AWS Media Convert Using AWS SDK For Python(Boto3) and AWS Lambda

1.Creating Role:

I)Lambda:

i) Go to IAM Console and navigate to Role menu in the console.

ii) Create a Role and select AWS Service as Trusted Entity and select Lambda as Use Case.

iii) Select “AWSLambdaBasicExecutionRole” and “AmazonS3FullAccess” in the add permission menu.

iv) Name the Role as “VODLambdaRole”.

v)Now the role is created, select the “VODLambdaRole” role and navigate to permission.

vi)Click Edit and Select Inline Policy and copy the below code in the json format.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*",
      "Effect": "Allow",
      "Sid": "Logging"
    },
    {
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "<ARN for vod-MediaConvertRole>"
      ],
      "Effect": "Allow",
      "Sid": "PassRole"
    },
    {
      "Action": [
        "mediaconvert:*"
      ],
      "Resource": [
        "*"
      ],
      "Effect": "Allow",
      "Sid": "MediaConvertService"
    }
  ]
}
```

II)MediaConvert:

i) Go to IAM Console and navigate to Role menu in the console.

ii) Create a Role and select AWS Service as Trusted Entity and select MediaConvert as Use Case.

iii)For MediaConvert, all the permissions are predefined so we can go to name creation and create role as mentioned above.

2.Create a Lambda Function to convert Videos:

i) Navigate to AWS Lambda and choose create Function.

ii) Choose “Author from scratch” and name the function as “VODLambdaConvert” and select python-3.8 as runtime environment.

iii)Choose already existing role in role selection and choose the created lambda role.

3. Lambda Function:

i) In the Lambda Main Page, Select Configuration tab below and select Code Entry from, Choose Upload a file from Amazon S3 and paste this url.

URL : Lambda Convert Code

ii) Rename the Handler to “convert.handler”

iii) Next navigate to Environmental Variable in Configuration Tab and these value that is mentioned below,

- DestinationBucket = vod-lastname (or whatever you named your bucket in module 1)
- MediaConvertRole = arn:aws:iam::ACCOUNT NUMBER:role/vod-MediaConvertRole
- Application = VOD

4. Test the Code and Set Triggers:

i) Now navigate to Test Tab and select create new test case event and give an event name. Copy the below code and paste it there.

```
{
 "Records": [
   {
     "eventVersion": "2.0",
     "eventTime": "2017-08-08T00:19:56.995Z",
     "requestParameters": {
       "sourceIPAddress": "54.240.197.233"
     },
     "s3": {
       "configurationId": "90bf2f16-1bdf-4de8-bc24-b4bb5cffd5b2",
       "object": {
         "eTag": "2fb17542d1a80a7cf3f7643da90cc6f4-18",
         "key": "vodconsole/TRAILER.mp4",
         "sequencer": "005989030743D59111",
         "size": 143005084
       },
       "bucket": {
         "ownerIdentity": {
           "principalId": ""
         },
         "name": "rodeolabz-us-west-2",
         "arn": "arn:aws:s3:::rodeolabz-us-west-2"
       },
       "s3SchemaVersion": "1.0"
     },
     "responseElements": {
       "x-amz-id-2": "K5eJLBzGn/9NDdPu6u3c9NcwGKNklZyY5ArO9QmGa/t6VH2HfUHHhPuwz2zH1Lz4",
       "x-amz-request-id": "E68D073BC46031E2"
     },
     "awsRegion": "us-west-2",
     "eventName": "ObjectCreated:CompleteMultipartUpload",
     "userIdentity": {
       "principalId": ""
     },
     "eventSource": "aws:s3"
   }
 ]
}
```

ii) Select Create and run the test code. You can check if the code properly if the output is same as mentioned below.

```
{
"body": "{}",
"headers": {
  "Access-Control-Allow-Origin": "*",
  "Content-Type": "application/json"
},
"statusCode": 200
}
```

iii) Select Add triggers in the main screen and choose S3 in the drop-down menu.

iv) Select the input bucket and set event type as “PUT” and save the settings.

v) Now upload a video to the input bucket and check the Media Convert jobs to see if the Lambda Function Triggers.

vi) Check Output Bucket to see if the video is converted and working.