

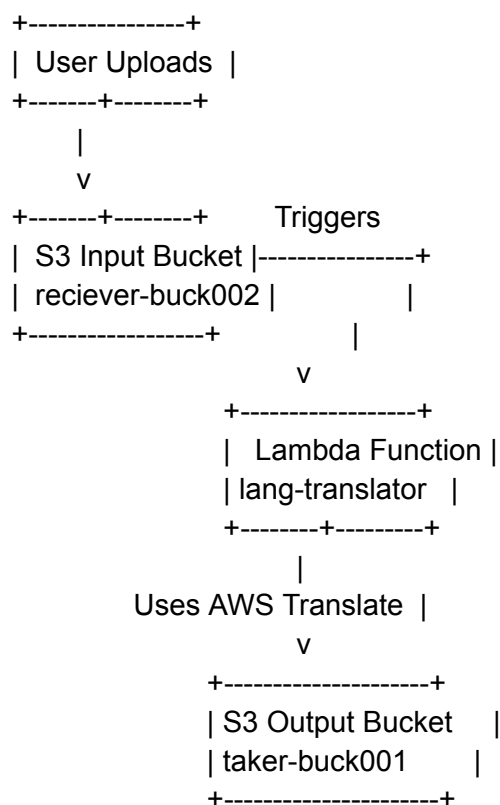
AWS Automated language translation pipeline

Project Overview

This project implements a **serverless language translation pipeline** using AWS. The goal is to **automatically translate uploaded text files** from an input S3 bucket and **store the translated output** in a separate output bucket using Amazon Translate and AWS Lambda.

1.

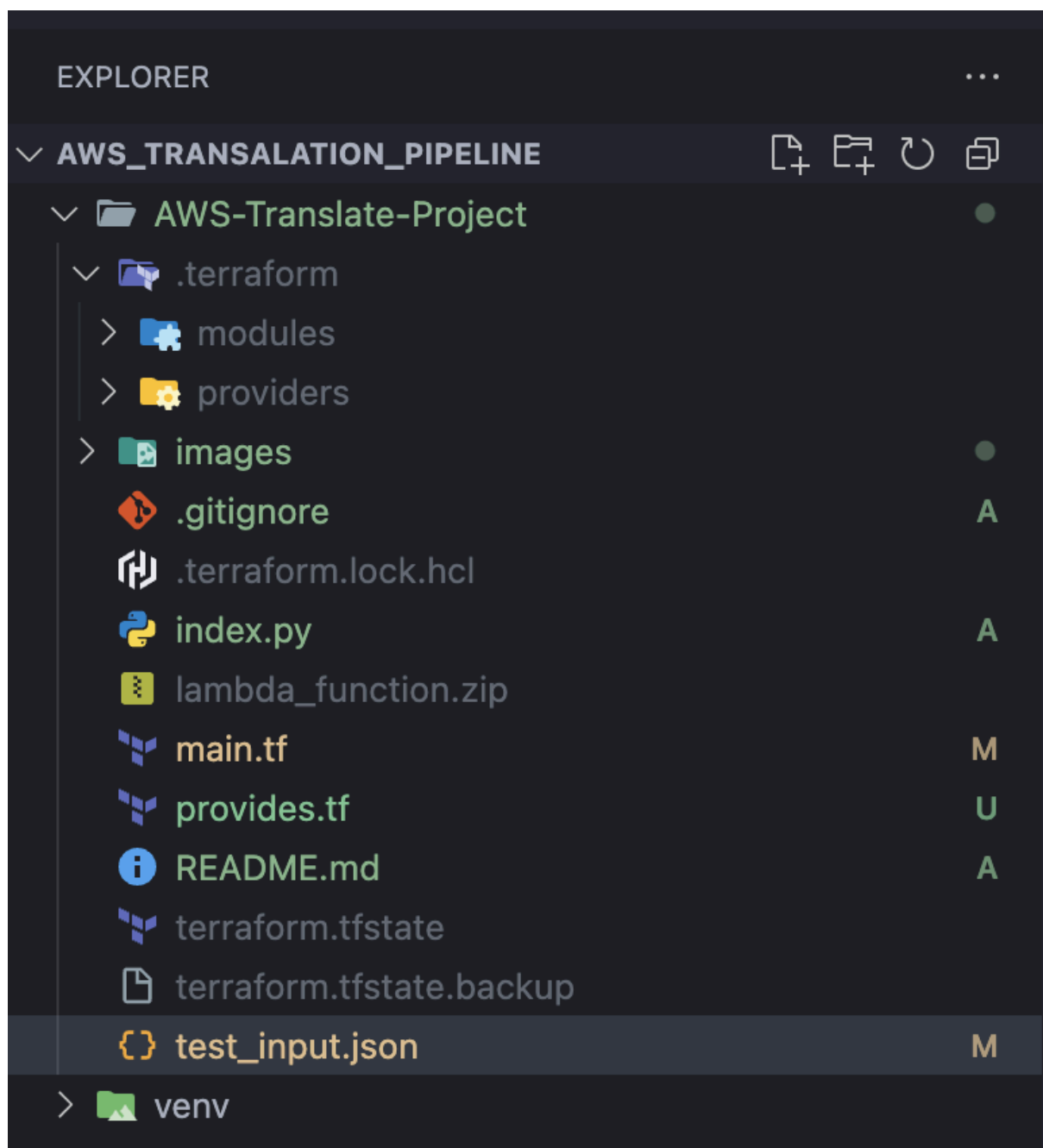
Workflow diagram



Project Structure

```
translation-pipeline/
|
|— terraform/
|   |— main.tf           # All infrastructure defined here
|   |— providers.tf
```

```
├── lambda/
│   ├── index.py          # Lambda function logic
│   └── lambda_function.zip # Zipped package for deployment
├── test/
│   └── test_input.json    # Sample input file for testing
└── README.md             # Project documentation
```



Tools and resources

Component	Service/Tool	Purpose
Compute	AWS Lambda	Serverless backend for translation
File Storage	Amazon s3	Input and Output buckets
IAM	AWS IAM Roles/Policies	Secure access between services
Infrastructure	Terraform	IAC to deploy all resources
Monitoring	Cloudwatch	Lambda execution logging
Translation API	Amazon Translate	Actual Text translation

IAM & Permissions

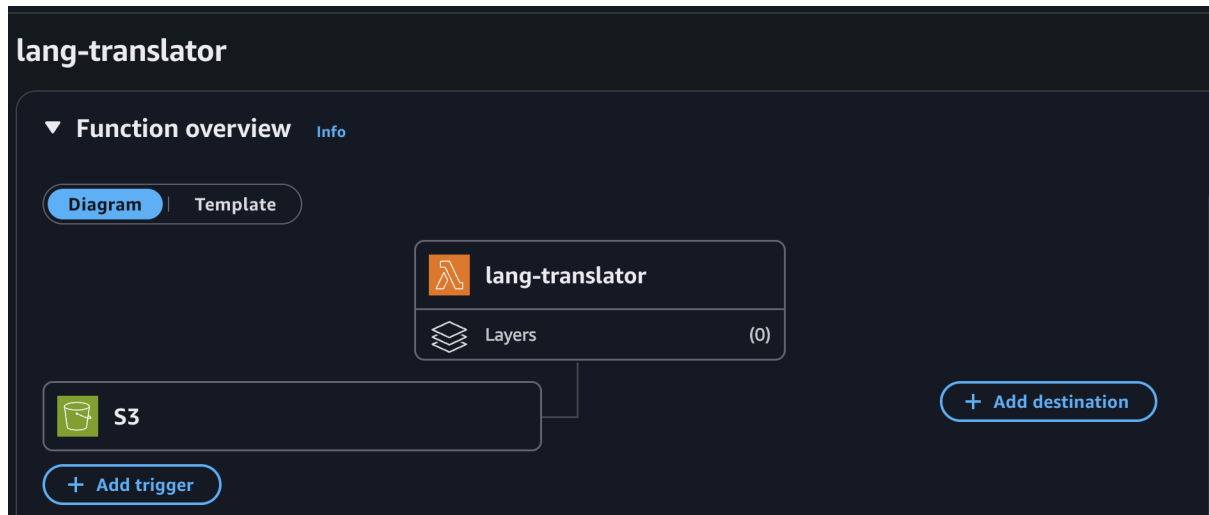
The Lambda function has an IAM role that allows:

- translate: TranslateText
- s3:GetObject, s3:ListBucket on input bucket
- s3:PutObject on output bucket
- logs:* for writing to CloudWatch
-

Event Trigger Flow

2. A JSON file is uploaded to receiver-buck002.
3. An **S3 event notification** triggers the lang-translator Lambda.

4. Lambda reads the file and extracts translation fields.
5. Uses Amazon Translate to translate text.
6. Stores the result in taker-buck001.



Sample Lambda Input Format

```
AWS-Translate-Project > {} test_input.json > ...
1  [
2    {
3      "Text": "Hello, Welcome Back?",
4      "SourceLanguageCode": "en",
5      "TargetLanguageCode": "fr"
6    },
7    {
8      "Text": "I am part of IAM Masters.",
9      "SourceLanguageCode": "en",
10     "TargetLanguageCode": "es"
11   }
12 ]
13
```

Sample Output File

```
test_input-translated.json > No Selection

1 {"translations": [{"original": "Hello, Welcome Back?", "translated": "Bonjour, bon
    retour\u00a0?", "from": "en", "to": "fr"}, {"original": "I am part of IAM
    Masters.", "translated": "Formo parte de IAM Masters.", "from": "en", "to":
    "es"}]}

2
```

How to Deploy

1. Install Terraform

```
# -----
# 1. S3 BUCKETS
# -----

resource "aws_s3_bucket" "input_bucket" {
  bucket = "reciever-buck002"
  force_destroy = true
}

resource "aws_s3_bucket" "output_bucket" {
  bucket = "taker-buck001"
  force_destroy = true
}

# Lifecycle for input bucket
resource "aws_s3_bucket_lifecycle_configuration" "input_lifecycle" {
  bucket = aws_s3_bucket.input_bucket.id

  rule {
    id      = "expire-inputs"
    status  = "Enabled"

    filter {
      prefix = ""
    }

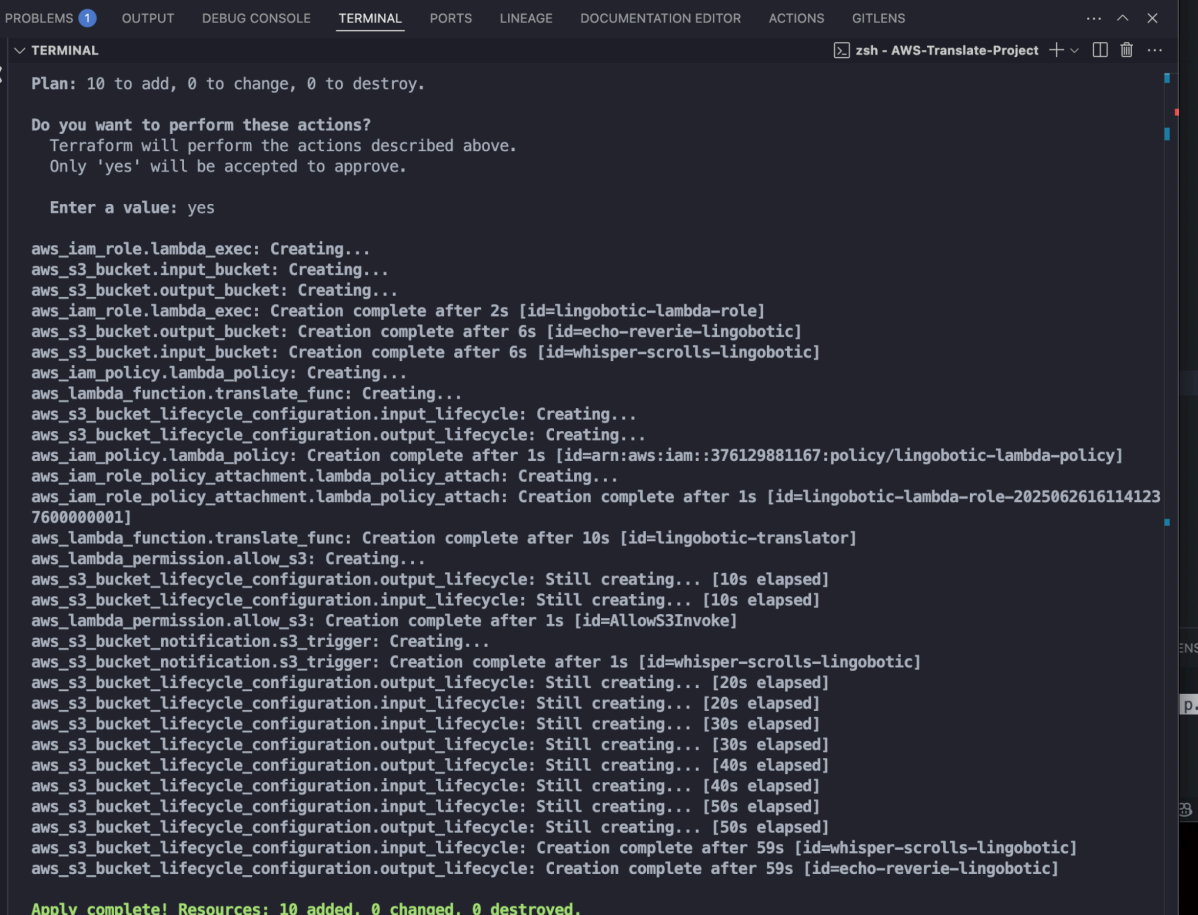
    expiration {
      days = 30
    }
  }
}

# Lifecycle for output bucket
resource "aws_s3_bucket_lifecycle_configuration" "output_lifecycle" {
  bucket = aws_s3_bucket.output_bucket.id
}
```

terraform init

Terraform plan

4. Deploy to Lambda via Terraform or manually.



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS LINEAGE DOCUMENTATION EDITOR ACTIONS GITLENS
zsh - AWS-Translate-Project + - [ ] [ ] ...

Plan: 10 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

aws_iam_role.lambda_exec: Creating...
aws_s3_bucket.input_bucket: Creating...
aws_s3_bucket.output_bucket: Creating...
aws_iam_role.lambda_exec: Creation complete after 2s [id=lingobotic-lambda-role]
aws_s3_bucket.output_bucket: Creation complete after 6s [id=echo-reverie-lingobotic]
aws_s3_bucket.input_bucket: Creation complete after 6s [id=whisper-scrolls-lingobotic]
aws_iam_policy.lambda_policy: Creating...
aws_lambda_function.translate_func: Creating...
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Creating...
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Creating...
aws_iam_policy.lambda_policy: Creation complete after 1s [id=arn:aws:iam::376129881167:policy/lingobotic-lambda-policy]
aws_iam_role_policy_attachment.lambda_policy_attach: Creating...
aws_iam_role_policy_attachment.lambda_policy_attach: Creation complete after 1s [id=lingobotic-lambda-role-20250626161141237600000001]
aws_lambda_function.translate_func: Creation complete after 10s [id=lingobotic-translator]
aws_lambda_permission.allow_s3: Creating...
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Still creating... [10s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Still creating... [10s elapsed]
aws_lambda_permission.allow_s3: Creation complete after 1s [id=AllowS3Invoke]
aws_s3_bucket_notification.s3_trigger: Creating...
aws_s3_bucket_notification.s3_trigger: Creation complete after 1s [id=whisper-scrolls-lingobotic]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Still creating... [20s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Still creating... [20s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Still creating... [30s elapsed]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Still creating... [30s elapsed]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Still creating... [40s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Still creating... [40s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Still creating... [50s elapsed]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Still creating... [50s elapsed]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Creation complete after 59s [id=whisper-scrolls-lingobotic]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Creation complete after 59s [id=echo-reverie-lingobotic]

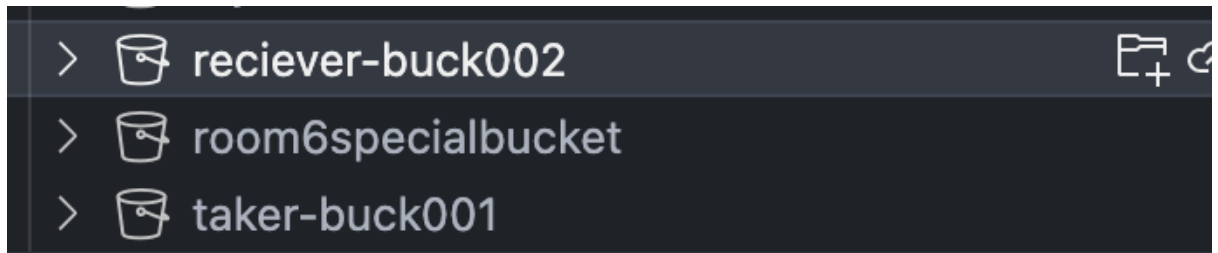
Apply complete! Resources: 10 added, 0 changed, 0 destroyed.
```

zip lambda_function.zip [index.py](#)



```
index.py
lambda_function.zip
```

5. Check if resources have been created, eg, buckets

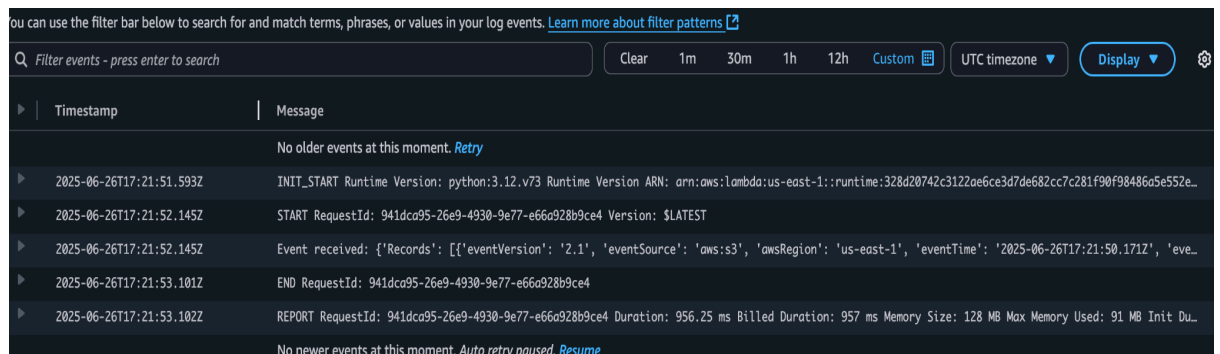


6. Upload test file:

```
aws s3 cp ../test/test_input.json s3://reciever-buck002/
```

Monitoring

- Use CloudWatch Logs:



```

aws logs tail /aws/lambda/lang-translator --follow
2025-06-26T17:14:56.002000+00:00 2025/06/26/[$LATEST]287dfb7c5ead44eb82b25852b5dfa9c4 INIT_START Runtime Version: python:3.12.v73 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:328d20742c3122ae6ce3d7de682cc7c281f90f98486a5e552e4849d3a1838c05
2025-06-26T17:14:56.503000+00:00 2025/06/26/[$LATEST]287dfb7c5ead44eb82b25852b5dfa9c4 START RequestId: 398fdc0b-90e3-4891-a558-750cf67e9427 Version: $LATEST
2025-06-26T17:14:56.504000+00:00 2025/06/26/[$LATEST]287dfb7c5ead44eb82b25852b5dfa9c4 Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1', 'eventTime': '2025-06-26T17:14:54.648Z', 'eventName': 'ObjectCreated:Put', 'userIdentity': {'principalId': 'AWS:AIDAVPEYQWRHVB45JNTL3'}, 'requestParameters': {'sourceIPAddress': '154.160.10.106'}, 'responseElements': {'x-amz-request-id': '3AYHXFPXPMRD3ZV', 'x-amz-id-2': 'iaMCIWCaZKMLZW9H1fxAY/AGaHenKYNMM790tqM5p73TnED2Ze8wKEvx6dR7f8AFVUVJ7tzSxE3bU6a gKJst7hqYaiNjsh'}, 's3': {'s3SchemaVersion': '1.0', 'configurationId': 'tf-s3-lambda-20250626163540655700000002', 'bucket': {'name': 'reciever-buck002', 'ownerIdentity': {'principalId': 'A1E482WNADL55'}, 'arn': 'arn:aws:s3:::reciever-buck002'}, 'object': {'key': 'test_input.json', 'size': 224, 'eTag': '16c6d47206bfc095337bf37575cd5caa', 'sequencer': '00685080E9BBCEE77'}}]}
2025-06-26T17:14:57.494000+00:00 2025/06/26/[$LATEST]287dfb7c5ead44eb82b25852b5dfa9c4 END RequestId: 398fdc0b-90e3-4891-a558-750cf67e9427
2025-06-26T17:14:57.494000+00:00 2025/06/26/[$LATEST]287dfb7c5ead44eb82b25852b5dfa9c4 REPORT RequestId: 398fdc0b-90e3-4891-a558-750cf67e9427 Duration: 989.94 ms Billed Duration: 990 ms Memory Size: 128 MB Max Memory Used: 91 MB Init Duration: 498.33 ms
2025-06-26T17:21:51.593000+00:00 2025/06/26/[$LATEST]05c1c4719ce6400cb7e96b45b416f08b INIT_START Runtime Version: python:3.12.v73 Runtime Version ARN: arn:aws:lambda:us-east-1::runtime:328d20742c3122ae6ce3d7de682cc7c281f90f98486a5e552e4849d3a1838c05
2025-06-26T17:21:52.145000+00:00 2025/06/26/[$LATEST]05c1c4719ce6400cb7e96b45b416f08b START RequestId: 941dca95-26e9-4930-9e77-e66a928b9ce4 Version: $LATEST
2025-06-26T17:21:52.145000+00:00 2025/06/26/[$LATEST]05c1c4719ce6400cb7e96b45b416f08b Event received: {'Records': [{'eventVersion': '2.1', 'eventSource': 'aws:s3', 'awsRegion': 'us-east-1', 'eventTime': '2025-06-26T17:21:50.171Z', 'eventName': 'ObjectCreated:Put', 'userIdentity': {'principalId': 'AWS:AIDAVPEYQWRHVB45JNTL3'}, 'requestParameters': {'sourceIPAddress': '154.160.10.106'}, 'responseElements': {'x-amz-request-id': 'N1QC6M23V72WY89N', 'x-amz-id-2': 'M/PbBQCwFzea38hCFiilVcNkJPVtRsLVepHK8B1dwc3/aZ+YV1Gdp500PW9ahHIBxFOlsG7ynMF2uTYLDa5lW0UkiCexfc'}, 's3': {'s3SchemaVersion': '1.0', 'configurationId': 'tf-s3-lambda-20250626163540655700000002', 'bucket': {'name': 'reciever-buck002', 'ownerIdentity': {'principalId': 'A1E482WNADL55'}, 'arn': 'arn:aws:s3:::reciever-buck002'}, 'object': {'key': 'test_input.json', 'size': 224, 'eTag': '16c6d47206bfc095337bf37575cd5caa', 'sequencer': '00685080E9BBCEE77'}}]}

```

Future Improvements

- Add SNS notifications on successful translation
- Add DynamoDB to store translation history
- Add a front-end upload interface
- Include language detection (if SourceLanguageCode is missing)
- Validate file content format before processing

Cleanup terraform destroy

```

terraform destroy
aws_iam_role.lambda_exec: Refreshing state... [id=tranc-lambda-role]
aws_s3_bucket.output_bucket: Refreshing state... [id=taker-buck001]
aws_s3_bucket.input_bucket: Refreshing state... [id=reciever-buck002]
aws_iam_policy.lambda_policy: Refreshing state... [id=arn:aws:iam::376129881167:policy/tranc-lambda-policy]
aws_s3_bucket_lifecycle_configuration.output_lifecycle: Refreshing state... [id=taker-buck001]
aws_s3_bucket_lifecycle_configuration.input_lifecycle: Refreshing state... [id=reciever-buck002]
aws_lambda_function.translate_func: Refreshing state... [id=lang-translator]
aws_iam_role_policy_attachment.lambda_policy_attach: Refreshing state... [id=tranc-lambda-role-20250626163529247000000001]
aws_lambda_permission.allow_s3: Refreshing state... [id=AllowS3Invoke]
aws_s3_bucket_notification.s3_trigger: Refreshing state... [id=reciever-buck002]

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
- destroy

Terraform will perform the following actions:

# aws_iam_policy.lambda_policy will be destroyed
- resource "aws_iam_policy" "lambda_policy" {
  - arn          = "arn:aws:iam::376129881167:policy/tranc-lambda-policy" -> null
  - attachment_count = 1 -> null
  - id           = "arn:aws:iam::376129881167:policy/tranc-lambda-policy" -> null
  - name        = "tranc-lambda-policy" -> null
  - path        = "/" -> null
  - policy      = isonencode(

```


IAM Masters

Nathaniel Sackey

Chrysolite Yeboah

Emmanuel Osei

Kelvin NeeQuaye

Eric Datsa

Mariam Zakuyu

GitHub: [Github repo](#)