

A photograph of two men sitting on a grey sofa in a modern office. They are both looking at their laptops. One man is holding a black laptop, and the other is holding a white one. There is a small round wooden coffee table in front of them with some papers and a mug on it. A large orange lamp is hanging above them. The background shows a glass wall and some pipes on the ceiling.

# GOING SERVERLESS WITH YOUR OWN ONLINE TRANSLATOR



*An easy AWS lambda use case*

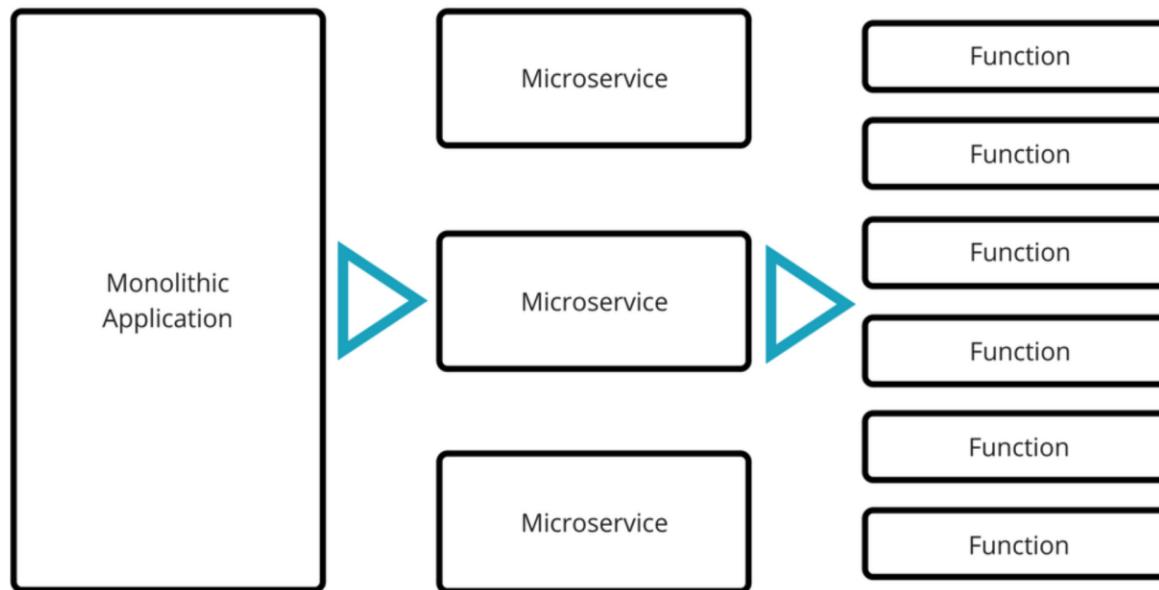
# AGENDA

- INTRO TO SERVERLESS
- ARCHITECTURE USE CASES
- AWS SERVERLESS SERVICES
- BEST PRACTICES
- STEP BY STEP CONFIG
- TRANSLATOR DEMO
- TRANSLATOR BUDGET

# INTRO TO SERVERLESS

---

Another name for serverless computing is Function-as-a-Service FaaS, referring to the way developers **assemble code into building blocks called functions**.



Following microservices approach where a monolith is split into small elements that can be scaled and updated separately and parallelly, **FaaS breaks each microservice even further**.

# INTRO TO SERVERLESS

---

For example, for your home security camera you obviously don't want to record everything that happens on your street 24/7. That's why we use motion-activated cameras to detect suspicious behaviour when we're not at home.

Serverless architecture works similarly, just like a motion sensor, it only works or run when **a pre-programmed event is triggered**.

Serverless is stateless, meaning it only executes a task and doesn't store or reuse requests.

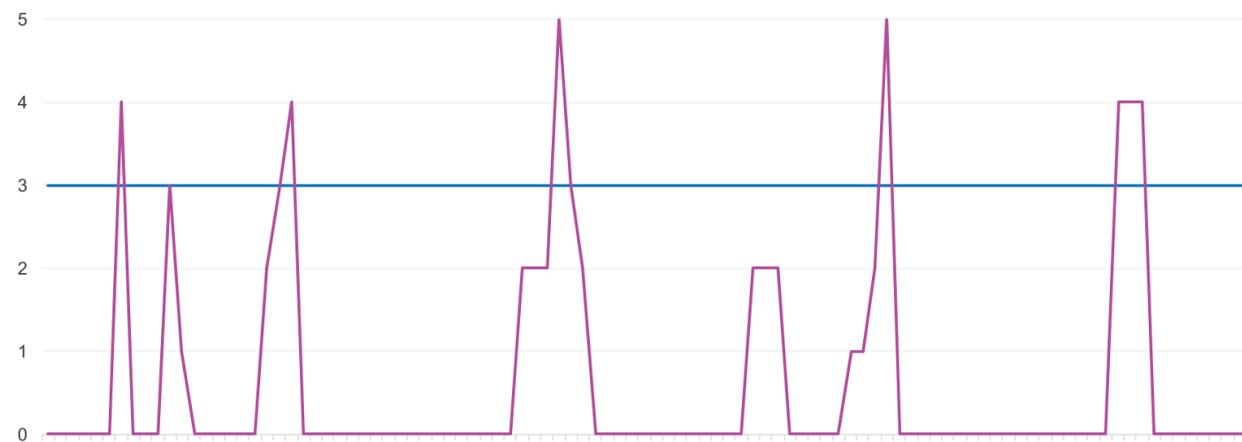


# INTRO TO SERVERLESS

---

Serverless approach is flexible and ideal for scaling applications. FaaS vendors takes each of your functions and runs them in containers. This allows you to scale them endlessly and automatically.

This is another difference between serverless and traditional cloud – here you don't have to purchase the assumed amount of resources; **you can be as flexible as possible.**



## Virtual Machines

Virtual Machines run constantly at the provisioned capacity



## Serverless

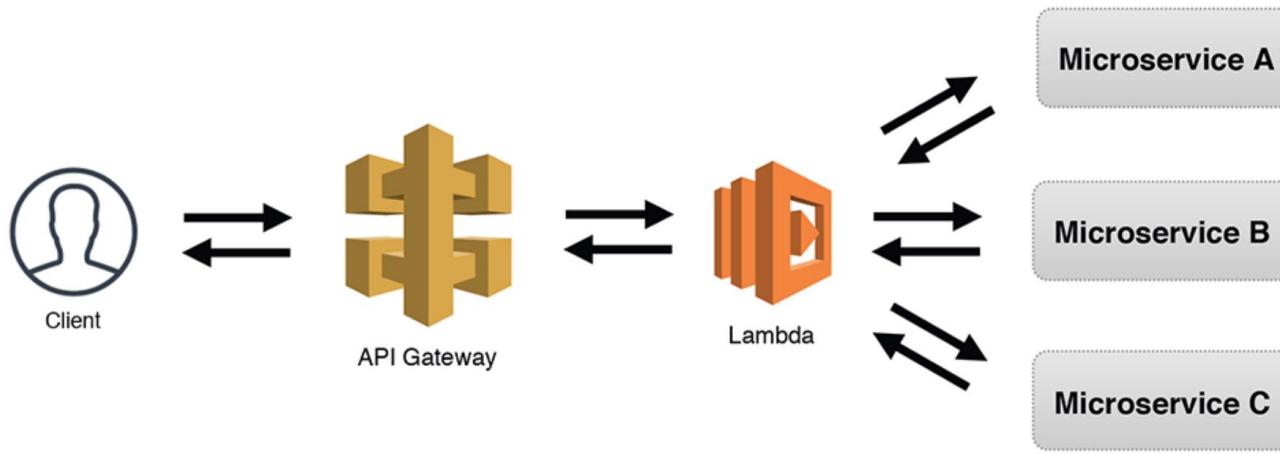
Function-as-a-Service respond and scale to meet demands

# INTRO TO SERVERLESS

---

In a traditional cloud model, servers have to be ready to process requests at all times. Constant server availability leads to significant backend costs every month, irrespective of CPU time and memory that are actually used. It's like leaving your heater on all year long, regardless of the weather or change of seasons, and paying huge sums for unneeded electricity.

Alternatively, serverless vendors like AWS, allow you to **pay a fraction of a price per request**, which means that your costs will depend only on how much traffic you had this month.



# INTRO TO SERVERLESS

---



## Function-as-a-Service

Developers assemble code into building blocks called functions.



## Event-driven coding

Function executes a task and doesn't store or reuse requests.



## Scalable service

Small elements flexible and ideal for scale them automatically.



## Billing per invocation

Costs will depend only on how much traffic your function had.

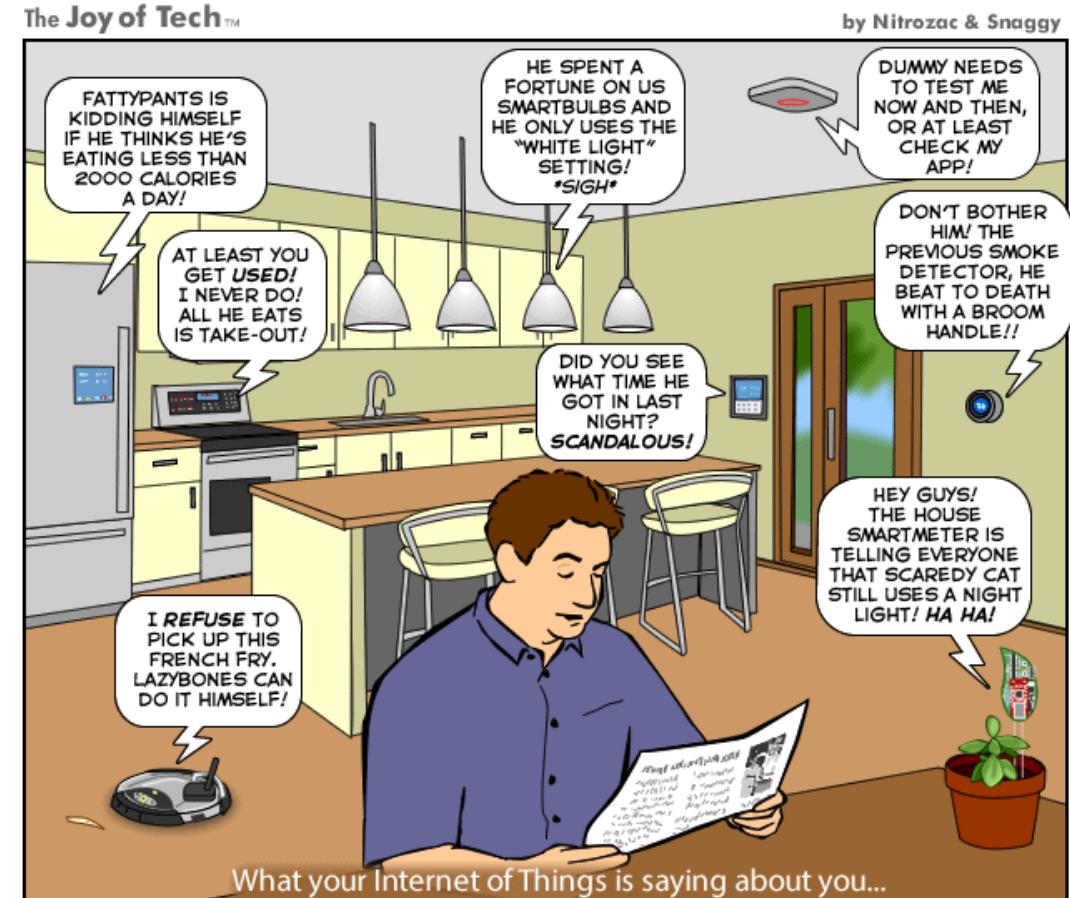
# ARCHITECTURE USE CASES

## Internet of Things applications

The real-time response nature of the serverless approach works great for IoT use cases.

Motion activated cameras, applications that react to changes in weather, temperature, or health conditions are perfect for the serverless

Your services won't be sitting idle 24/7.



© 2014 Geek Culture

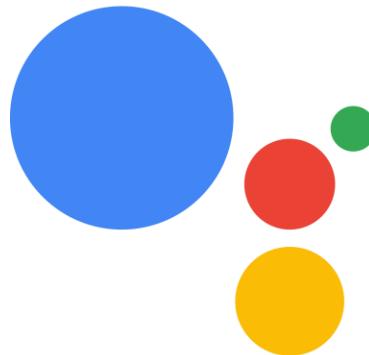
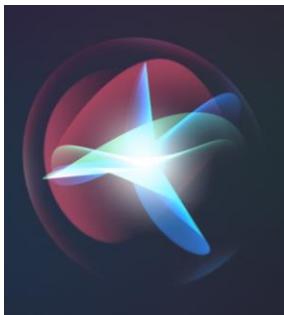
joyoftech.com

# ARCHITECTURE USE CASES

---

## Virtual assistants and chatbots

People using chats expect immediate responses which is why serverless data processing can be faster. As your application grows from one hundred to several thousand users, your processing time should also stay the same which is automated with FaaS.

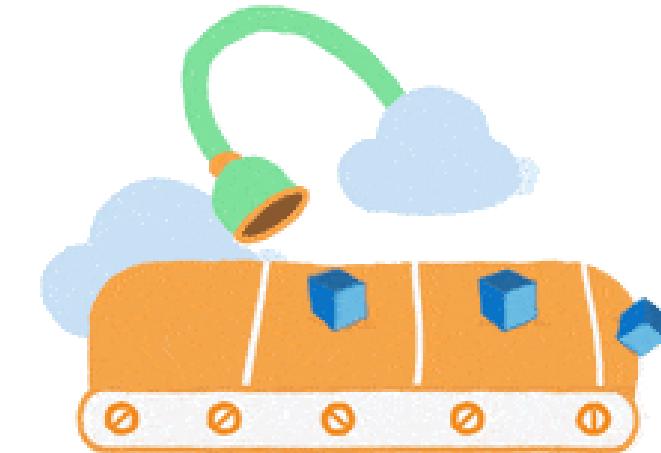
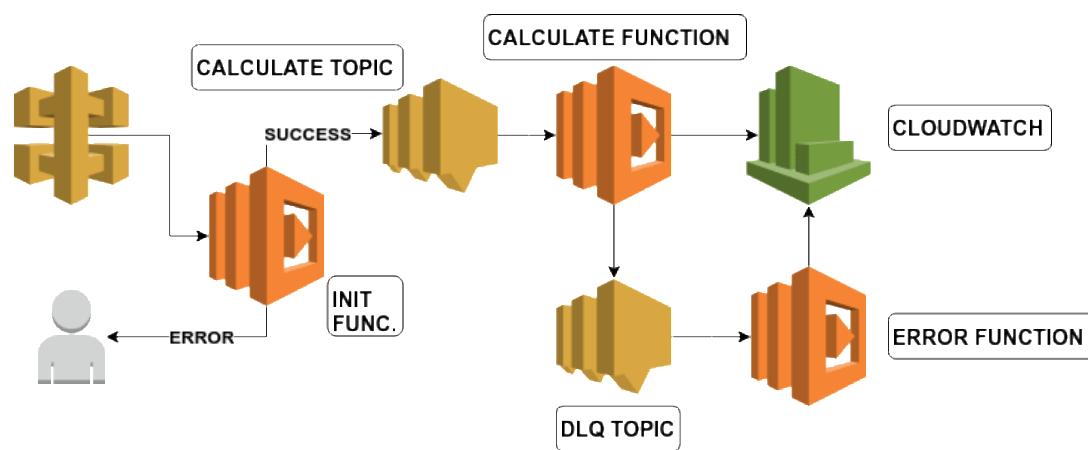


# ARCHITECTURE USE CASES

---

## Agile and Continuous Integration pipelines

The idea of running the code only when a certain event is triggered is perfectly in line with Agile or Continuous Integration principles. Separating the codebase into functions also helps with bug fixing and shipping updates. Serverless is an overall friendly way for maximum automation and rapid deployment processes.



# ARCHITECTURE USE CASES

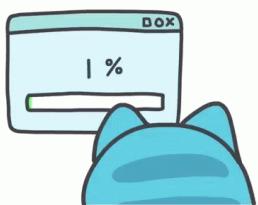
---



Sorting



Filtering



Real-time File processing



Real-time Stream processing



3rd-party API request

# AWS SERVERLESS SERVICES

---

AWS provides **fully managed services** that can be used to build and run serverless applications.



This services handle application fault tolerance and availability capabilities. Tasks like provisioning, maintaining, and managing servers for backend components such as compute, databases, storage, message queueing, stream processing, and many more are no longer required.

# AWS SERVERLESS SERVICES

---

## AWS Lambda

Serverless computing system that lets you **automatically run code** in response to many types of events, such as HTTP requests from Amazon API gateway, table updates in Amazon DynamoDB, and state transitions.

The code runs on a high-availability computer infrastructure, where AWS performs all the administrative duties of that compute resource, like providing maintenance over the operating system, automatically scaling and managing capacity provisions, and handling security patching and code monitoring and logging.

<https://aws.amazon.com/lambda/>



# AWS SERVERLESS SERVICES

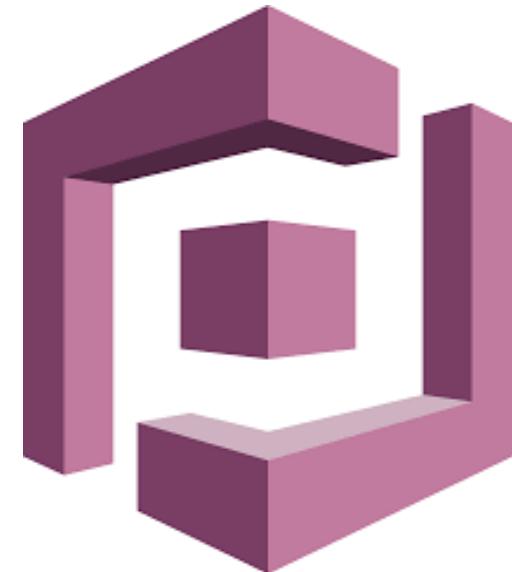
---

## AWS Cognito

Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily. Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

- Sign-In & Sign-up to mobile and web apps
- Federation (you like MS AD? Fine, fancy social networks? got it)
- Control AWS resources access, and restrict it to the public you want

<https://aws.amazon.com/cognito/>



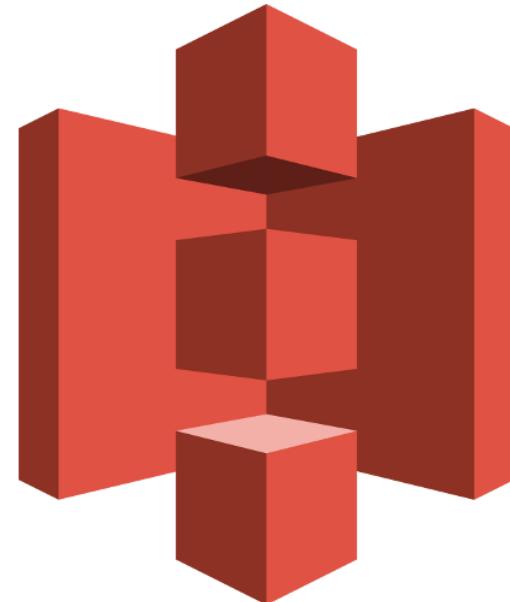
# AWS SERVERLESS SERVICES

---

## AWS Simple Storage Service (S3)

Object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can use it to store and protect any amount of data.

- Unlimited storage (Limit 5TB per file)
- Event notifications and Lifecycle management
- Host entire static websites
- Designed for 99.99999999% (11 9's) of data durability



<https://aws.amazon.com/s3/>

# AWS SERVERLESS SERVICES

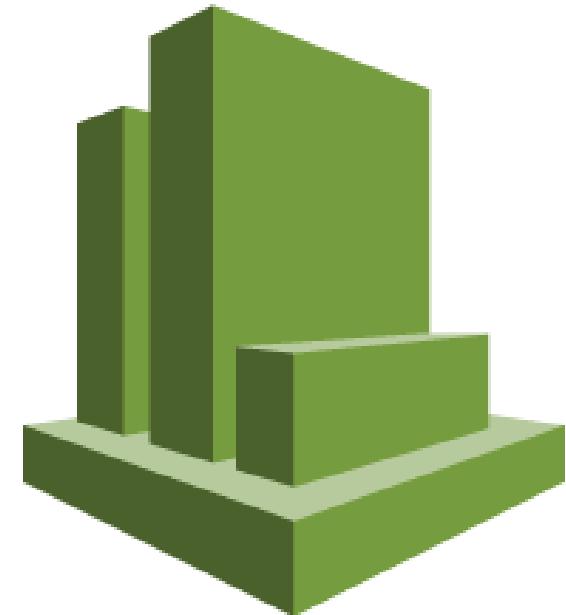
---

## AWS CloudWatch

Monitoring and observability service that provides data and actionable insights to monitor your applications, respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health.

- Observability applications and infrastructure
- Easily collect and store logs, metrics and events
- Built-in and custom metrics
- Log analytics and auto scaling

<https://aws.amazon.com/cloudwatch/>



# BEST PRACTICES

---

## When you go serverless...



- Each function should do only one thing.
- Avoid using connection-based services e.g. RDBMS.
- Code for scaling and consider how it scales.
- Use as few libraries in your functions as possible.
- One function per route (if using HTTP).
- Learn to use messages and queues (async FTW).
- Data flows, not data lakes.
- Functions don't call other functions.

# STEP BY STEP CONFIG

1

Allow access to your site with **Amazon CloudFront**, which allows you to get an HTTPS link to your page and which is required by some browsers to record audio.

CloudFront Distributions > E2H1XQB0BRVS22

General Origins and Origin Groups Behaviors Error Pages Restrictions Validations Tags

**Edit**

Distribution ID: E2H1XQB0BRVS22  
ARN: arn:aws:cloudfront:...:distribution/E2H1XQB0BRVS22  
Log Prefix: -  
Delivery Method: Web  
Cookie Logging: Off  
Distribution Status: Deployed  
Comment: CloudFront distribution over Voice Translator App - to enable HTTPS  
Price Class: Use All Edge Locations (Best Performance)  
AWS WAF Web ACL: -  
State: Enabled  
Alternate Domain Names (CNAMEs): translate.carlosaherrera.com  
SSL Certificate: \*.carlosaherrera.com (5840ae4c-7c29-489d-9fc4-87d209167d82)  
Domain Name: db07t6s3225t0.cloudfront.net  
Custom SSL Client Support: Clients that Support Server Name Indication (SNI) - (Recommended)  
Security Policy: TLSv1.1\_2016  
Supported HTTP Versions: HTTP/1.1, HTTP/1.0  
IPv6: Disabled  
Default Root Object: voice-translator.html  
Last Modified: 2019-11-24 11:11 UTC-5  
Log Bucket: -

If you need to prevent users in selected countries from accessing your content, you can specify either a whitelist (countries where they can access your content) or a blacklist (countries where they cannot). For more information, see [Restricting the Geographic Distribution of Your Content](#) in the *Amazon CloudFront Developer Guide*.

**Edit**

	Restriction	Status	Type
<input type="checkbox"/> Geo Restriction	Disabled	-	

**CloudFront Distributions**

Action Distribution Settings Delete Enable Disable

Delivery Method	ID	Domain Name	Comment	Origin
Web	E2H1XQB0BRVS22	db07t6s3225t0.cloudfront.net	CloudFront distribution over Voice Translator App	voicetest-voicetranslatorbucket

# STEP BY STEP CONFIG

2.

Host your page on **Amazon S3**, which simplifies the whole solution. This is also the place to save the input audio file recorded in the browser.

Amazon S3 > voicetest-voicetranslatorbucket

Présentation Propriétés Autorisations Public Gestion

Saisir un préfixe et appuyer sur Entrée pour lancer la recherche. Pour annuler, appuyer sur ESC.

Charger + Créer un dossier Télécharger Actions ▾ USA Est (Virginie du Nord) Affichage 1 à 5

Nom	Dernière modification	Taille	Classe de stockage
css	--	--	--
graphics	--	--	--
voice-translator-config.js	nov. 22, 2019 3:20:24 PM GMT-0500	205.0 o	Standard
voice-translator.html	nov. 22, 2019 3:20:44 PM GMT-0500	6.7 Ko	Standard
voice-translator.js	nov. 22, 2019 3:20:56 PM GMT-0500	7.7 Ko	Standard

Affichage 1 à 5

3:36 PM Ven. 22 nov. s3.console.aws.amazon.com cherrera @ devendava Global Support

Overview Properties Permissions Management

Versioning Keep multiple versions of an object in the same bucket. Learn more

Server access logging Set up access log records that provide details about access requests. Learn more

Static website hosting Host a static website, which does not require server-side technologies. Learn more

Bucket hosting

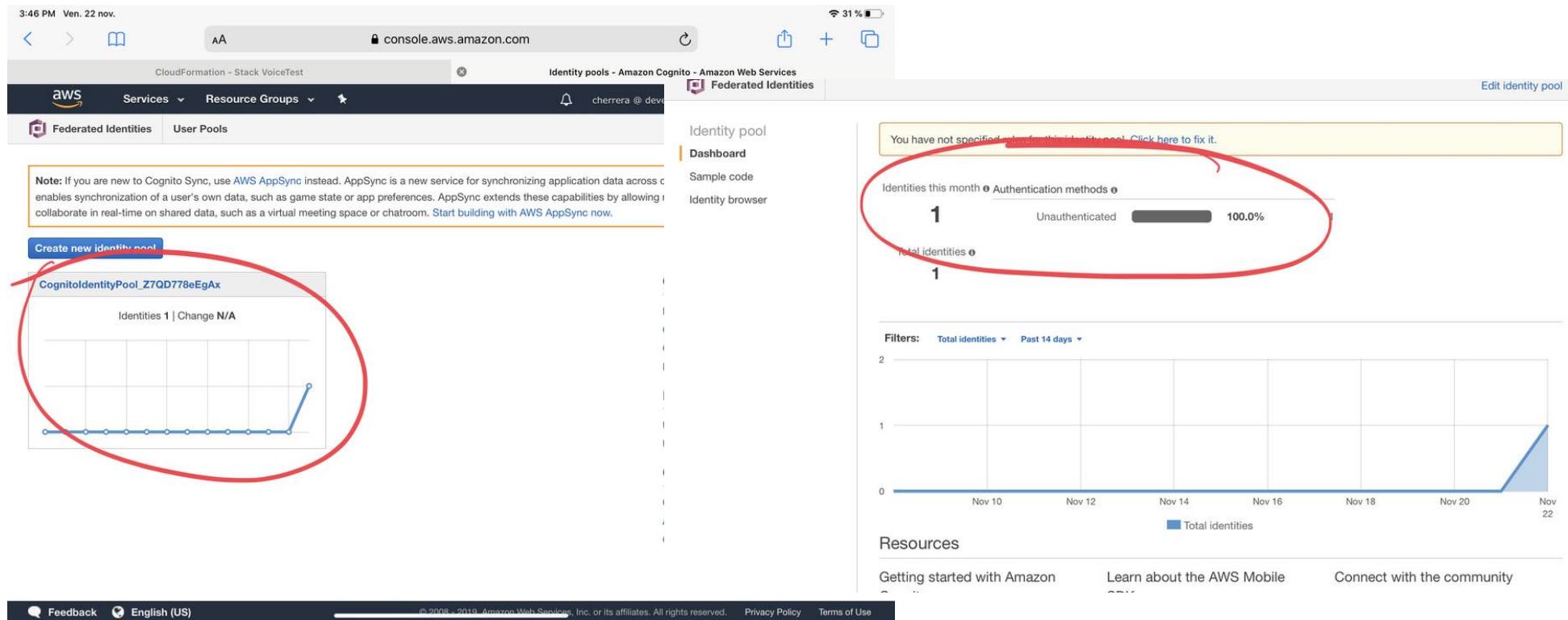
Object-level logging Record object-level API activity using the CloudTrail data events feature (additional cost). Learn more

Default encryption Automatically encrypt objects when stored in Amazon S3. Learn more

Feedback English (US) © 2006–2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

# STEP BY STEP CONFIG

## 3. Gain secure access to S3 and Lambda from the browser with **Amazon Cognito**.



# STEP BY STEP CONFIG

```
AWS.config.update({
  region: awsRegion,
  credentials: new AWS.CognitoIdentityCredentials({
    IdentityPoolId: IdentityPoolId
  })
});
```

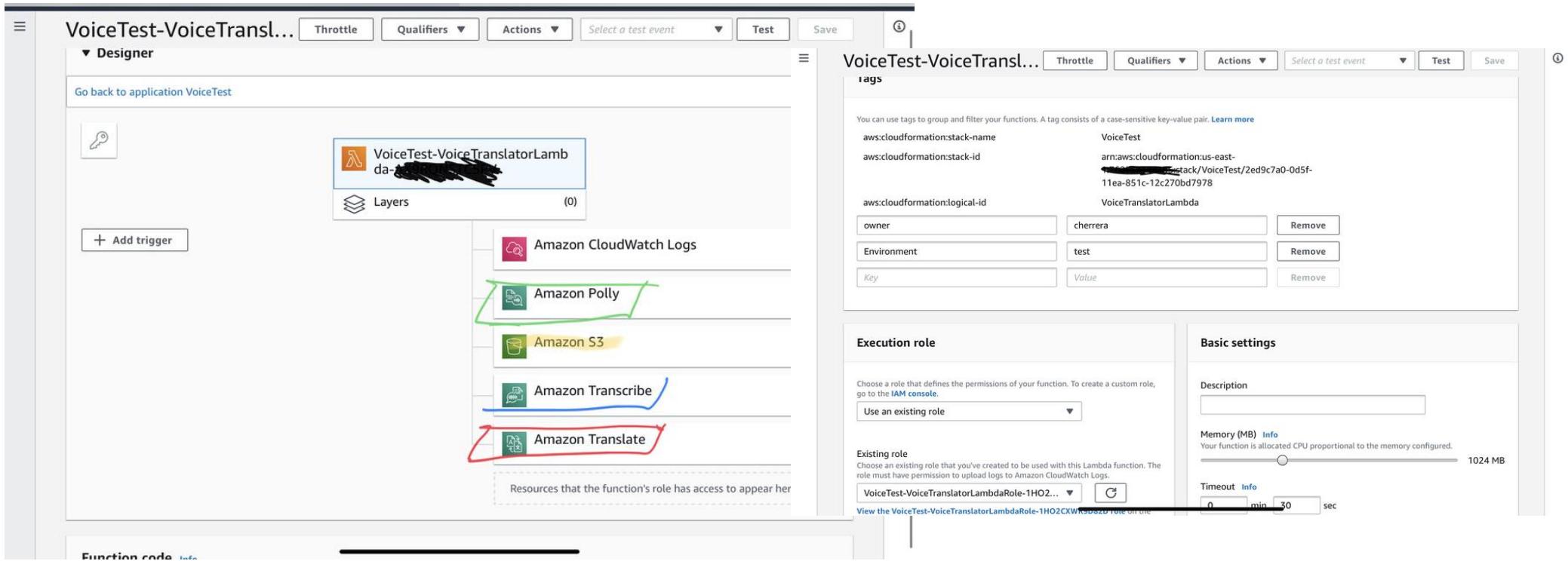
▶ voice-translator config.js voice-translator config.js [awsIdentity] config

```
1 var bucketName = "voicetest-voicetranslatorbucket-123456789012";
2 var IdentityPoolId = "us-east-1:d55044d7-f714-40cf-aebe-9723cb563dca";
3 var lambdaFunction = "VoiceTest-VoiceTranslatorLambda-AX9RQNS775YV";
4
```

```
CognitoUnAuthorizedRole:
Type: "AWS::IAM::Role"
Properties:
  AssumeRolePolicyDocument:
    Version: "2012-10-17"
    Statement:
      - Effect: "Allow"
        Principal:
          Federated: "cognito-identity.amazonaws.com"
        Action:
          - "sts:AssumeRoleWithWebIdentity"
Policies:
  - PolicyName: "CognitoUnauthorizedPolicy"
    PolicyDocument:
      Version: "2012-10-17"
      Statement:
        - Effect: "Allow"
          Action:
            - "lambda:InvokeFunction"
          Resource:
            - !GetAtt VoiceTranslatorLambda.Arn
Effect: Allow
Action:
  - "s3:PutObject"
Resource:
  Fn::Join:
    - ""
    - - "arn:aws:s3:::"
      - Ref: "VoiceTranslatorBucket"
      - "/"
```

# STEP BY STEP CONFIG

4. Save the input audio file on S3 and invoke a **Lambda function**. In the input of the function, provide the name of audio file, and pass the source and target language.



# STEP BY STEP CONFIG

4.

Save the input audio file on S3 and invoke a **Lambda function**. In the input of the function, provide the name of audio file, and pass the source and target language.

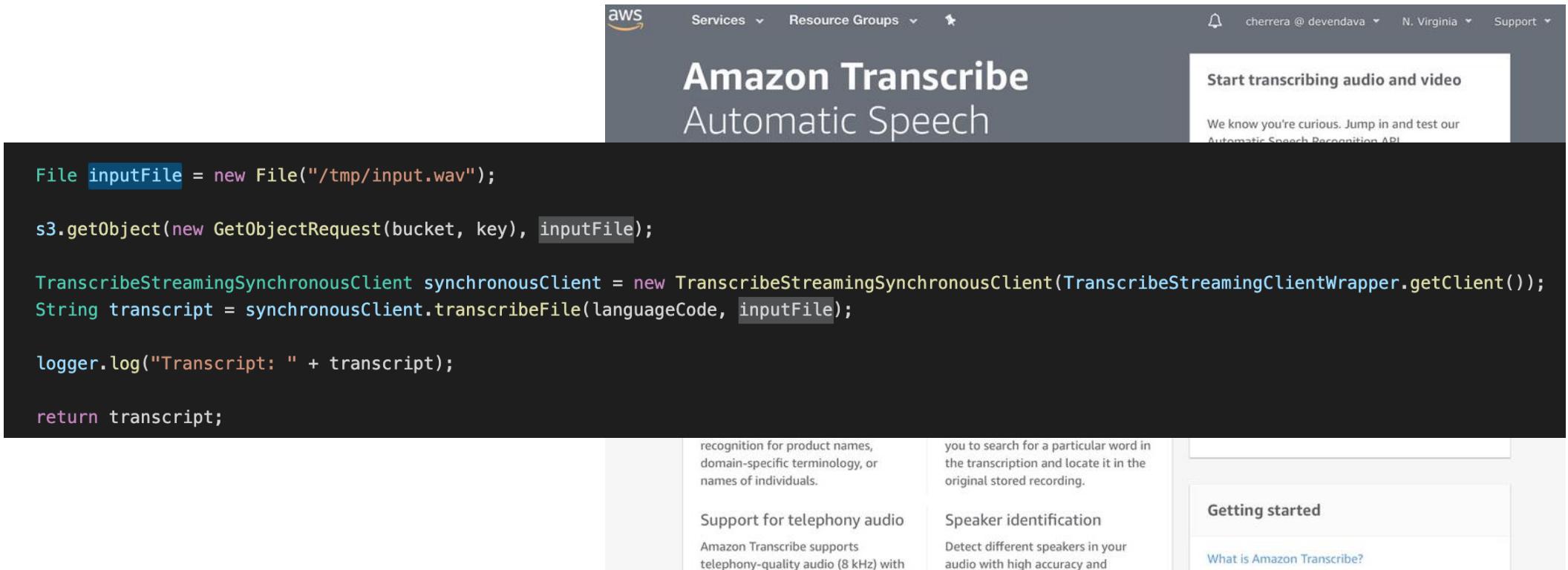
```
s3.upload(
{
  Key: inputKey,
  Body: blob
},
function(err, data) {
if (err) {
  return alert("There was an error uploading your recording: ", err.message);
} else {
  var lambda = new AWS.Lambda({ region: awsRegion, apiVersion: "2015-03-31" });
  var input = {
    FunctionName: lambdaFunction,
    InvocationType: "RequestResponse",
    LogType: "None",
    Payload: JSON.stringify({
      bucket: bucketName,
      key: inputKey,
      sourceLanguage: source_language,
      targetLanguage: target_language
    })
  };

  lambda.invoke(input, function(err, data) {
    if (err) {
      console.log(err);
      alert("There was a problem with Lambda function!!! ");
    } else {
      var resultUrl = data.Payload.replace(/[^"]+/g, "");
      resetView();
      document.getElementById("audio-output").innerHTML =
        '<audio controls autoplay><source src="' +
        resultUrl +
        '" type="audio/mpeg"></audio><br/>';
    }
  });
}
});
```

# STEP BY STEP CONFIG

---

## 5. Convert audio into text with Amazon Transcribe.



The screenshot shows the AWS Lambda function configuration interface. On the left, there is a code editor window containing Java code for transcribing audio files:

```
File inputFile = new File("/tmp/input.wav");

s3.getObject(new GetObjectRequest(bucket, key), inputFile);

TranscribeStreamingSynchronousClient synchronousClient = new TranscribeStreamingSynchronousClient(TranscribeStreamingClientWrapper.getClient());
String transcript = synchronousClient.transcribeFile(languageCode, inputFile);

logger.log("Transcript: " + transcript);

return transcript;
```

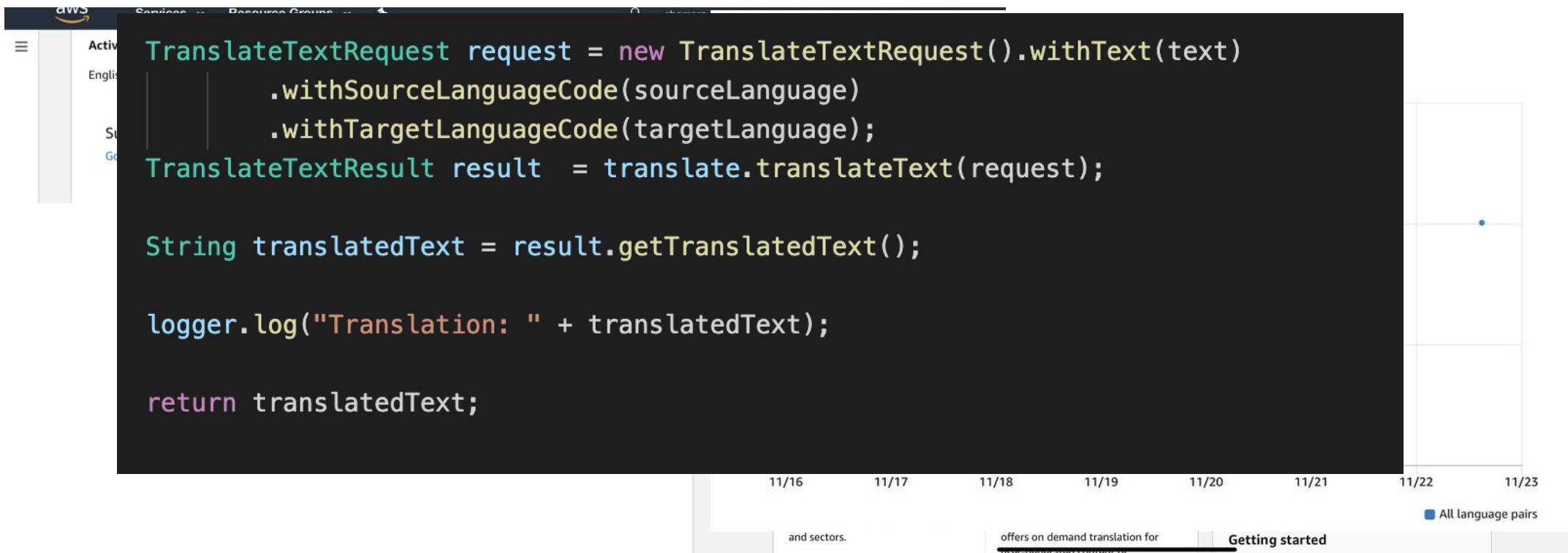
On the right, there are several tabs and sections:

- Services**: A dropdown menu.
- Resource Groups**: A dropdown menu.
- cherrera @ devendava**: User information.
- N. Virginia**: Region selection.
- Support**: Support link.
- Amazon Transcribe Automatic Speech**: Main title.
- Start transcribing audio and video**: Call-to-action button.
- We know you're curious. Jump in and test our Automatic Speech Recognition API**: Test API link.
- Getting started**: Link to get started.
- What is Amazon Transcribe?**: Link to learn more about the service.

# STEP BY STEP CONFIG

---

## 6. Translate the transcribed text from one language to another with **Amazon Translate**.



A screenshot of a code editor displaying Java code for translating text using the Amazon Translate API. The code uses the AWS SDK for Java. It creates a `TranslateTextRequest` object, sets the source language code, target language code, and the text to be translated. It then calls `translate.translateText` to get the result, retrieves the translated text, and logs it. Finally, it returns the translated text.

```
TranslateTextRequest request = new TranslateTextRequest().withText(text)
    .withSourceLanguageCode(sourceLanguage)
    .withTargetLanguageCode(targetLanguage);
TranslateTextResult result = translate.translateText(request);

String translatedText = result.getTranslatedText();

logger.log("Translation: " + translatedText);

return translatedText;
```

The code editor has a dark theme. The Java code is highlighted in various colors (light blue for strings, pink for class names, etc.). Below the code editor is a navigation bar with dates from 11/16 to 11/23. A button labeled "Getting started" is highlighted in blue. At the bottom right, there are logos for "DEWEEK" and "endava".

# STEP BY STEP CONFIG

7.

```
String outputFileName = "/tmp/output.mp3";

SynthesizeSpeechRequest synthesizeSpeechRequest = new SynthesizeSpeechRequest()
    .withOutputFormat(OutputFormat.Mp3)
    .withVoiceId(voiceId)
    .withText(text);

try (FileOutputStream outputStream = new FileOutputStream(new File(outputFileName))) {
    SynthesizeSpeechResult synthesizeSpeechResult = polly.synthesizeSpeech(synthesizeSpeechRequest);
    byte[] buffer = new byte[2 * 1024];
    int readBytes;

    try (InputStream in = synthesizeSpeechResult.getAudioStream()) {
        while ((readBytes = in.read(buffer)) > 0) {
            outputStream.write(buffer, 0, readBytes);
        }
    }

} catch (Exception e) {
    logger.log(e.toString());
}

return outputFileName;
```

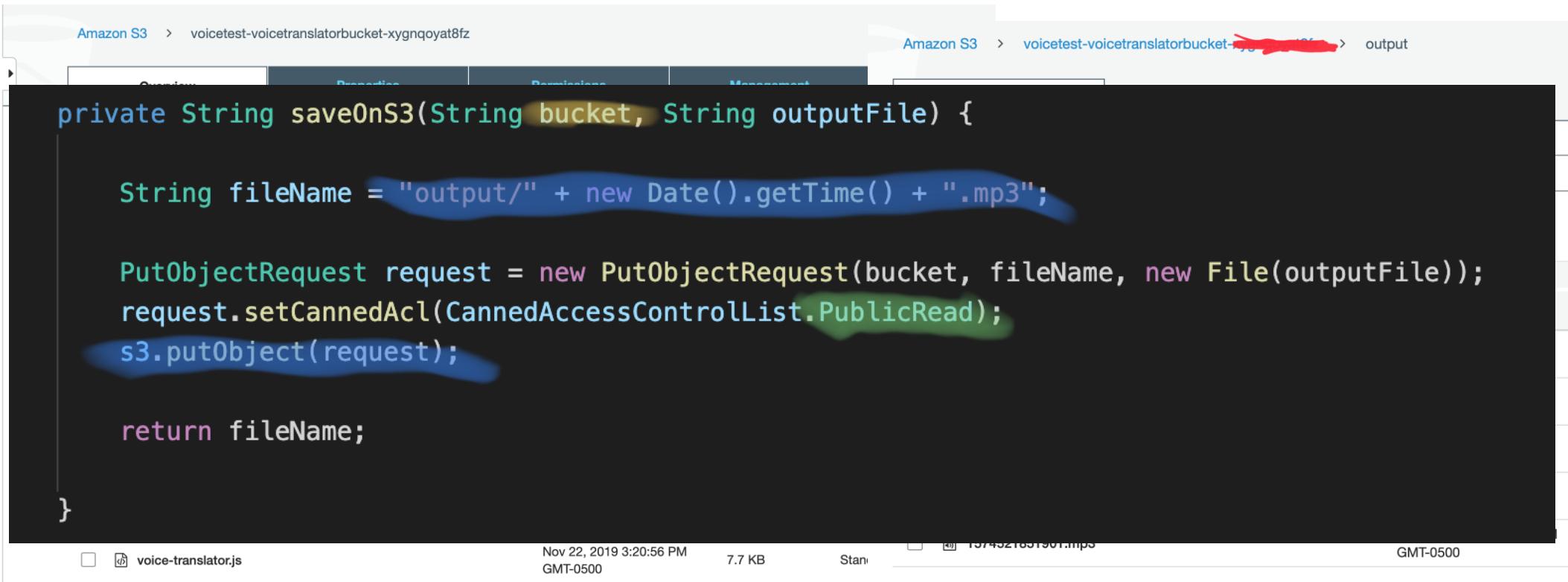
Feedback

eserved. Privacy Policy Terms of Use

# STEP BY STEP CONFIG

---

8. Save the output audio file back to S3, and then return the file name to your page. You could return the audio file itself, but for simplicity, save it on S3 and just return its name.



The screenshot shows two views of the AWS S3 console. The left view shows the root bucket 'voicetest-voicetranslatorbucket-xygnqoyat8fz'. The right view shows a folder named 'output' within the same bucket. A file named '1574521651501.mp3' is listed in the 'output' folder. The code in the foreground is a Java snippet for saving files to S3.

```
private String saveOnS3(String bucket, String outputFile) {  
  
    String fileName = "output/" + new Date().getTime() + ".mp3";  
  
    PutObjectRequest request = new PutObjectRequest(bucket, fileName, new File(outputFile));  
    request.setCannedAcl(CannedAccessControlList.PublicRead);  
    s3.putObject(request);  
  
    return fileName;  
  
}
```

voice-translator.js Nov 22, 2019 3:20:56 PM 7.7 KB Standard 1574521651501.mp3 GMT-0500

# STEP BY STEP CONFIG

---

9.

Automatically play the translated audio to the user.



```
    } else {  
        var resultUrl = data.Payload.replace(/[^"]+/g, "");  
        resetView();  
        document.getElementById("audio-output").innerHTML =  
            '<audio controls autoplay><source src="' +  
            resultUrl +  
            '" type="audio/mpeg"></audio><br/>';  
    }  
}
```

# STEP BY STEP CONFIG

10.

Accelerate the speed of delivering the file with **CloudFront**.

## Edit Distribution

### Distribution Settings

Price Class:  Use All Edge Locations (Best Performance)

AWS WAF Web ACL:  Use Only U.S., Canada and Europe  
Use U.S., Canada, Europe, Asia, Middle East and Afr  
 Use All Edge Locations (Best Performance)

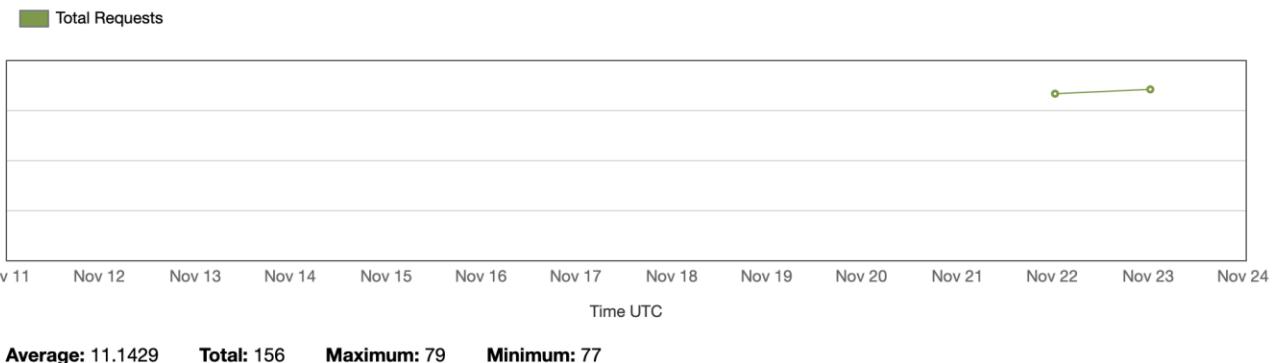
Alternate Domain Names (CNAMEs):

SSL Certificate:  Default CloudFront Certificate (\*.cloudfront.net)

Choose this option if you want your users to use HTTPS or <https://d111111abcdef8.cloudfront.net/logo.jpg>.  
Important: If you choose this option, CloudFront requires tha

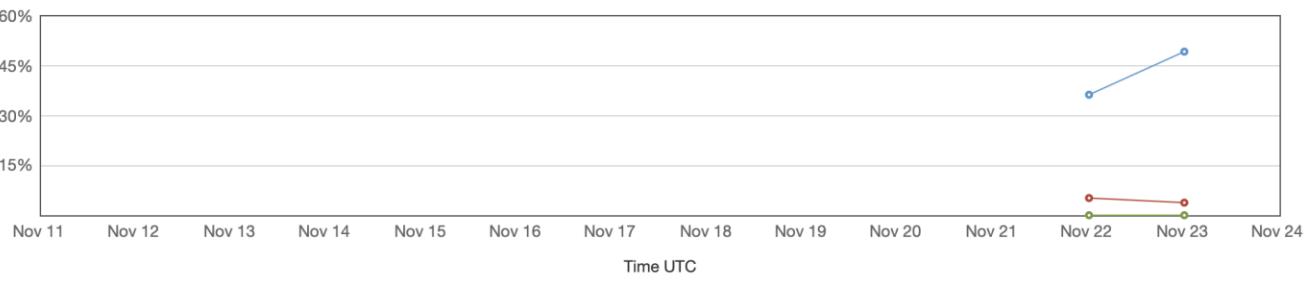
Custom SSL Certificate (example.com):

Choose this option if you want your users to access your cor  
You can use a certificate stored in AWS Certificate Manager i  
(N. Virainia Region). or you can use a certificate stored in IAN



Percentage of Viewer Requests by Result Type [Show Details](#)

Hits  Misses  Errors



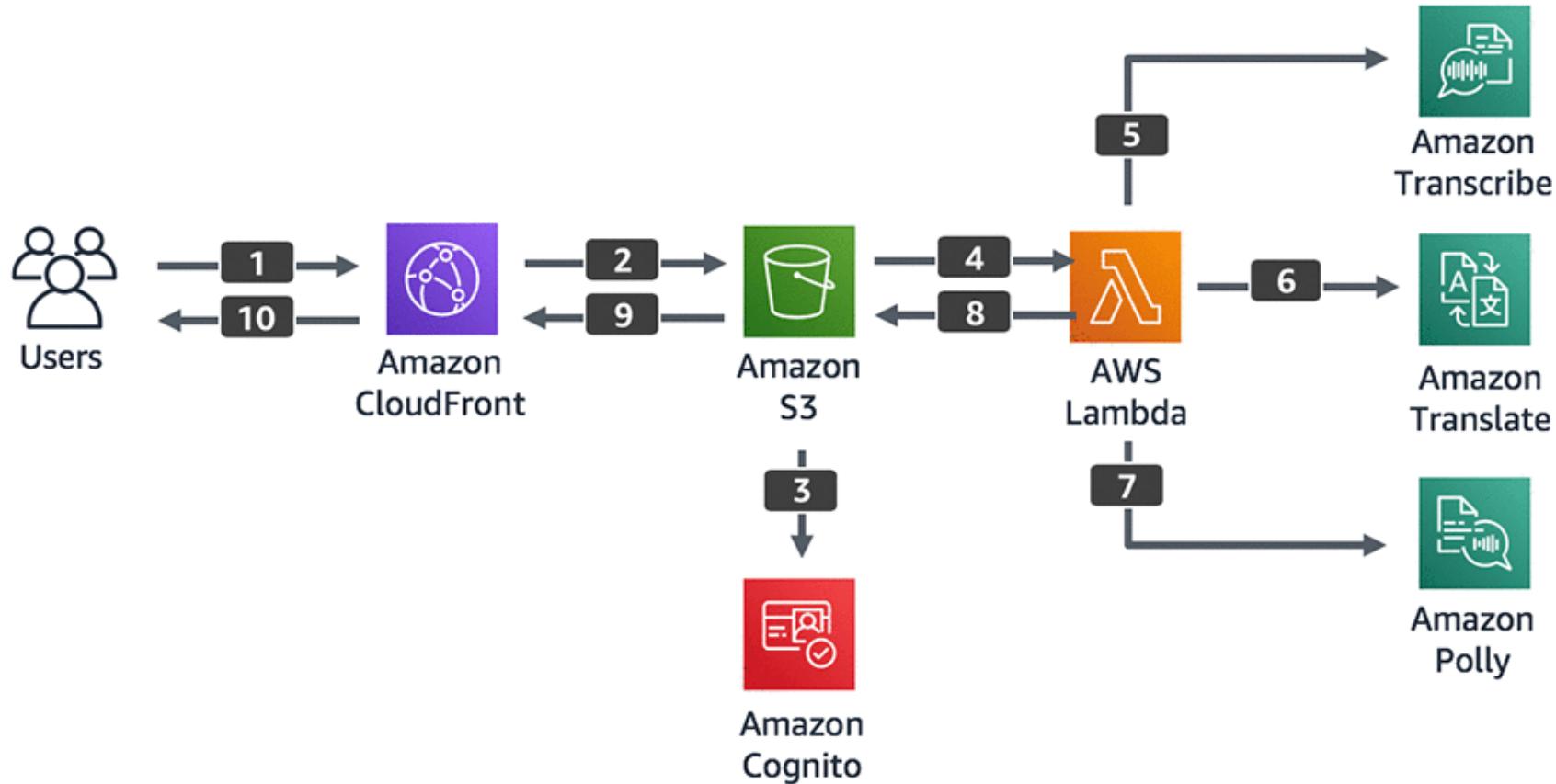
# DOING THIS MANUALLY? ... NONSENSE

---

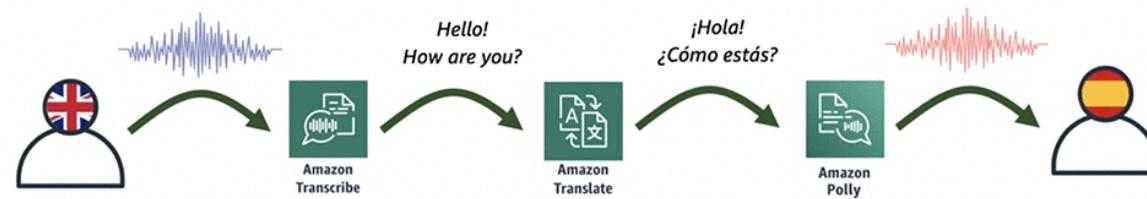
## CLOUDFORMATION/IaC TO THE RESCUE



# FLOW OF SERVICE INTERACTIONS



# DEMO TIME!



<https://translate.carlosaherrera.com>



# SERVERLESS TRANSLATOR DEMO

---



**It's not a bug,  
it's a feature.**

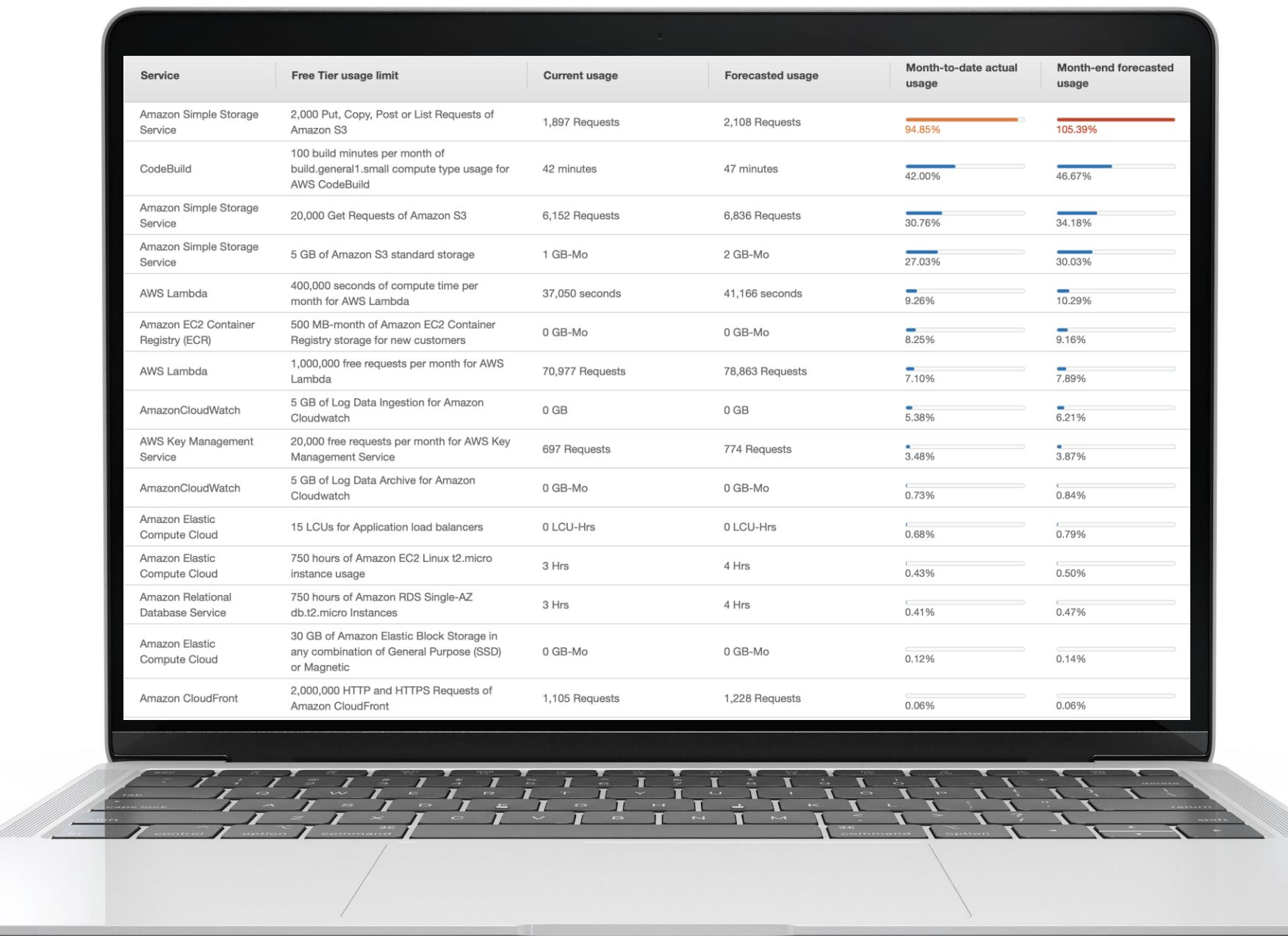
<https://github.com/Carlos4ndresh/voicetranslateapp/issues>

# TRANSLATOR BUDGET

This demo used the following services:

- S3
- Lambda
- Cloudwatch
- KMS
- CloudFront
- Cognito

*Costs, the not so cool side...*



CloudFront		\$0.00
Europe		\$0.00
Amazon CloudFront EU-Requests-Tier1	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Amazon CloudFront EU-Requests-Tier2-HTTPS	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Bandwidth	1.000 Requests	\$0.00
\$0.00 per GB - data transfer out under the global monthly free tier	54.000 Requests	\$0.00
South America		\$0.00
Amazon CloudFront SA-Requests-Tier1	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Amazon CloudFront SA-Requests-Tier2-HTTPS	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Bandwidth	8.000 Requests	\$0.00
\$0.00 per GB - data transfer out under the global monthly free tier	744.000 Requests	\$0.00
United States		\$0.00
Amazon CloudFront US-Requests-Tier1	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Amazon CloudFront US-Requests-Tier2-HTTPS	\$0.00 per request - HTTP or HTTPS under the global monthly free tier	\$0.00
Bandwidth	5.000 Requests	\$0.00
\$0.00 per GB - data transfer out under the global monthly free tier	293.000 Requests	\$0.00
Lambda		\$0.00
US East (N. Virginia)		\$0.00
AWS Lambda Lambda-GB-Second	AWS Lambda - Compute Free Tier - 400,000 GB-Seconds - US East (Northern Virginia)	\$0.00
AWS Lambda Request	AWS Lambda - Requests Free Tier - 1,000,000 Requests - US East (Northern Virginia)	\$0.00
US West (Oregon)		\$0.00



You can run this example paying USD \$0 using **AWS FreeTier (12 months)**

Check each service **FreeTier limit**, not everything that shines is gold!

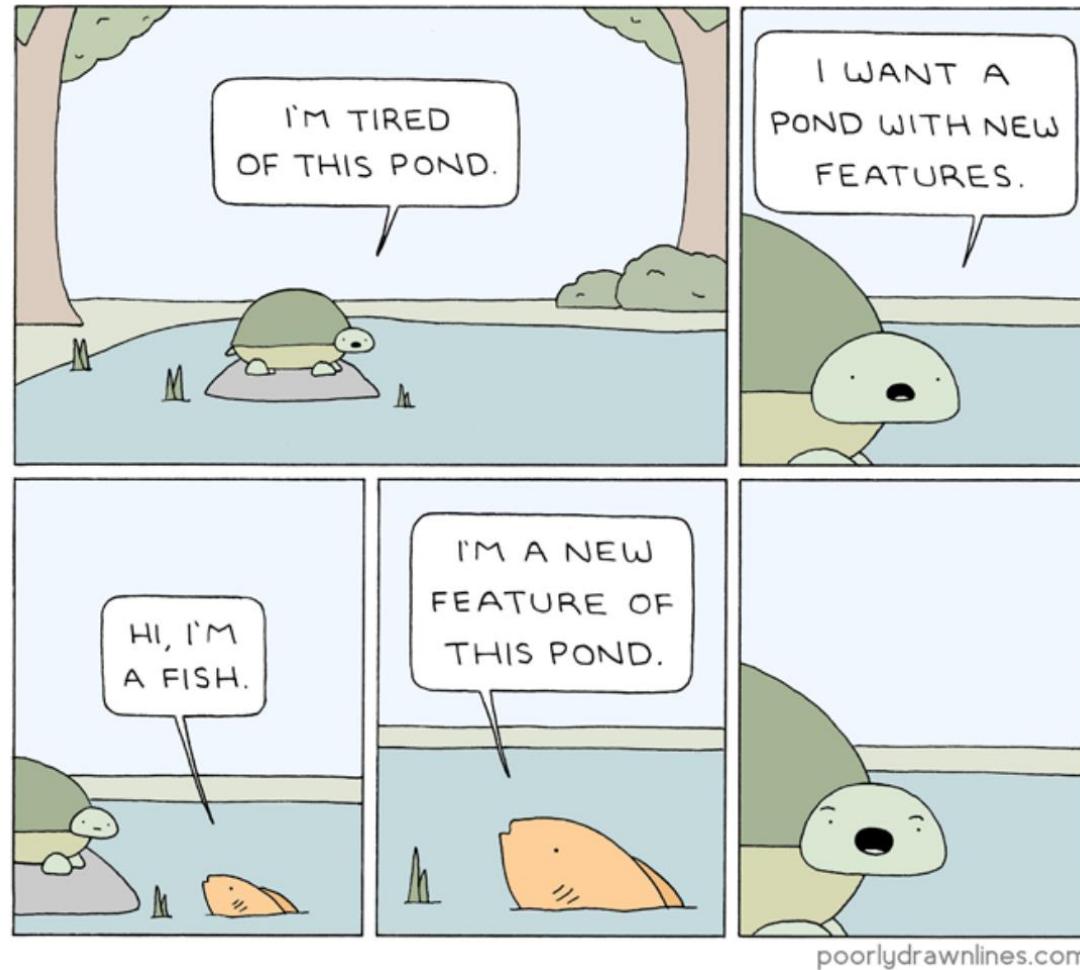
There are services that are not FreeTier included but, the trend it's that all will be in the future...

▼ Simple Storage Service		\$0.00
▼ US East (N. Virginia)		\$0.00
Amazon Simple Storage Service Requests-Tier1		\$0.00
\$0.00 per request - PUT, COPY, POST, or LIST requests under the monthly global free tier	636.000 Requests	\$0.00
Amazon Simple Storage Service Requests-Tier2		\$0.00
\$0.00 per request - GET and all other requests under the monthly global free tier	3,593.000 Requests	\$0.00
Amazon Simple Storage Service TimedStorage-ByteHrs		\$0.00
\$0.000 per GB - storage under the monthly global free tier	0.009 GB-Mo	\$0.00
▼ Transcribe		\$0.00
▼ US East (N. Virginia)		\$0.00
Amazon Transcribe StreamingAudio		\$0.00
Processed streaming transcription request in us-east-1 - free tier	810.000 second	\$0.00
▼ Translate		\$0.00
▼ US East (N. Virginia)		\$0.00
Amazon Translate TranslateText		\$0.00
Processed translation request in US East (N. Virginia) - free-tier	1,398.000 Character	\$0.00
Processed translation request in US East (N. Virginia) - free-tier	276.000 Character	\$0.00
▼ CloudWatch		\$0.00
▼ US East (N. Virginia)		\$0.00
Amazon CloudWatch		\$0.00
\$0.00 per request - first 1,000,000 requests	203.000 Requests	\$0.00
AmazonCloudWatch PutLogEvents		\$0.00
First 5GB per month of log data ingested is free.	0.000131 GB	\$0.00
▼ Polly		\$0.00
▼ US East (N. Virginia)		\$0.00
Amazon Polly USE1-SynthesizeSpeech-Characters		\$0.00
First 5,000,000 characters (across regions) per month are free	1,896.000 Characters	\$0.00
Amazon Polly USE1-SynthesizeSpeechNeural-Characters		\$0.00
First 1,000,000 characters (across regions) per month are free for Neural TTS	148.000 Characters	\$0.00

...even if you're already out of the FreeTier, this could cost as low as US \$1

So it's almost...





# THANK YOU!

**CATHERINE CRUZ**

DEVOPS ENGINEER

catherine.cruz@endava.com

**CARLOS HERRERA**

DEVOPS ENGINEER

carlos.herrera@endava.com