specific version of that language like Python 3.7 or Java 11.  As Of Now, Aws Lambda Doesn’t Support All Programming Languages, But It Does Support A Number Of The Most Popular Languages And Runtimes. This Is The Full List Of What’s Supported:   * Node.Js 8.10 * Node.Js 10.X (Normally The Latest Lts Version From The 10.X Series) * Node.Js 12.X (Normally The Latest Lts Version From The 12.X Series) * Python 2.7 * Python 3.6 * Python 3.7 * Python 3.8 * Ruby 2.5 * Java 8 * This Includes Jvm-Based Languages That Can Run On Java 8’s Jvm — The Latest Clojure 1.10 And Scala 2.12 Both Run On Java 8 So Can Be Used With Aws Lambda:- * Java 11 * Go 1.X (Latest Release) * C# — .Net Core 1.0 * C# — .Net Core 2.1 * Powershell Core 6.0   All These Runtimes Are Maintained By Aws And Are Provided In An Amazon Linux Or Amazon Linux 2 Environment. For Each Of The Supported Languages, Aws Provides An Sdk That Makes It Easier For You To Write Your Lambda Functions And Integrate Them With Other Aws Services.  A Few Additional Runtimes Are Still In The Pre-Release Stage. These Runtimes Are Being Developed As A Part Of Aws Labs And Are Not Mentioned In The Official Documentation:   * [Rust 1.31](https://github.com/awslabs/aws-lambda-rust-runtime) * [C++](https://github.com/awslabs/aws-lambda-cpp)   The C++ Runtime Also Serves As An Example For Creating Custom Runtimes For Aws Lambda. [See The Aws Docs](https://docs.aws.amazon.com/lambda/latest/dg/runtimes-custom.html) For The Details Of How To Create A Custom Runtime If Your Language Isn’t Supported By Default.  **4)WHAT IS AWS DYNAMODB TABLE?**  Simba DynamoDB ODBC and JDBC Drivers | Simba | Magnitude  Since the Amazon DynamoDB data warehouse is schema-less, Simba DynamoDB connectors with SQL Connector allow users to define schemas on the fly, providing a powerful SQL window into DynamoDB dataAmazon Dynamodb Is A Fully Managed NoSql Database Service That Provides Fast And Predictable Performance With Seamless Scalability. Dynamodb Lets You Offload The Administrative Burdens Of Operating And Scaling A Distributed Database So That You Don't Have To Worry About Hardware Provisioning, Setup And Configuration, Replication, Software Patching, Or Cluster Scaling. Dynamodb Also Offers Encryption At Rest, Which Eliminates The Operational Burden And Complexity Involved In Protecting Sensitive Data.  With Dynamodb, You Can Create Database Tables That Can Store And Retrieve Any Amount Of Data And Serve Any Level Of Request Traffic. You Can Scale Up Or Scale Down Your Tables' Throughput Capacity Without Downtime Or Performance Degradation. You Can Use The Aws Management Console To Monitor Resource Utilization And Performance Metrics.  Dynamodb Provides On-Demand Backup Capability. It Allows You To Create Full Backups Of Your Tables For Long-Term Retention And Archival For Regulatory Compliance Needs. For More Information, See [On-Demand Backup And Restore For Dynamodb.](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/BackupRestore.html)  You Can Create On-Demand Backups And Enable Point-In-Time Recovery For Your Amazon Dynamodb Tables. Point-In-Time Recovery Helps Protect Your Tables From Accidental Write Or Delete Operations. With Point-In-Time Recovery, You Can Restore A Table To Any Point In Time During The Last 35 Days. For More Information, See [Point-In-Time Recovery: How It Works.](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/PointInTimeRecovery_Howitworks.html)  Dynamodb Allows You To Delete Expired Items From Tables Automatically To Help You Reduce Storage Usage And The Cost Of Storing Data That Is No Longer Relevant. For More Information, See [Expiring Items By Using Dynamodb Time To Live (Ttl).](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/TTL.html)  Similar To Other Database Systems, Amazon Dynamodb Stores Data In Tables. You Can Manage Your Tables Using A Few Basic Operations.   * [Creating A Table](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.CreateTable) * [Describing A Table](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.DescribeTable) * [Updating A Table](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.UpdateTable) * [Deleting A Table](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.DeleteTable) * [Listing Table Names](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.ListTables) * [Describing Provisioned Throughput Quotas](https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithTables.Basics.html#WorkingWithTables.Basics.DescribeLimits)   Dynamodb Tables Are Schemeless, Except For The Primary Key, So The Items In A Table Can All Have Different Attributes, Sizes, And Data Types.  The Total Size Of An Item Is The Sum Of The Lengths Of Its Attribute Names And Values. You Can Use The Following Guidelines To Estimate Attribute Sizes:   * Strings Are Unicode With Utf-8 Binary Encoding. The Size Of A String Is (Length Of Attribute Name) + (Number Of Utf-8-Encoded Bytes). * Numbers Are Variable Length, With Up To 38 Significant Digits. Leading And Trailing Zeroes Are Trimmed. The Size Of A Number Is Approximately (Length Of Attribute Name) + (1 Byte Per Two Significant Digits) + (1 Byte). * A Binary Value Must Be Encoded In Base64 Format Before It Can Be Sent To Dynamodb, But The Value's Raw Byte Length Is Used For Calculating Size. The Size Of A Binary Attribute Is (Length Of Attribute Name) + (Number Of Raw Bytes). * The Size Of A Null Attribute Or A Boolean Attribute Is (Length Of Attribute Name) + (1 Byte). * An Attribute Of Type List Or Map Requires 3 Bytes Of Overhead, Regardless Of Its Contents. The Size Of A List Or Map Is (Length Of Attribute Name) + Sum (Size Of Nested Elements) + (3 Bytes) . The Size Of An Empty List Or Map Is (Length Of Attribute Name) + (3 Bytes).   **5)EXPLAIN AWS IAM SERVICE?**  AWS Identity And Access Management (IAM) Enables You To Manage Access To AWS Services And Resources Securely. Using IAM, You Can Create And Manage AWS Users And Groups, And Use Permissions To Allow And Deny Their Access To AWS Resources.  [Cloud Security](https://www.simplilearn.com/things-you-must-know-about-cyber-security-in-the-cloud-article) Is The Highest Priority In Aws. When You Host Your Environment In The Cloud, You Can Be Assured That It’s Hosted In A Data Centre Or In A Network Architecture That’s Built To Meet The Requirements Of The Most Security-Sensitive Organization.  Additionally, This High Level Of Security Is Available On A Pay-As-You-Go Basis, Meaning There Is Really No Upfront Cost, And The Cost For Using The Service Is A Lot Cheaper Compared To An On-Premises Environment.  There Are Many Types Of Security Services Available But Some Of Them Are Widely Used By Aws, Such As:   * Iam * Key Management System (Kms) * Cognito * Web Access Firewall (Waf)   Iam Enables You To Manage Access To Aws Services And Resources In A Very Secure Manner.  With Iam You Can Create Groups And Allow Those Users Or Groups To Access.  **WHY IAM?**  Before Aws Or Iam, Passwords Were Often Shared In Corporate Environments In A Very Insecure Manner: Over The Phone Or Through Email.  Often Only One Admin Password Existed, Which Was Commonly Stored In A Set Location, Or There Was Only One Person Who Could Reset It, And You Needed To Call The Person To Ask For The Admin Password Over The Phone. That Was Not Secure At All, Because Anybody Could Walk By And Eavesdrop And Then Walk Away With The Password And Access To Your System And Information.  Today We Have A More Secure Communication Tool: A Third-Party Application Called Slack, Which Is Hosted On Aws. It Helps People To Share A Document Through The Application So That Eavesdropping Is Eliminated.  **AWS IAM ALLOWS YOU TO:**   * [**Manage IAM Users A**](https://aws.amazon.com/iam/features/manage-users/)**nd** [**Their Access**](https://aws.amazon.com/iam/features/managing-user-credentials/) **–**   You Can Create Users In Iam, Assign Them Individual Security Credentials (In Other Words, Access Keys, Passwords, And [Multi-Factor Authentication](https://aws.amazon.com/iam/features/mfa/) Devices), Or Request Temporary Security Credentials To Provide Users Access To Aws Services And Resources. You Can Manage Permissions In Order To Control Which Operations A User Can Perform.   * [**Manage Iam Roles**](https://aws.amazon.com/iam/features/manage-roles/) **And** [**Their Permissions**](https://aws.amazon.com/iam/features/manage-permissions/) **–**   You Can Create Roles In Iam And Manage Permissions To Control Which Operations Can Be Performed By The Entity, Or Aws Service, That Assumes The Role. You Can Also Define Which Entity Is Allowed To Assume The Role. In Addition, You Can Use [Service-Linked Roles](https://docs.aws.amazon.com/console/iam/service-linked-role) To Delegate Permissions To Aws Services That Create And Manage Aws Resources On Your Behalf.   * [**Manage Federated Users**](https://aws.amazon.com/identity/federation/) **And** [**Their Permissions**](https://aws.amazon.com/iam/features/manage-permissions/) **–**   You Can Enable Identity Federation To Allow Existing Identities (Users, Groups, And Roles) In Your Enterprise    **6)To Understand Aws Lambda, Create Your First Lambda Functions Using Python / Java / Nodejs. Create Aws Lambda Function And Configure A Trigger For Amazon Simple Storage Service(Amazon S3). The Trigger Invokes Your Lambda Function Every Time That You Add An Object To Your Amazon S3 Bucket. Allow Aws Lambda To Access Amazon Dynamodb Table .Create IAM Role That Allows Full Access To Dynamodb Table [Terminate the resources after performing the**  **practical].**   1. Creating A Role For Full Access Of Aws Dynamodb. Services -> Iam ->Role. 2. Then Click On Create Role      1. After Clicking On Create Role Choose Lambda Then Click Next Permissions.      1. Then In Search Bar Search (AmazonDynamoDBfullaccess) And Then Select It Then Next.      1. Then In Tag Section Click On Next      1. After Than Type Your ‘Role Name’ → Create Role        1. Click On Service→ Lambda 2. Then Click On Create Function      1. Then Give Name To Your Function.Then In Runtime Section Select ‘Python 36’ Then Change The Default Execution Role To Use An Existing Role →Choose The Role Which Was Created Earlier →Create Function.      1. In Code Source Section→ Lambda Function → Paste The Below Code.     import boto3  from uuid import uuid4  def lambda\_handler(event, context):  s3 = boto3.client("s3")  dynamodb = boto3.resource('dynamodb')  for record in event['Records']:  bucket\_name = record['s3']['bucket']['name']  object\_key = record['s3']['object']['key']  size = record['s3']['object'].get('size', -1)  event\_name = record ['eventName']  event\_time = record['eventTime']  dynamoTable = dynamodb.Table('newtable')  dynamoTable.put\_item(  Item={'unique': str(uuid4()), 'Bucket': bucket\_name, 'Object': object\_key,'Size': size, 'Event': event\_name, 'EventTime': event\_time})    After Deploying:-     1. Then Service→S3→ Create Bucket→ Name→ Create Bucket          1. Then Service→ Lambda→ Function→ Add Trigger      1. Select S3 Bucket Which Was Previously Created → Add        1. After That Service→ Dynamodb → Create Table   (Table Name = Newtable; Primary Key = Unique)→ Create         1. Then Service→ S3→ Bucket→ Upload→ Add File(Select Any File)→ Upload→ Close          1. And Now As You Added The File Now Go To Your Dynamodb And Their You Will See Your File All Detail Like It Size ,Name,Added Date Etc     **FOR TERMINATING THE RESOURCES:-**  **1.Amazon Dynamo Db:-**   * Now Delete The Table. * Click On Delete And Type Delete.     **2.Amazon S3 Bucket.**   * Now Go To Services-> S3. * We Need To Delete Bucket. * But Before Deleting The Bucket * We Need To MakeThe Bucket Empty * For That Click On The Bucket * Select All The Files Which Are Uploaded * And Delete The Files By Typing “Permenantely Delete” * And Then Click On “Delete Objects”. * Now Delete The Bucket. * Click On Delete It Will Ask For Name Of The Bucket . * Type The Name And Click On Delete Bucket. * Now Go To Services Lambda And Delete Lambda Function By Clicking On Actions->Delete |