

Practica 2 -G18

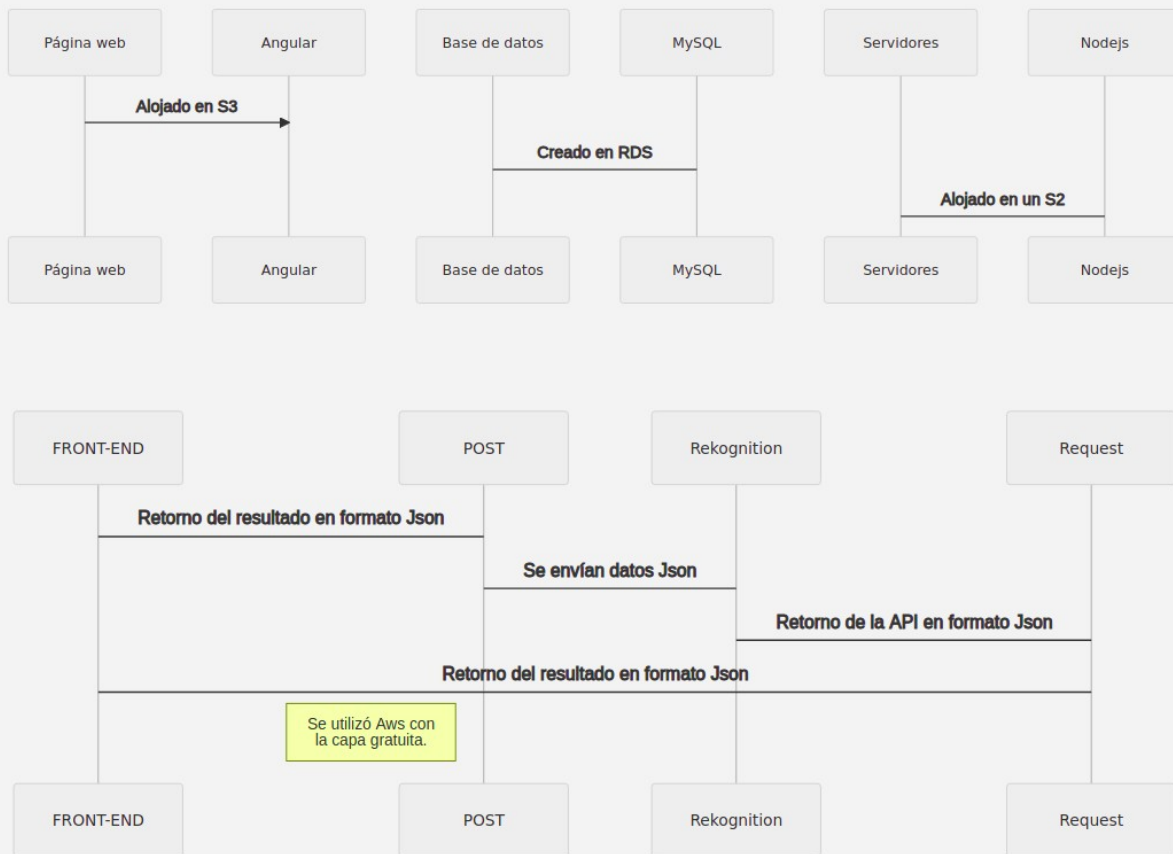
Datos

201503746 - RAYMUNDO ALEXANDER IXVALAN PACHECO

201503577 - HENRY FRANCISCO LEÓN HERNÁNDEZ

Arquitectura

La arquitectura que se utilizó para realizar la presente práctica fue la siguiente:



Usuario de IAM

Para el s3: semi1-practica1-S3FullAccess

Resumen

Eliminar al usuario

ARN de usuario

arn:aws:iam::285662992045:user/semi1-practica1-S3FullAccess

Ruta

/

Hora de creación

2021-03-06 14:53 CST

Permisos

Grupos

Etiquetas

Credenciales de seguridad

Access Advisor

Políticas de permisos (1 política aplicada)

Añadir permisos

Añadir una política insertada

Nombre de la política	Tipo de política
Asociada directamente	
AmazonS3FullAccess	Política administrada por AWS
Límite de permisos (no definido)	

Para el Ec2 de nodejs

Usuarios > Administrador_201503577

Resumen

Eliminar al usuario

ARN de usuario

arn:aws:iam::285662992045:user/Administrador_201503577

Ruta

/

Hora de creación

2021-02-12 16:45 CST

Permisos

Grupos (1)

Etiquetas

Credenciales de seguridad

Access Advisor


Políticas de permisos (2 políticas aplicadas)

Añadir permisos

Añadir una política insertada

Nombre de la política	Tipo de política
Asociada directamente	
IAMUserChangePassword	Política administrada por AWS
Asociados desde el grupo	
AdministratorAccess	Política administrada por AWS del grupo Admin
Límite de permisos (no definido)	


Para la traducción de textos:

ARN de usuario	arn:aws:iam::285662992045:user/semi1-Translate 
Ruta	/
Hora de creación	2021-03-31 18:08 CST

Permisos Grupos Etiquetas Credenciales de seguridad Access Advisor

▼ Políticas de permisos (1 política aplicada)

Añadir permisos

Nombre de la política ▼
Asociada directamente
▶  TranslateFullAccess

Para el uso de rekognition:

ARN de usuario	arn:aws:iam::285662992045:user/semi1-tarea3-rekognition 
Ruta	/
Hora de creación	2021-03-04 03:54 CST

Permisos Grupos Etiquetas Credenciales de seguridad Access Advisor

▼ Políticas de permisos (1 política aplicada)

Añadir permisos

Nombre de la política ▼
Asociada directamente
▶  AmazonRekognitionFullAccess

3 Funcionalidades del Chatbot.

Función 1

Reservación o contratación de los servicios de seguridad (este servicio servirá para verificación de Login con análisis facial)

Función 2

Cancelación del servicio de seguridad.

Función 3

Mostrará la información de lo que trata el servicio de seguridad.

Funciones de Amazon Rekognition implementadas

compareFaces

Función para comparar una captura o foto tomada por alguna cámara que servirá para comparar si el usuario existe dentro del sistema.

```
client.compareFaces(params, function (err, response) {
  if (err) {
    console.log(err, err.stack);
  } else {
    response.FaceMatches.forEach(data2 => {
      let position = data2.Face.BoundingBox
      let similarity = data2.Similarity
      console.log(`The face at: ${position.Left}, ${position.Top} matches with ${similarity} % confidence`)

      const paramsDelete = {
        Bucket: bucket,
        Key: filepath
      };

      if (similarity >= 75) {
        console.log("La captura coincide con la foto actual");
        res.status(200).json(data[i]);
      }
      else {
        console.log("La captura no coincide con la foto actual");
        res.json({ mensaje: 0 });
      }
    })
    s3.deleteObject(paramsDelete, function (err, data) {
      if (err) {
        console.log(err, err.stack); // an error occurred
      }
      else {
        console.log(data);           // successful response
      }
    });
  } // for response.faceDetails
} // if
});
```

detectLabels

Función para obtener todos los escenarios de una foto y así poder asociar la foto a los distintos álbumes detectados.

```
//Detectamos los labels para crear los albumes de la foto
client.detectLabels(params, function (err, response) {
  if (err) {
    console.log(err, err.stack);
    res.json({ mensaje: "Error, al obtener los álbumes", success: false })
  } else {
    //Recorremos cada label
    response.Labels.forEach(data => {
      //console.log(` Name:      ${data.Name}`);
      // Traducimos cada label, Si se tarda mucho. Comentar desde aquí
      let text = data.Name
```

detectFaces

Función para obtener los rasgos o gestos de una foto de perfil

```
client.detectFaces(params, function (err, response) {
  var data_response = {}
  if (err) {
    console.log(err, err.stack);
    res.json({ mensaje: "Error al analizar la foto de perfil", success: false })
  } else {
    response.FaceDetails.forEach(data => {
      let low = data.AgeRange.Low
      var vars = []
      let high = data.AgeRange.High
      console.log(`The detected face is between: ${low} and ${high} years old`)
      console.log("Estos son los atributos que tenes acceder y mostrar papu:")
      console.log(` Age.Range.Low:      ${data.AgeRange.Low} - ${data.AgeRange.High}`)
      //data_response.ageRange = `${data.AgeRange.Low} - ${data.AgeRange.High}`
      var it1 = {'ageRange': `${data.AgeRange.Low} - ${data.AgeRange.High}`}
      console.log(` Gender.Confidence:    ${data.Gender.Value}`)
      console.log(` Gender.Confidence:    ${data.Gender.Confidence}`)
      //data_response.Gender = data.Gender.Value
      var it2 = {'Gender': data.Gender.Value}
      vars.push(it1)
      vars.push(it2)
      if (data.Smile.Value) {
        console.log(` Smile.Confidence:      ${data.Smile.Confidence}`)
        //data_response.Smile = "True"
        var it = {'Smile': "True"}
        vars.push(it)
      }
    })
  }
})
```

detectText

Función para extraer textos detectados en una imagen.

```
client.detectText(params, function (err, data) {  
  if (err) { res.json({ mensaje: "Error" }) }  
  else {  
    console.log(data);  
    var data_response = {}  
    var Texts = []  
    data.TextDetections.forEach(response => {  
      var data = { "Text": response.DetectedText }  
      Texts.push(data);  
      console.log(response.DetectedText);  
    });  
    data_response.Texts = Texts;  
    console.log(data_response)  
    res.json({ Texts });  
  }  
});
```