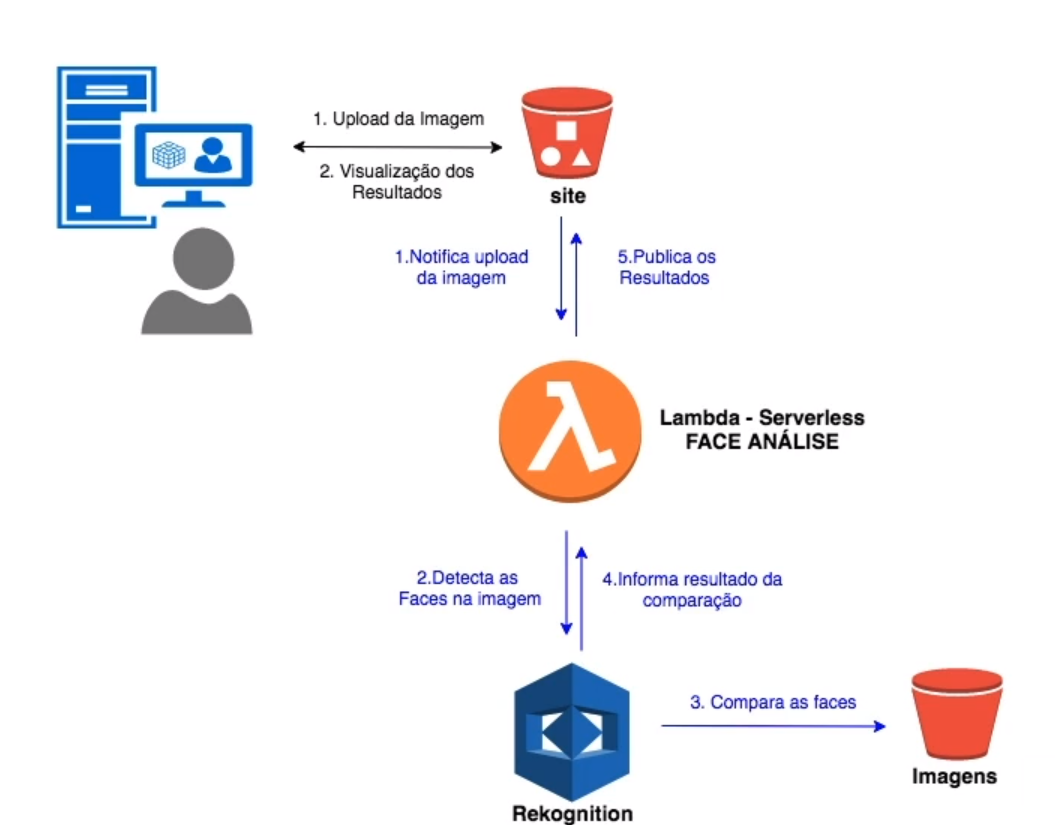
**Face-Analise**

Este projeto corresponde a função lambda que visa comparar uma imagem atráves da API de reconhecimento facial da AWS (API boto3 Rekognition) com as imagens que está no servidor de armazenamento de imagens. Todas as vezes que subir uma imagem com o nome \_analise.jpg no servidor de imagens então a função lambda irá fazer uma série de rotina com a seguinte regra:

* No arquivo \_analise.jpg se a imagem comparada tiver relevancia com as imagens existente no servidor de armazenamento de imagens então é exibido no site estático. O site estático irá exibir uma lista de imagens, nomes e porcentagem de relevancia.
* O site estático fica em outro servidor de armazenamento, se não encontrar nenhum tipo de relevancia então não é mostrado na lista.

O desenho da solução pode ser visualizado abaixo:



Para usar o projeto face-analise é preciso verificar o pré-requisitos.

**Pré requisitos**

* Instalar a interface de linha de comandos da aws

<https://aws.amazon.com/pt/cli/>

**Função Lambda**

É literalmente uma função que pode ser construída em qualquer linguagem de plataforma baixa como serviço na AWS.

As funções lambdas foca na aplicação e não na infra estrutura na Amazon e tem as seguintes vantagens:

* Computação sem o gerenciamento do servidor - foco no serviço;
* Escalabilidade - Combinar com outros serviços e escalar de maneira facil sem precisar de preocupar com os recursos;
* Disponibilidade;

Algumas observações podem ser visualizadas abaixo:

1. É preciso configurar a memória da função o mínimo é 128MB;
2. É preciso configurar o tempo máximo da execução da função, ao exceder o tempo a função é encerrada;

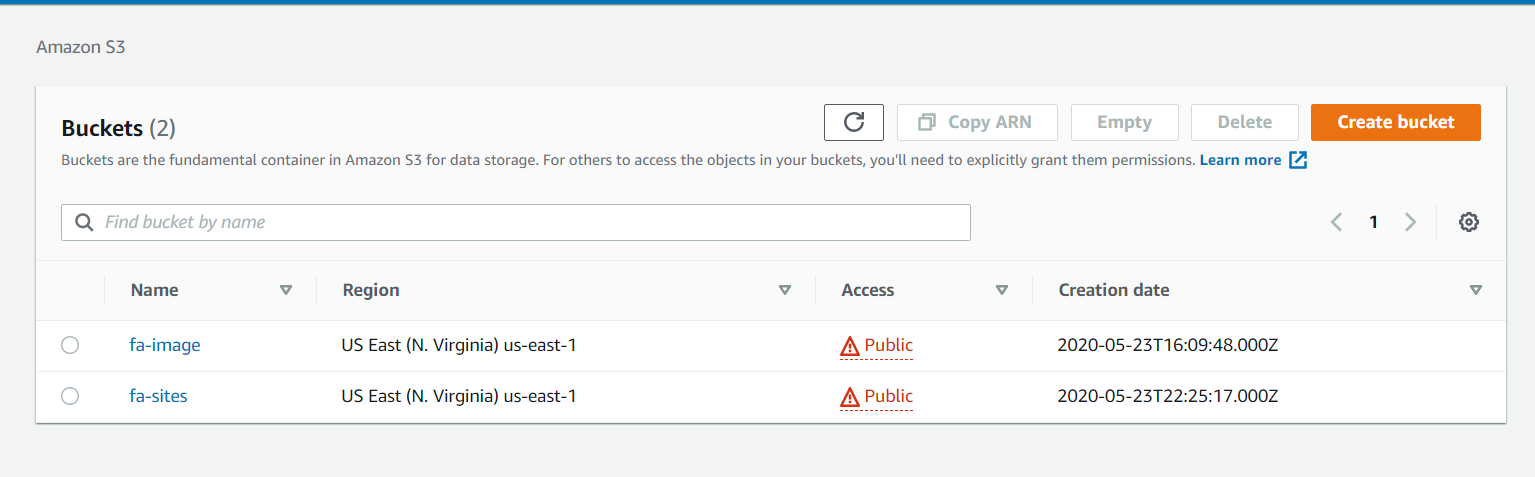
Quando uma função lambda é criada é necessario colocar uma trigger para que essa função seja disparada.

A trigger pode ser configurada por várias formas entre elas Pode configurar um trigger para pelo serviço de *CloudWatch* usando o cron expression UTC (Brasil as horas está -3 ou seja se por exemplo é 12:00 horário de Brasília então a horas utc é 15:00). Com esse serviço a função pode ser disparada de acordo com a configuração do cron que corresponde o tempo para ser disparada.

Uma outra maneira de disparar a função é inserir triggers prontas como a S3.

**Entendendo o Projeto**

A primeira coisa a fazer é criar o serviço de armazenamento S3 na AWS chamado *fa-image,* isto pode ser realizado entrando no serviço S3 da AWS e criando o serviço conforme a imagem. E logo em seguida criar o serviço fa-sites conforme a imagem abaixo:



Em fa-sites inserir o seguinte trecho de código em Permissions -> Bucket Policy:

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "PublicReadGetObject",

"Effect": "Allow",

"Principal": "\*",

"Action": "s3:GetObject",

"Resource": "arn:aws:s3:::fa-sites/\*"

}

]

}

Inserir também a permissão do CORS

<?xml version="1.0" encoding="UTF-8"?>

<CORSConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">

<CORSRule>

<AllowedOrigin>\*</AllowedOrigin>

<AllowedMethod>GET</AllowedMethod>

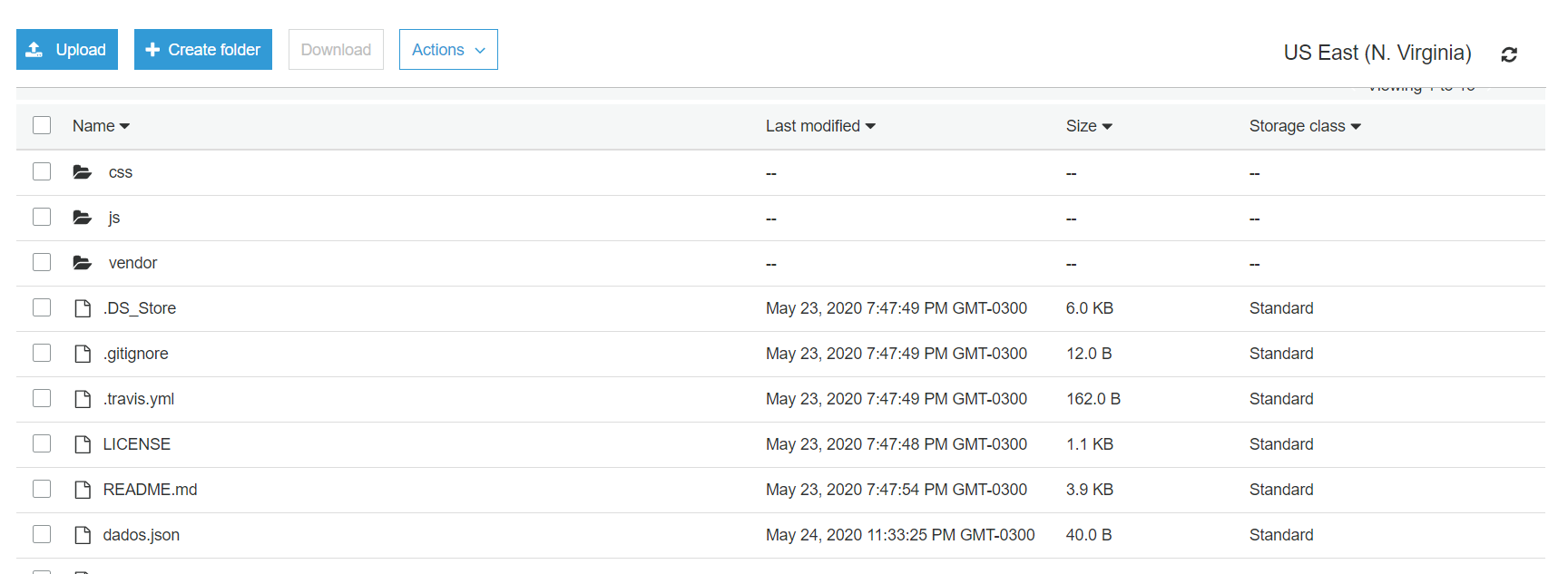
<MaxAgeSeconds>3000</MaxAgeSeconds>

<AllowedHeader>\*</AllowedHeader>

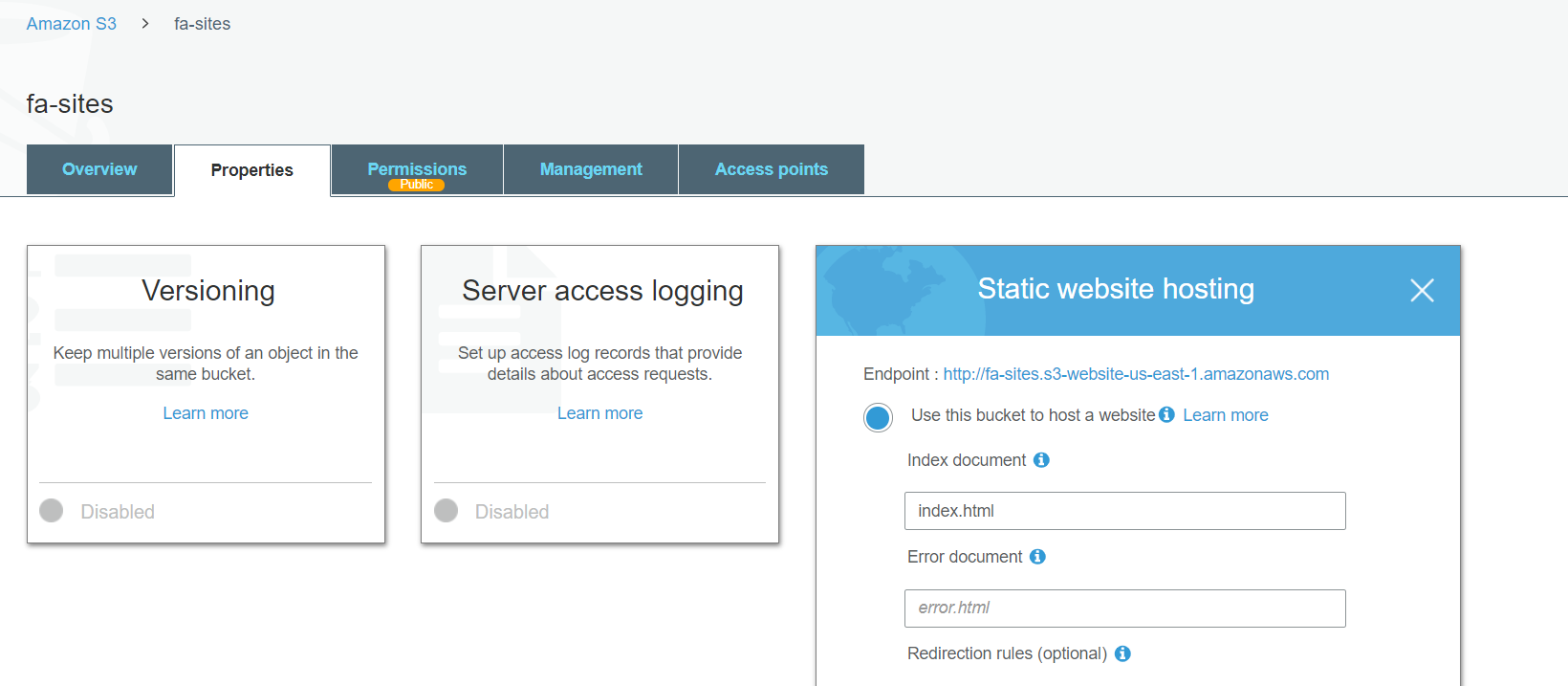
</CORSRule>

</CORSConfiguration>

Para que essas imagens estáticas possam está públicas. Em seguida fazer drag end drop do projeto estático que está no diretório fa-site para S3 fa-sites da AWS conforme a imagem abaixo:



Em fa-sites entrar no recurso de properties para inserir a configuração de um site stático.



Em seguida em fa-image configurar a permissão em Permissions -> Bucket Policy:

Com o seguinte trecho:

|  |
| --- |
| {  "Version": "2012-10-17",  "Id": "PolicyForCloudFrontPrivateContent",  "Statement": [  {  "Effect": "Allow",  "Principal": "\*",  "Action": "s3:GetObject",  "Resource": "arn:aws:s3:::fa-image/\*",  "Condition": {  "StringLike": {  "aws:Referer": "http://fa-sites.s3-website-us-east-1.amazonaws.com/\*"  }  }  }  ]  } |

Permitir que apenas o recurso fa-sites tenha acesso ao fa-image

Com estas configurações faça o seguinte comando para inserir as imagens dentro do diretório images:

aws s3 sync . s3://fa-image

Suba a imagem específica JPG com alguns rostos parecido com o diretório imagens no arquivo chamado \_analise.jpg com seguinte comando:

aws s3 cp \_analise.jpg s3://fa-image

Rode o comando index.py para que as imagens possa ser indexada pela API Rekognition.

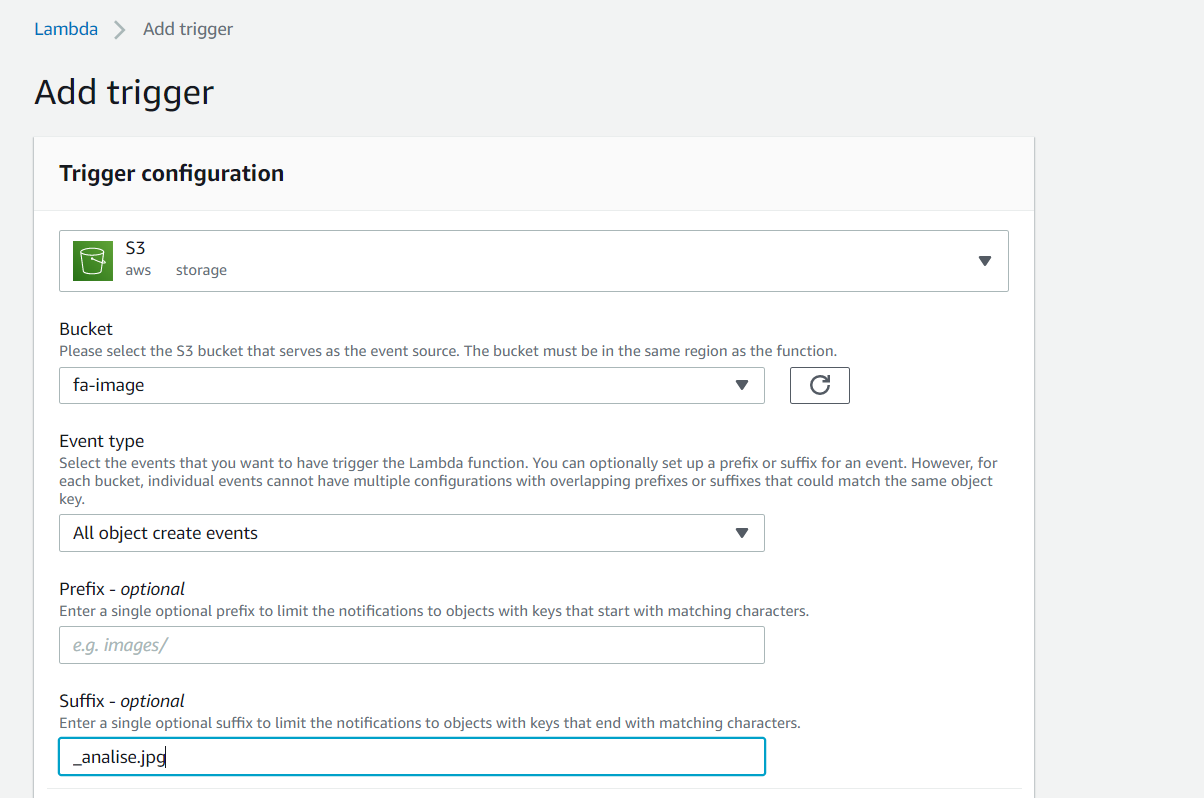
Use o comando aws rekognition list-faces --collection-id faces | findstr ExternalImageId ou

aws rekognition list-faces --collection-id faces | grep ExternalImageId para verificar as imagens indexadas.

O arquivo faceanalise.py é um arquivo em python, para se tornar uma função lambda faça o zip do arquivo faceanalise.py em seguida crie a função lambda no serviço da AWS como o nome faceAnalise em seguinda faça o seguinte comando no terminal:

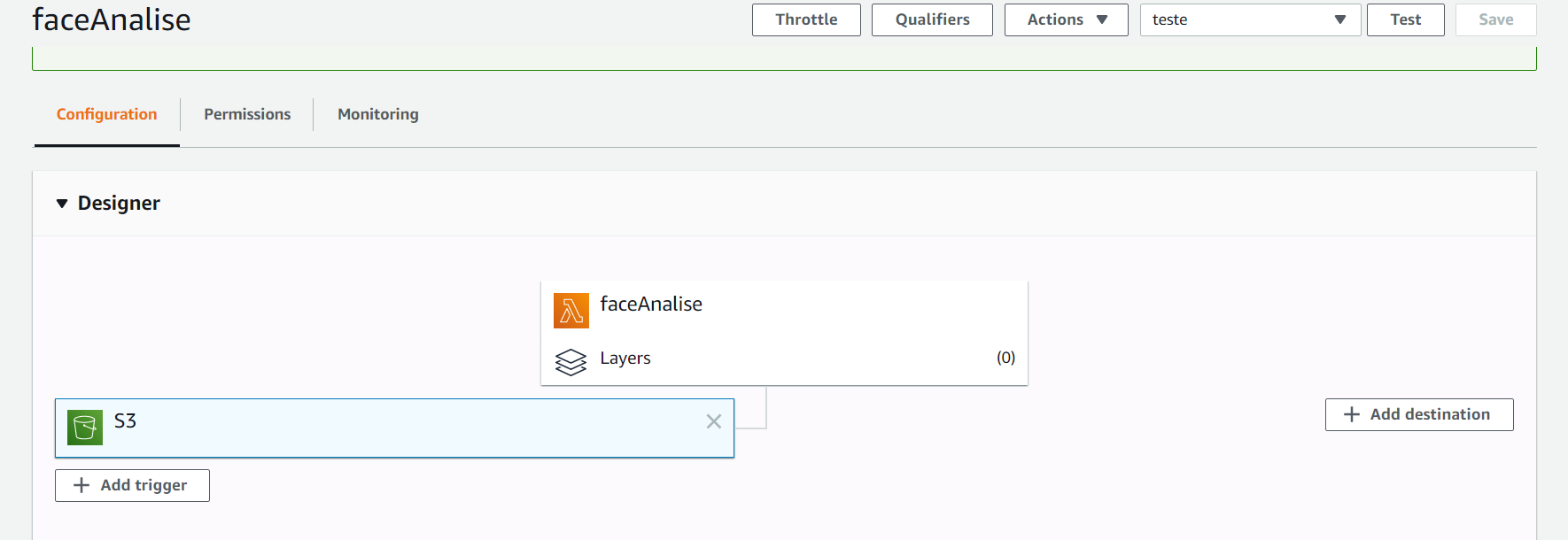
aws lambda update-function-code --function-name faceAnalise --zip-file fileb://faceanalise.zip

Insira a trigger para disparar a função lambda faceAnalise.



Exemplo da inserção da trigger s3.

Ao inserir a imagem \_analise.jpg no serviço de armazenamento fa-image então o evento trigger é disparado para que a função faceAnalise seja executada.



Versionamento e alias podem ser criados com os seguintes comandos:

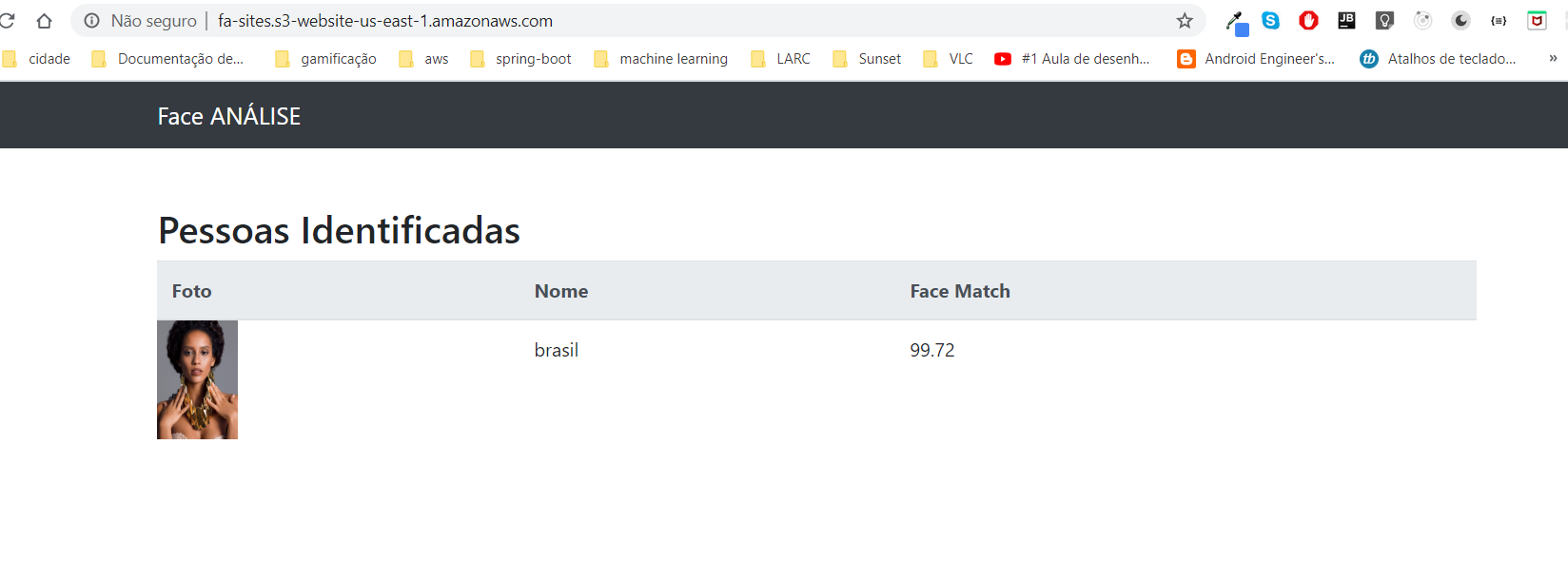
Criar versionamento no aws lambda

aws lambda publish-version --function-name <nome da funçao lambda>` Exemplo: aws lambda publish-version --function-name faceAnalise

Criar um alias no aws lambda

aws lambda create-alias --function-name faceAnalise --function-version 1 --name PROD

Ao acessar o link do serviço de armazenamento fa-sites irá exibir que foi reconhecida na imagem \_analise.jpg.



**Referencia bibliográfica**

<https://docs.aws.amazon.com/cli/latest/reference/lambda/>