**AWS Automation with Python Boto3 and Lambda Functions**

**[PART - 1] Introduction**

**Aim:**

Learn how to automate common AWS tasks using Boto3 and Lambda Functions.

**Objectives:**

1. Cover the core concepts of Boto3 and Lambda.
2. Understand Boto3 and Lambda concepts with real-time scenarios.
3. Run Boto3 scripts on your local machine and trigger Lambda functions.
4. Gain the knowledge to apply different concepts of Boto3 and Lambda for various AWS Services by the end of this course.

**Pre-requisites:**

**What do you need for this course?**

1. AWS Account (Free tier account recommended).
2. Basic knowledge of AWS Services and Python (Not mandatory).
3. Knowledge of any Python IDE (Not mandatory).

**Note:** All videos will be uploaded to the “AWS Automation with Python Boto3” playlist.

**[PART - 2] Introduction to Boto3**

**What is Boto3?**

* Boto3 is the Python SDK/Library/Module/API for AWS.
* Boto3 allows you to directly create, update, and delete AWS services from Python scripts.
* Boto3 is built on top of the botocore module.

**Installing Boto3:**

**For Python-2.x:**

* **pip install boto3**

**For Python-3.x:**

* **pip3 install boto3**

**Installation Steps:**

**Install Python and Boto3 on Windows:**

1. Download Python 3.7.4 from [www.python.org](http://www.python.org/).
2. Set paths for Python and pip3.
3. Install Boto3 using **pip3 install boto3**.

**Install Python and Boto3 on Linux:**

1. Install dependencies:
   * **yum install gcc openssl-devel bzip2-devel libffi-devel**
2. Download and install Python 3.7.4:
   * **cd /usr/src**
   * **wget https://www.python.org/ftp/python/3.7.4/Python-3.7.4.tgz**
   * **tar xzf Python-3.7.4.tgz**
   * **cd Python-3.7.4**
   * **./configure --enable-optimizations**
   * **make altinstall**
3. Verify the installation:
   * **cd /usr/local/bin/**
   * **./python3.7 --version**
   * **./pip3.7 --version**
4. Create symbolic links:
   * **ln -s /usr/local/bin/python3.7 /bin/python3**
   * **ln -s /usr/local/bin/pip3.7 /bin/pip3**
5. Install Boto3:
   * **pip3 install boto3**

**[PART 3] Boto3 Environment Setup**

**Configuring AWS CLI:**

**What is AWS CLI?** The AWS Command Line Interface (AWS CLI) is a unified tool to manage AWS services. It allows you to control multiple AWS services from the command line and automate them through scripts.

**Download AWS CLI:**

* [AWS CLI Installation Guide](https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html)

**Configure Credentials:**

1. Login to AWS Management Console and create a new user with programmatic access and provide **AdministratorAccess**.
2. Configure root/IAM user access-keys/credentials using:
   * **aws configure** (Creates a DEFAULT profile).

**Using Different Profiles:**

* **aws configure --profile dev**
* **aws configure --profile qa**
* **aws configure --profile prod**

**First Automation Script:**

**List all IAM Users in your Account:**

1. Perform the task manually on AWS Management Console.
2. Navigate to the IAM Console:
   * Users
   * Groups
   * Roles
   * Policies

**[PART 4] Concepts of Boto3**

**Key Concepts:**

* **Session:**
  + Similar to the AWS Management Console; stores configuration information and allows the creation of Service, Clients, and Resources.
* **Resource and Client:**
  + Create AWS Service consoles (e.g., IAM Console, EC2 Console).
  + Created from a Session object.
  + Specify region name after the Profile name.

**Resource vs. Client:**

* **Resource:**
  + Higher Level Object-oriented service access.
  + Available for a few AWS Services (**['cloudformation', 'cloudwatch', 'dynamodb', 'ec2', 'glacier', 'iam', 'opsworks', 's3', 'sns', 'sqs']**).
  + Simple operations using (.) notation.
* **Client:**
  + Low-Level Service Access.
  + Output/response is a dictionary, requiring more effort to implement Boto3 scripts.
  + Supports all operations seen in the AWS Management Console.
  + Can switch to Client from Resource using the “Meta” concept.

**Choosing Between Resource and Client:**

* Use Resource for higher-level object-oriented service access.
* Use Client for low-level service access when more detailed control is needed.

**[PART 5] Boto3 Script with Boto3 Documentation**

**Examples:**

1. **List all IAM users in AWS Account using client objects.**
2. **List all running EC2 Instances in your AWS Account using client objects.**
3. **List all IAM users in AWS Account using resource objects.**
4. **List all running EC2 Instances in your AWS Account using resource objects.**

**Supported Resource Objects:**

* **['cloudformation', 'cloudwatch', 'dynamodb', 'ec2', 'glacier', 'iam', 'opsworks', 's3', 'sns', 'sqs']**