| **TASK**  **Aug 9, 2024** **Aim:-** To convert doc file to pdf file using the below AWS services. **Prerequisites:-** SQS  SNS(Optional)  Lambda  S3 Bucket  EC2 → to show UI **Diagram:-**  **Working:-****Create S3 Buckets:-****Bucket-1:-**            **Bucket-2:-**         Both the S3 Buckets are created.   **Create IAM Role:-**   Add Permissions such as follows:-   1. **AmazonS3FullAccess** 2. **AWSLambda\_FullAccess** 3. **AmazonSQSFullAccess**         Provide the role name as “**lambda\_role**”.      Role created Successfully.    Update     **Create SQS:-**           Setting up policy for SQS:-      Click on “**Add Condition**” and then click on “**Add Statement**”:-      Now, Click on **Generate Policy** and copy the policy to SQS;-  {  "Id": "Policy1723243592452",  "Version": "2012-10-17",  "Statement": [  {  "Sid": "Stmt1723243539035",  "Action": "sqs:\*",  "Effect": "Allow",  "Resource": "arn:aws:sqs:ap-south-1:891377318947:file-conversion-queue",  "Condition": {  "ArnEquals": {  "aws:SourceArn": "arn:aws:s3:::source-bucket-conversion"  }  },  "Principal": "\*"  }  ]  }    Then, change the **Access policy** in **SQS Queue and click on “Save Changes”**:-      Create a Event Notification in S3 bucket for SQS:-           **Create Lambda Function:-**      **Python Code:-** import boto3  import os  import uuid  from botocore.exceptions import NoCredentialsError, PartialCredentialsError  s3 = boto3.client('s3')  sqs = boto3.client('sqs')  ORIGINAL\_BUCKET = 'source-bucket-conversion'  CONVERTED\_BUCKET = 'destination-bucket-conversion'  QUEUE\_URL = 'https://sqs.ap-south-1.amazonaws.com/891377318947/file-conversion-queue'  def lambda\_handler(event, context):  for record in event['Records']:  receipt\_handle = record['receiptHandle']  try:  # Get the object from the S3 bucket  file\_key = record['body']  download\_path = f'/tmp/{uuid.uuid4()}\_{file\_key}'  s3.download\_file(ORIGINAL\_BUCKET, file\_key, download\_path)  # Perform the document conversion (example: converting .docx to .pdf)  converted\_path = convert\_document(download\_path)  # Upload the converted file back to S3  converted\_key = f'converted/{os.path.basename(converted\_path)}'  s3.upload\_file(converted\_path, CONVERTED\_BUCKET, converted\_key)    # Delete the message from the queue  sqs.delete\_message(QueueUrl=QUEUE\_URL, ReceiptHandle=receipt\_handle)  except NoCredentialsError:  print("Error: Credentials not available")  except PartialCredentialsError:  print("Error: Incomplete credentials")  except Exception as e:  print(f"Error processing {file\_key}: {str(e)}")    def convert\_document(input\_path):  # Example conversion logic  output\_path = input\_path.replace('.docx', '.pdf')  # Use a library like python-docx or other to perform actual conversion  # Here we simply rename the file for demonstration  os.rename(input\_path, output\_path)  return output\_path   **Create Trigger:-**       Before:-    After:-   **Conclusion:-** SQS Service to change the doc file to upload and conversion of doc to pdf is done here and uploaded in the s3 bucket |
| --- |