**Zombieland CLI Team**
Cali, Colombia

# Weekly status report, 1 of 3.

## Automatic Translations, Skillshare.

**June 4, 2020**

### Team

1. Carlos Andres Garcia (agzsoftsi)
2. Ivan Dario Lasso (ilasso)
3. Leonardo Calderon (leocjj)

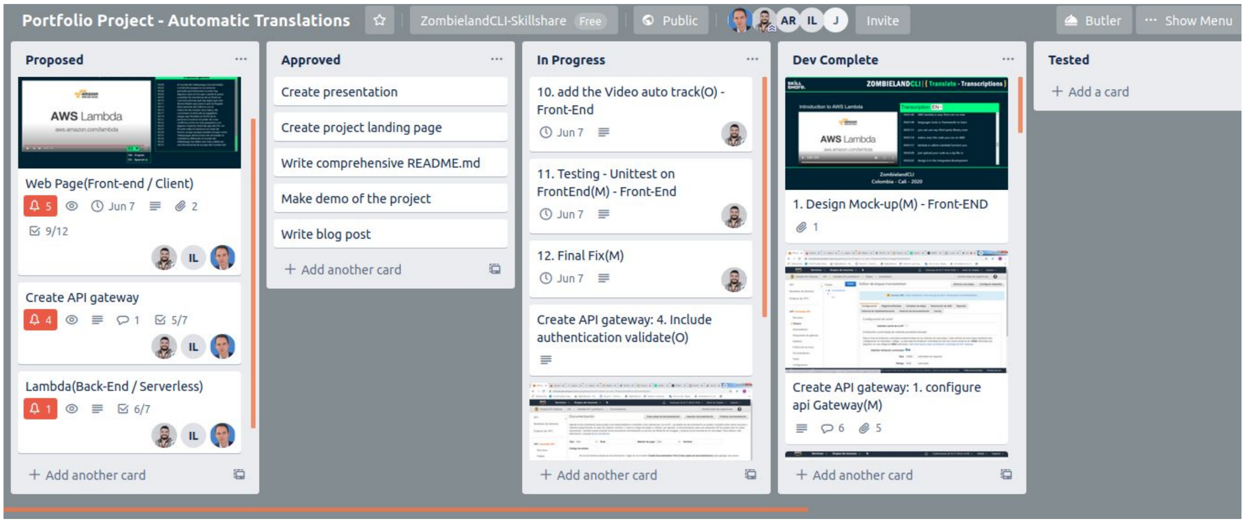
## Progress

This is the first of three weekly progress reports to present the status of the project advance in order to achieve main goals, to meet the schedule and requirements from Holberton and the Client.

As we will show next, we achieved 86% (2 of 14) of mandatory requirements from the Client at this point, measured in Trello project (see links related). At this moment we are working on the rest of the mandatory tasks and on additional tasks that are nice-to-have. All mandatory tasks in Dev Complete column in Trello and completely deployed to a landing page fully operational.

On June 4 meeting, the Client representative accepts the actual progress as a project fully operational and suggests to start research on a possible deployment on production environment connecting the service developed in this project with the external services used in Client's web page. We will start this advanced stage on the project, not accorded in original requirements, next week (June 8).

Mandatory tasks pending are related to additional tests and documentation and are in progress with an estimated finish date next week.



Portfolio Project - Automatic Translations

ZombielandCLI-Skillshare Free | Public | Invite | Butler | Show Menu

**Proposed**
- Web Page(Front-end / Client) — 5, Jun 7, 2, 9/12
- Create API gateway — 4, 1, 5/7
- Lambda(Back-End / Serverless) — 1, 6/7
- Add another card

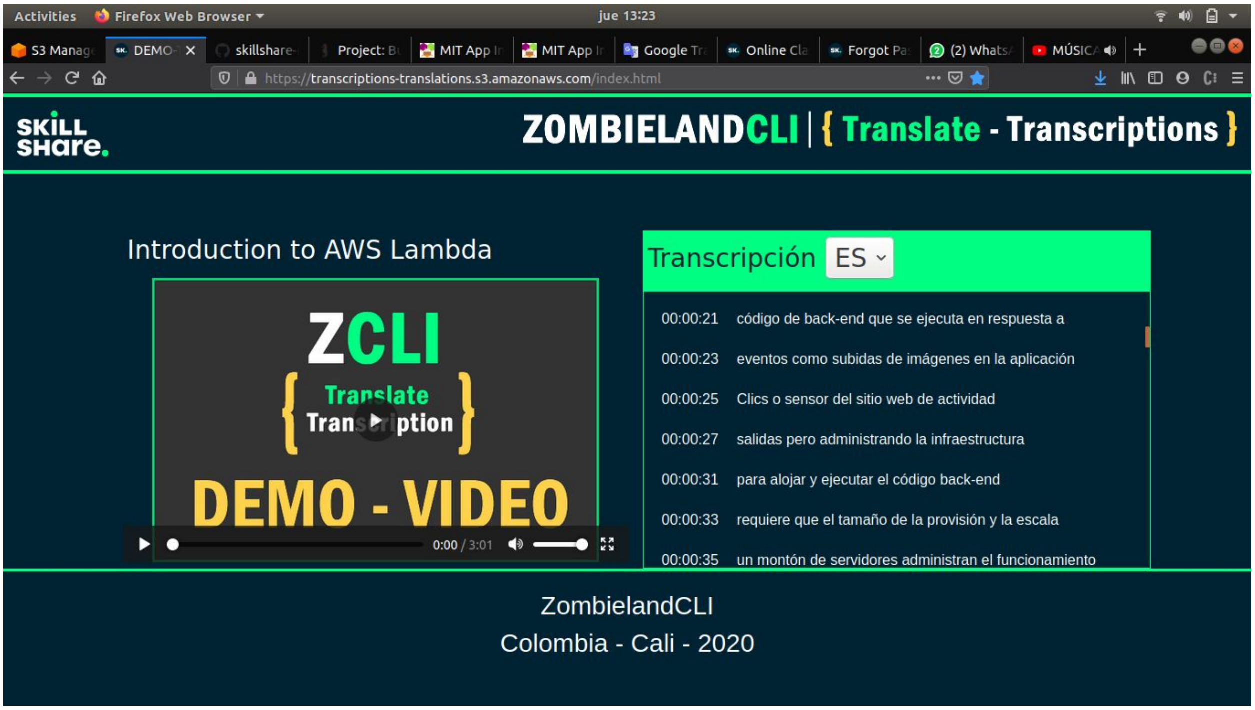
**Approved**
- Create presentation
- Create project landing page
- Write comprehensive README.md
- Make demo of the project
- Write blog post
- Add another card

**In Progress**
- 10. add the Video auto track(O) - Front-End — Jun 7
- 11. Testing - Unittest on FrontEnd(M) - Front-End — Jun 7
- 12. Final Fix(M) — Jun 7
- Create API gateway: 4. Include authentication validate(O)
- Add another card

**Dev Complete**
- 1. Design Mock-up(M) - Front-END — 1
- Create API gateway: 1. configure api Gateway(M) — 6, 5
- Add another card

**Tested**
- Add a card

Figure 1. Trello project plan (status on June 4)



https://transcriptions-translations.s3.amazonaws.com/index.html

ZOMBIELANDCLI | { Translate - Transcriptions }

Introduction to AWS Lambda

Transcripción ES

| | |
|---|---|
| 00:00:21 | código de back-end que se ejecuta en respuesta a |
| 00:00:23 | eventos como subidas de imágenes en la aplicación |
| 00:00:25 | Clics o sensor del sitio web de actividad |
| 00:00:27 | salidas pero administrando la infraestructura |
| 00:00:31 | para alojar y ejecutar el código back-end |
| 00:00:33 | requiere que el tamaño de la provisión y la escala |
| 00:00:35 | un montón de servidores administran el funcionamiento |

ZombielandCLI
Colombia - Cali - 2020

Figure 2. Landing page for the service.

# Challenges

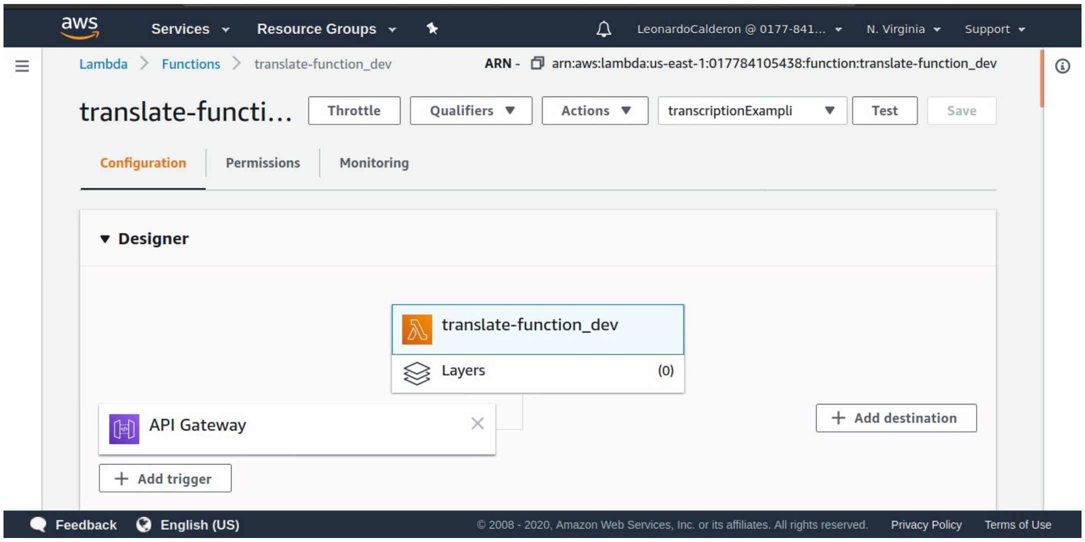
The main challenges in the project are related to the use implementation of new technologies specifically AWS services required by the client to implement back-end functionality.

All data used for the project are also en AWS services as S3 objects from where we need to recover input transcriptions for the new translation service. The following are the main AWS services researched as part of the implementation:

- Lambda function
- S3 objects
- API Gateway
- Amazon Translate

In the middle of the project, we found a special limitation for using the Amazon Translate service, this is limited to 5000 words per request so we needed to implement some code to work with that and also to deal with the speed of the service to avoid long time of wait for the user when requesting a new language translation. No other situations were presented

However, the original plan did not change due to several tests performed in parallel by the members of the team and pair programming sessions that helped to overcome the issues.



ARN - arn:aws:lambda:us-east-1:017784105438:function:translate-function_dev

translate-functi...

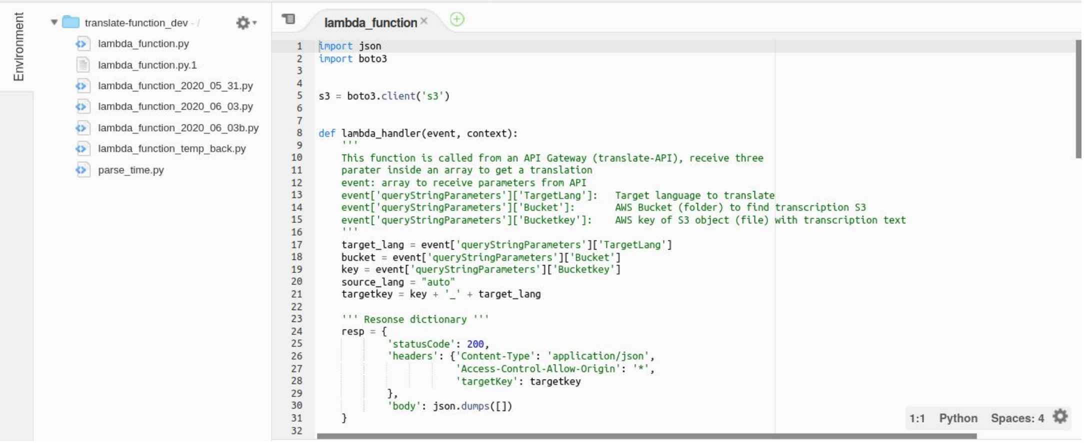
transcriptionExampli

translate-function_dev

API Gateway

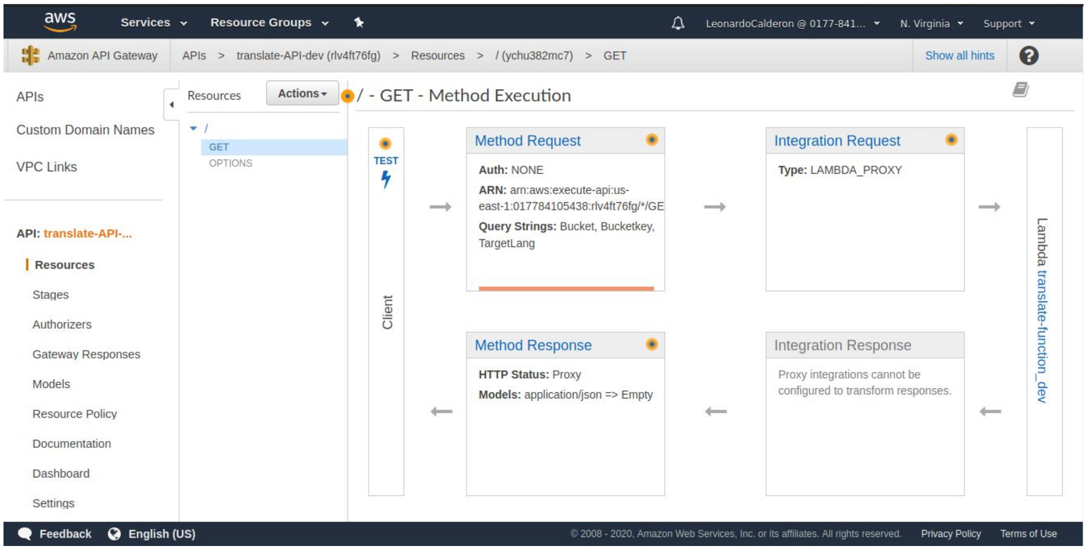
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Figure 3. Lambda function triggered by API Gateway request



```python
import json
import boto3


s3 = boto3.client('s3')


def lambda_handler(event, context):
    '''
    This function is called from an API Gateway (translate-API), receive three
    parater inside an array to get a translation
    event: array to receive parameters from API
    event['queryStringParameters']['TargetLang']:   Target language to translate
    event['queryStringParameters']['Bucket']:       AWS Bucket (folder) to find transcription S3
    event['queryStringParameters']['Bucketkey']:    AWS key of S3 object (file) with transcription text
    '''
    target_lang = event['queryStringParameters']['TargetLang']
    bucket = event['queryStringParameters']['Bucket']
    key = event['queryStringParameters']['Bucketkey']
    source_lang = "auto"
    targetkey = key + '_' + target_lang

    ''' Resonse dictionary '''
    resp = {
            'statusCode': 200,
            'headers': {'Content-Type': 'application/json',
                        'Access-Control-Allow-Origin': '*',
                        'targetKey': targetkey
            },
            'body': json.dumps([])
    }

```

Figure 4. Lambda function code in Python



/ - GET - Method Execution

Method Request

**Auth:** NONE

**ARN:** arn:aws:execute-api:us-east-1:017784105438:rlv4ft76fg/*/GE

**Query Strings:** Bucket, Bucketkey, TargetLang

Integration Request

**Type:** LAMBDA_PROXY

Method Response

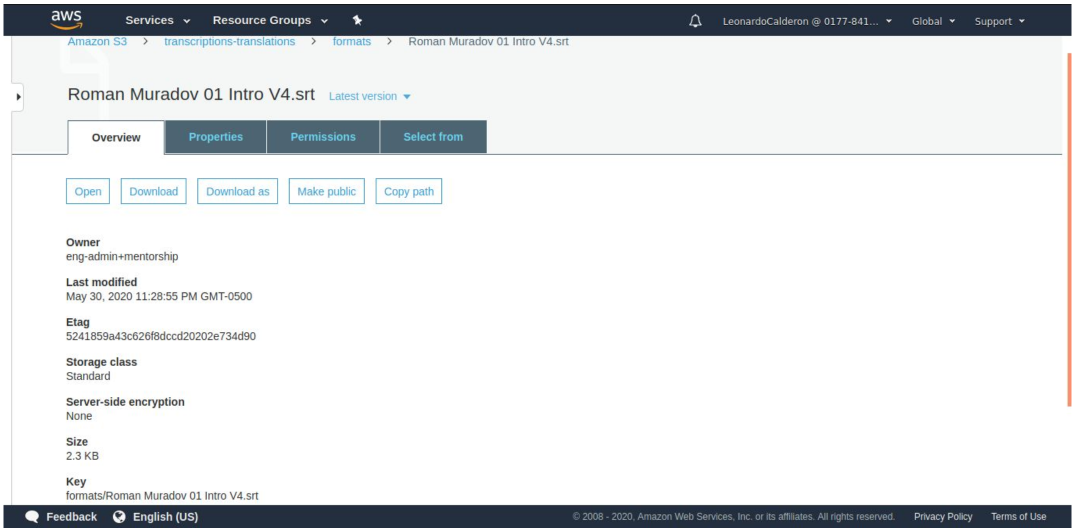
**HTTP Status:** Proxy

**Models:** application/json => Empty

Integration Response

Proxy integrations cannot be configured to transform responses.

Figure 5. API Gateway configuration of the GET method.


aws
Services
Resource Groups
LeonardoCalderon @ 0177-841...
Global
Support
Amazon S3 > transcriptions-translations > formats > Roman Muradov 01 Intro V4.srt
Latest version
Overview
Properties
Permissions
Select from
Open
Download
Download as
Make public
Copy path
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Roman Muradov 01 Intro V4.srt

**Owner**
eng-admin+mentorship

**Last modified**
May 30, 2020 11:28:55 PM GMT-0500

**Etag**
5241859a43c626f8dccd20202e734d90

**Storage class**
Standard

**Server-side encryption**
None

**Size**
2.3 KB

**Key**
formats/Roman Muradov 01 Intro V4.srt

Figure 6. S3 object in AWS

## Collaboration

We needed to deal with a special situation in this project. Due to global isolation, all the teamwork was performed remotely and we as members were forced to work virtually all the time. This was achieved using a videoconference service, individual accounts for Skillshare and AWS, development repository in GitHub, and a set of rules described next:

- Daily meeting with the Client representative was mandatory.
- Daily team meetings to share individual research and do pair programming to integrate codes and new features.
- Special rule to avoid migration to production code without the approval of all the team.

## Project Updates

According to the original requirements

Because we are one week in advance according to the schedule of the project, we decided to increase the scope and include some nice-to-have features and activities:

- Video auto-track front-end
- Authentication for API
- API documentation inside AWS
- Research deployment in real Skillshare production.

## Links related to the project

**Trello project:** https://trello.com/b/mfR5ex9u/portfolio-project-automatic-translations

**Landing page:** https://transcriptions-translations.s3.amazonaws.com/index.html

**Client's page:** https://www.skillshare.com/

**Official repository:** https://github.com/skillshare-mentorship/holberton-translate-transcriptions