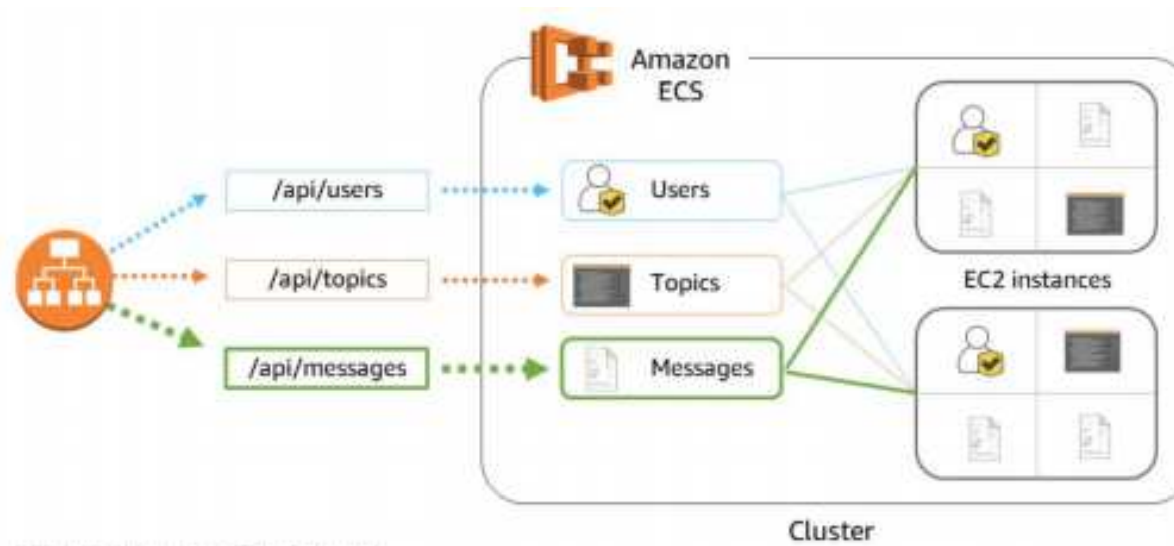


3.8

Microservicios ECS y Serverless lambda

ECS: Elastic Container service

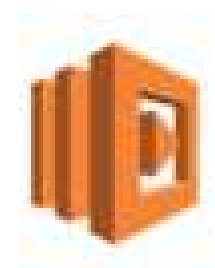
- AWS services for Docker ECS



Serverless: lambda

Crear y ejecutar aplicaciones y servicios sin gestionar servidores

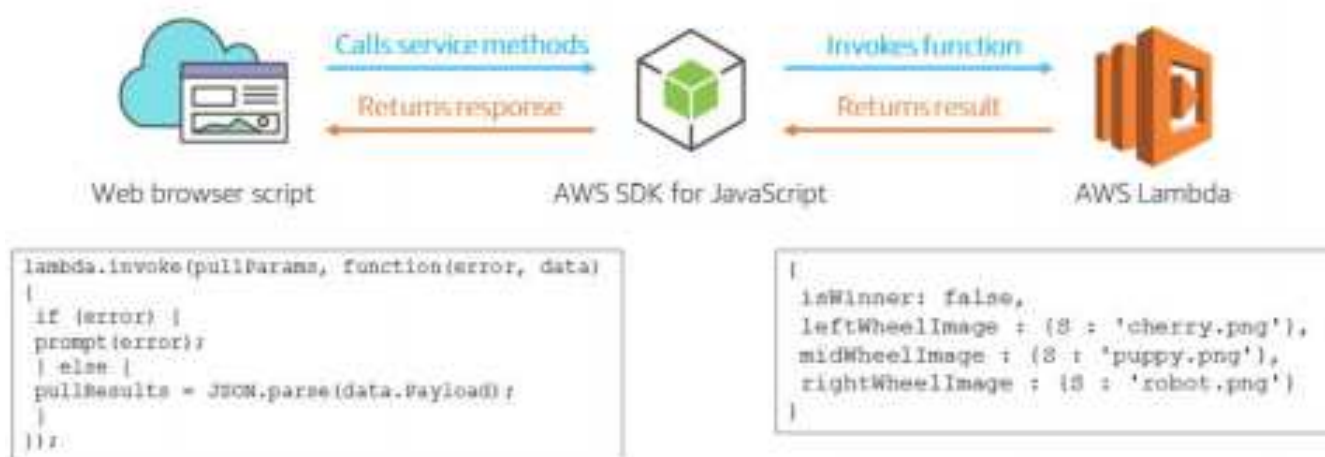
- Net Core 1.0 , Noide.js , Java , Python



AWS
Lambda

Serverless: lambda

Ejemplo de calculo de elemento aleatorio para un juego



Serverless: lambda

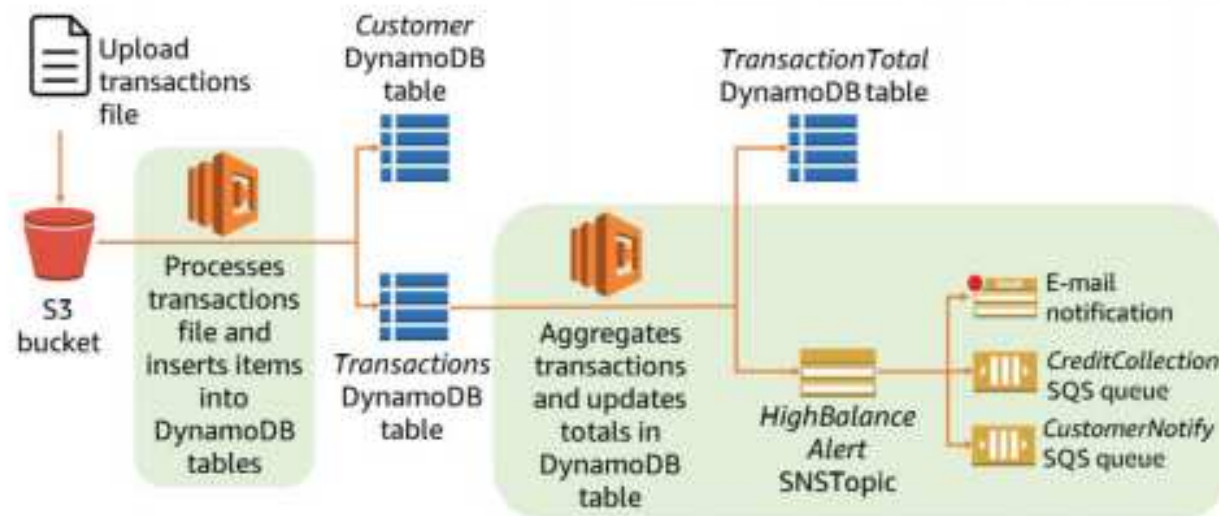
Los servicios pueden lanzarse al ocurrir eventos de recursos AWS



Serverless: lambda

Ejemplo: Realizar actualizaciones en una base de datos al modificar un fichero de un Bucket S3 y contabilizar las transacciones de una BBDD enviando un email y actualizando el contador en una base de datos

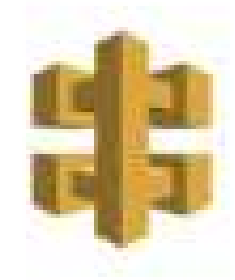
- 1 lambda para detectar cambios en S3 y actualizar BBDD
- 1 lambda para detectar cambios en BBDD y enviar notificaciones



Serverless: lambda

API Gateway

- Puede enviar trabajo hacia
 - Instancias EC2
 - AWS Lambda
 - Cualquier servicio WEB
- Servicios extra de seguridad



API
Gateway



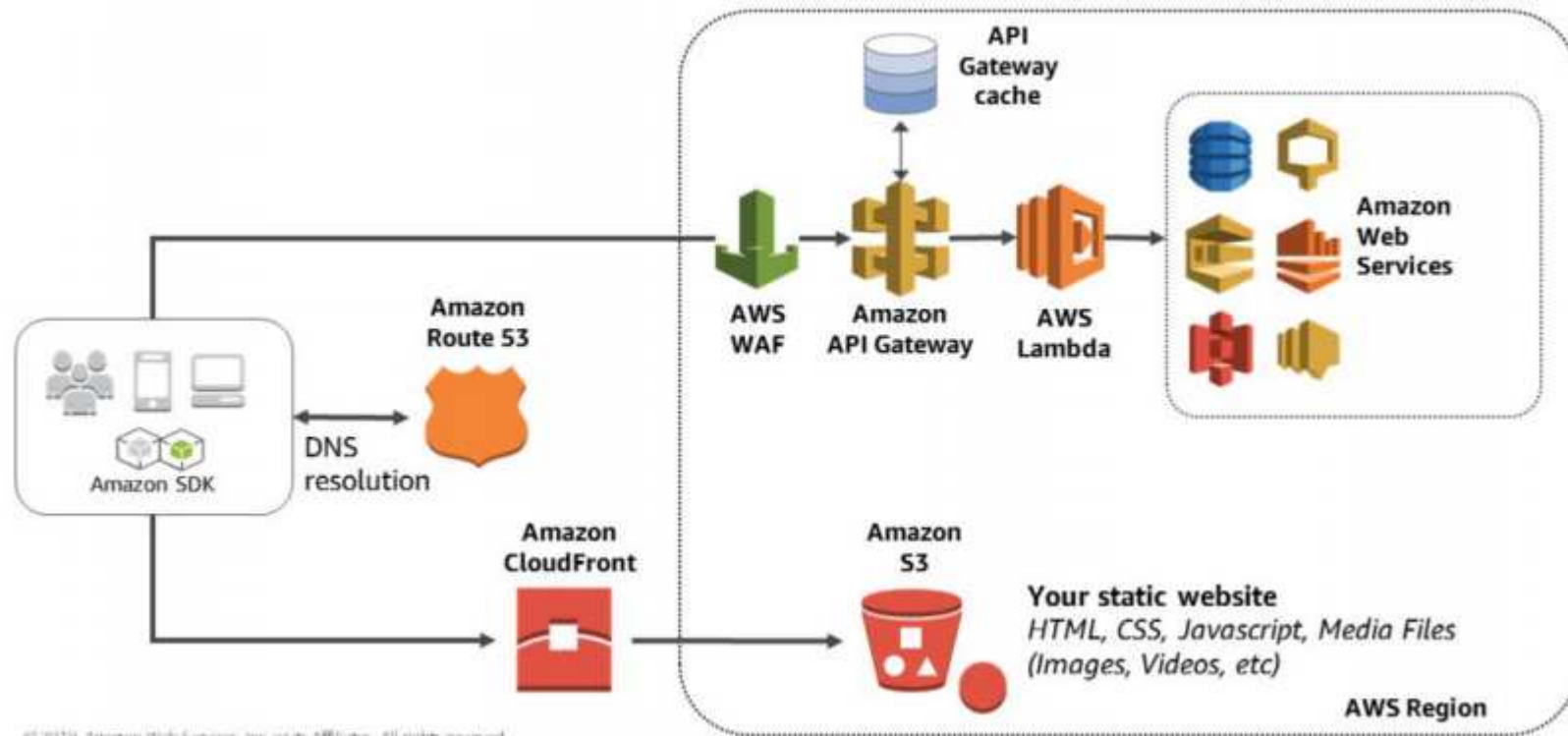
Prevents exposing
endpoints



Protection from
DDoS and
injection attacks

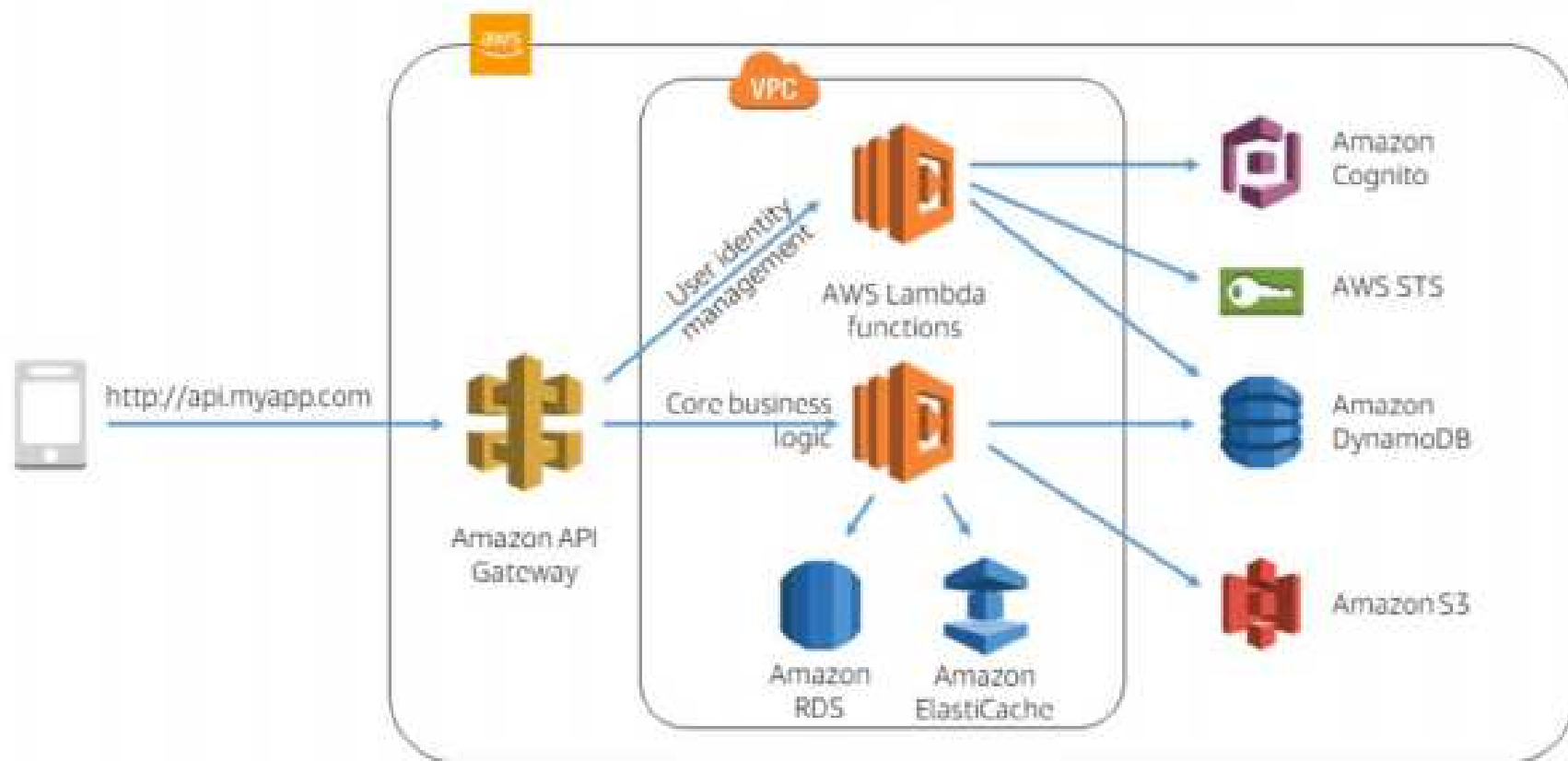
Serverless: lambda

Arquitectura genérica Serverless



Serverless: lambda

Arquitectura genérica Serverless para aplicaciones móviles

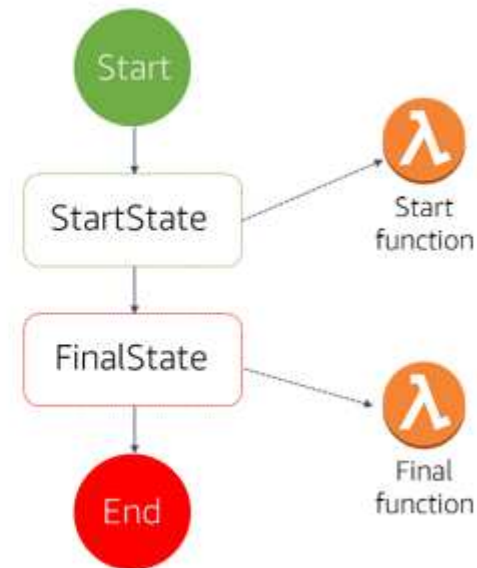


Serverless: lambda

Orquestación de servicios Lambda: AWS Step functions

- Permite definir una maquina de estado que coordina llamadas a los diferente Lambda

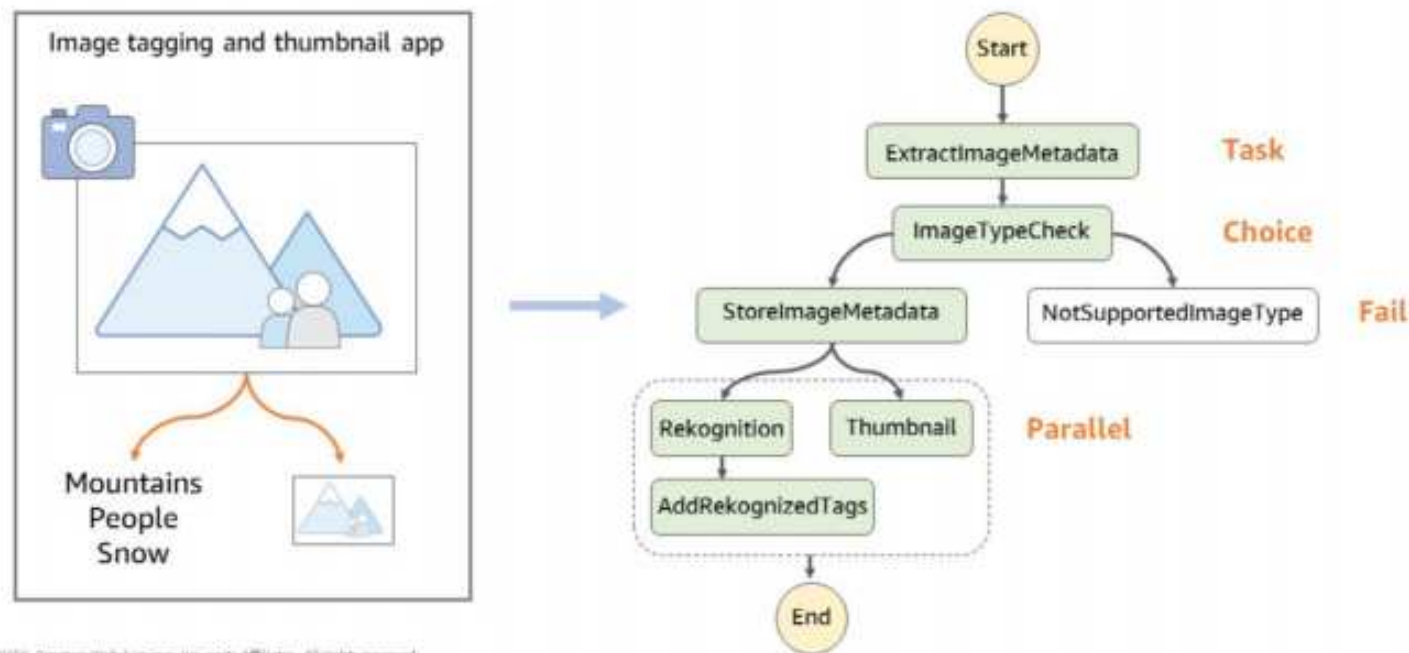
```
{
  "Comment": "An example of the ASL.",
  "StartAt": "StartState",
  "States": {
    "StartState": {
      "Type": "Task",
      "Resource": "arn:aws:lambda:us-east...,
      "Next": "FinalState"
    }
    "FinalState": {
      "Type": "Task",
      "Resource": "arn:aws:lambda:us-east...,
      "End": true
    }
  }
}
```



Serverless: lambda

Orquestación de servicios Lambda: AWS Step functions

- EJEMPLO



Serverless: lambda

Orquestación de servicios Lambda: AWS Step functions

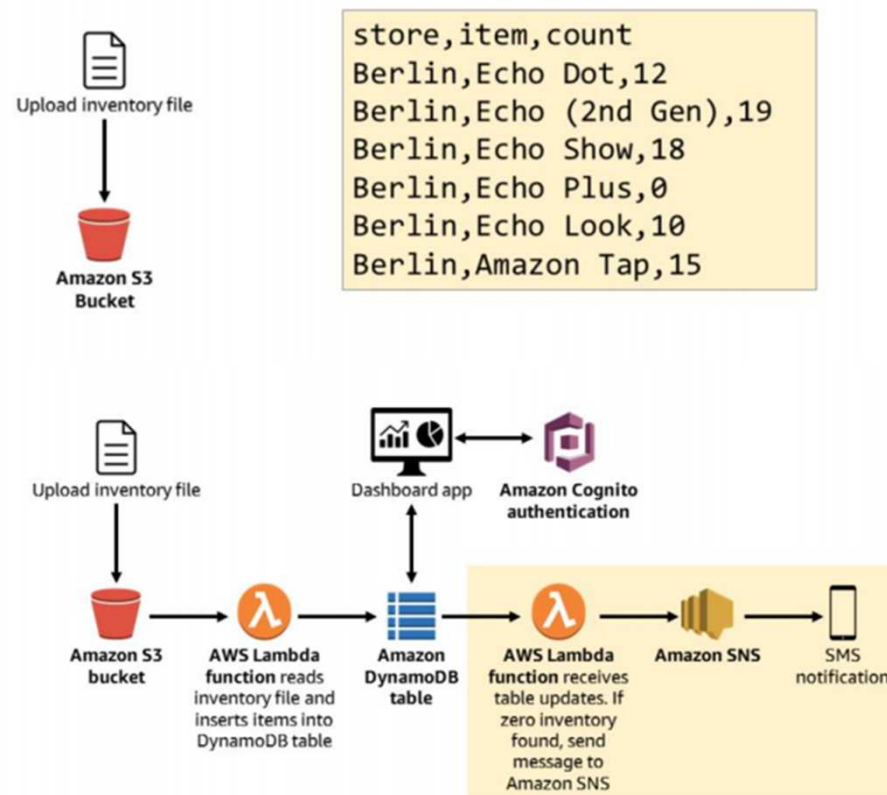
- EJEMPLO Video en demanda



Serverless: lambda

Orquestación de servicios Lambda

- EJEMPLO: Tracking de stock

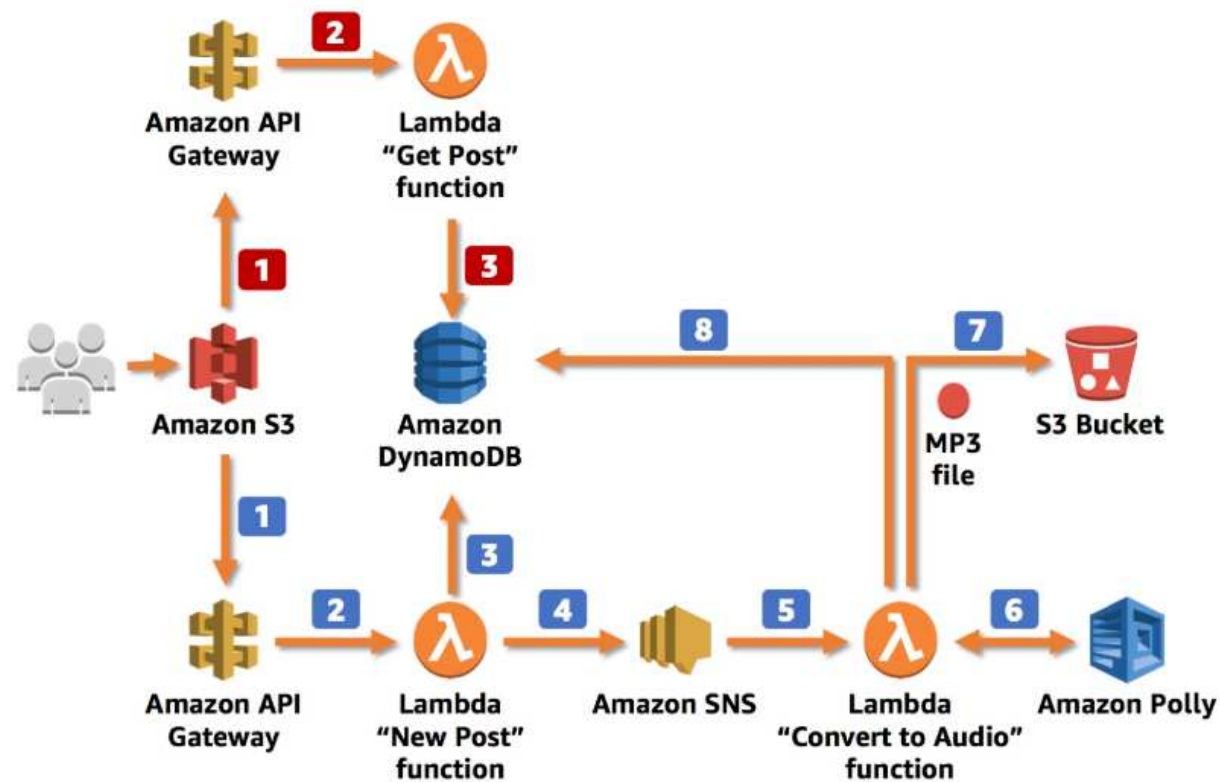


Serverless: lambda

- Automatizando el control de recursos AWS mediante AWS Lambda
- Para realizar Backups de servidores Ec2 (Volumenes EBS o AMIs) se podría
 - Tener un servidor dedicado Ec2
 - Con Script de Api Ec2 para realizar el snapshot
 - Y con Cron ejecutarlo cada X
- Se puede sustituir el servidor dedicado por un servicio Lambda
- En este caso se creara una Lambda que realizara el Backup vía AMI de todas aquellas instancias que tengan el Tag **Backup=yes**

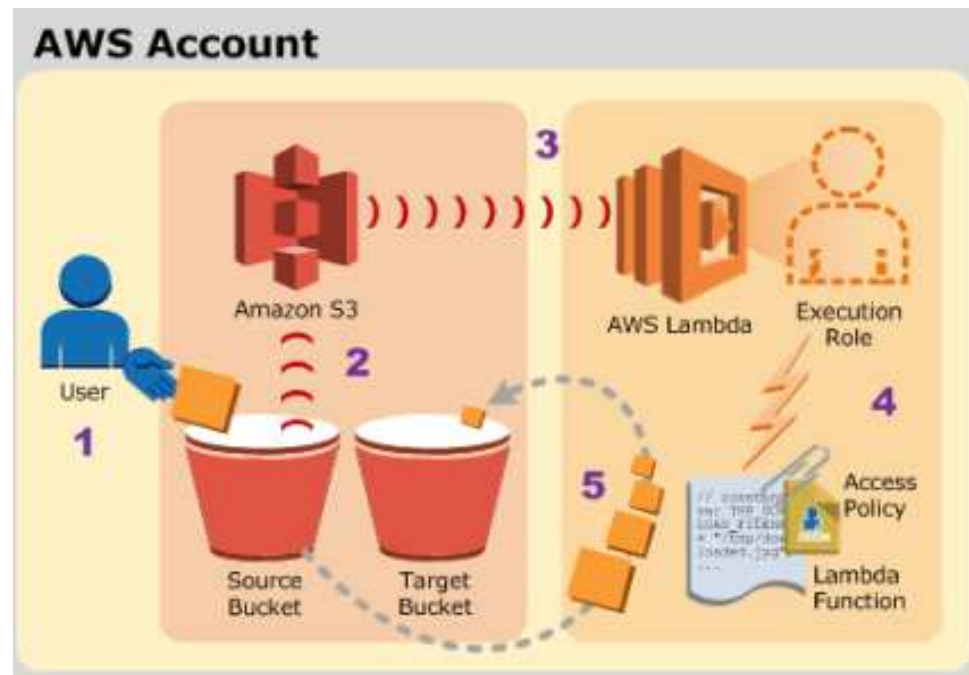
Serverless: lambda

- Ejemplo Text To Speech



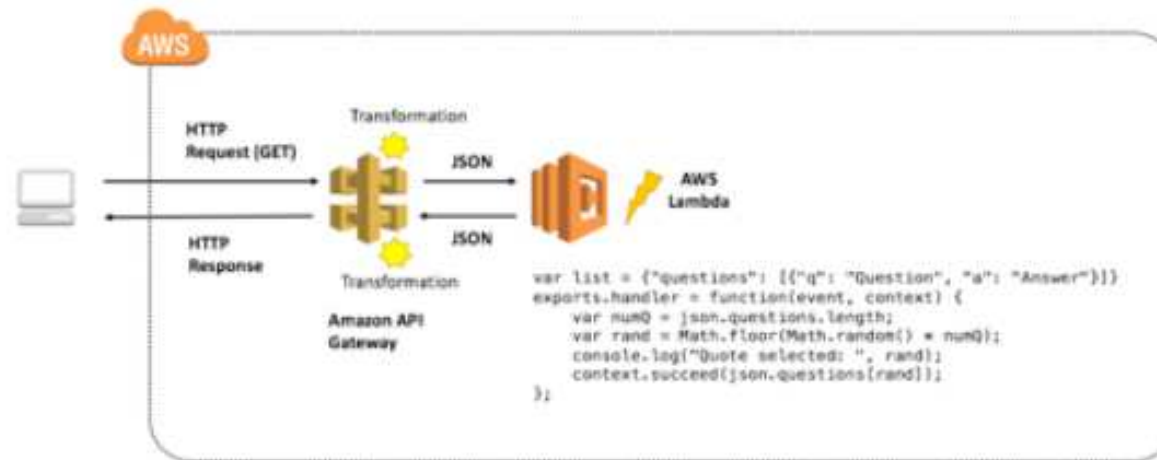
Serverless: lambda

- Práctica: QWIKLABS : Introduction to AWS Lambda
 - Un usuario subirá una imagen a S3 y esta acción creará un evento
 - El evento S3 lanzará una función Lambda que creará y guardará en un Bucket S3 un Thumbnail de la imagen subida a S3



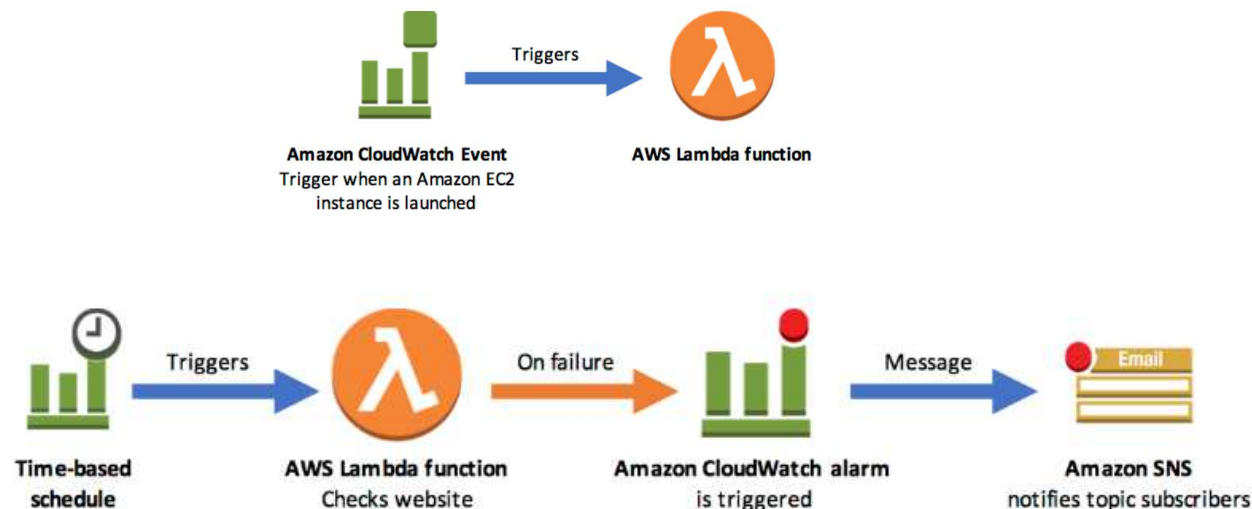
Serverless: lambda

- Práctica: QWIKLABS : Introduction to Amazon API Gateway
 - Se creara un pequeño API REST : /FAQ-API
 - Que contendrá una lista de preguntas y devolverá una de las preguntas de forma aleatoria por cada llamada
 - Se gestionará las llamadas al API REST mediante un API GATEWAY



Serverless: lambda

- Práctica: QWIKLABS :Serverless Architectures using Amazon CloudWatch Events and Scheduled Events with AWS Lambda
 - Esta práctica explica como podemos monitorizar las acciones de creación de recursos mediante Lambda
 - Se creara una función para detectar la creación de instancias EC2 y crear mensajes de log
 - SE creara una función Lambda para detectar la caída de servidores WEB



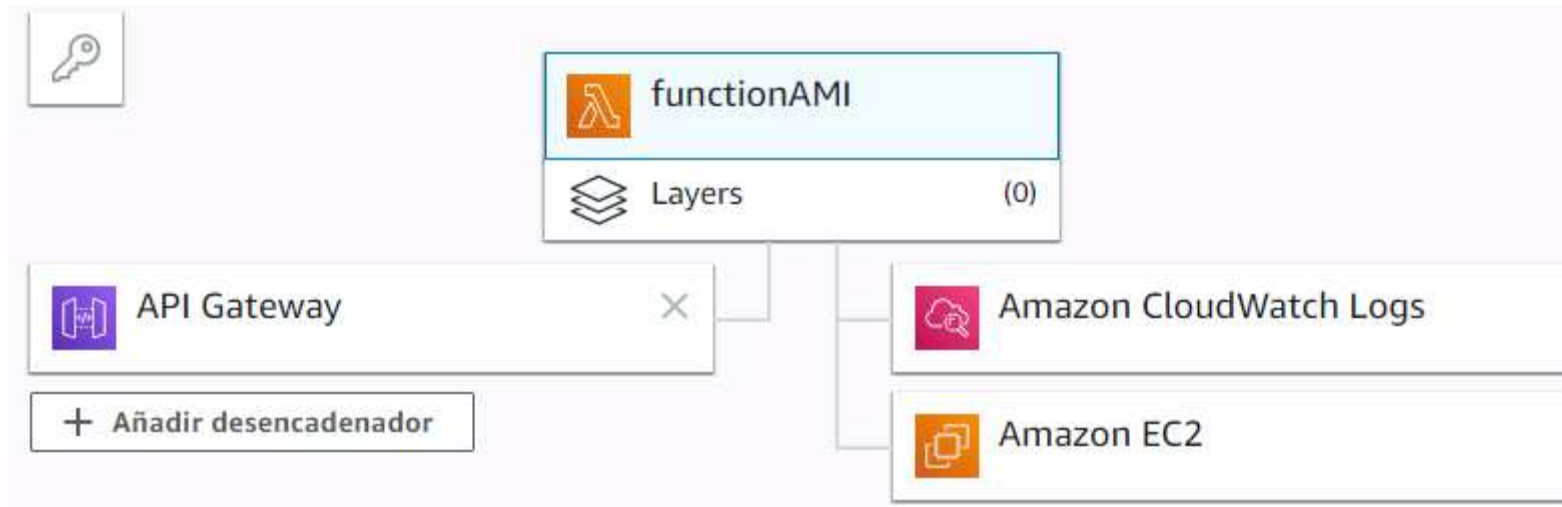
Serverless: lambda

- Ejercicios Serverless
 - Estos ejercicios se deben de realizar con el starter account o el classrom de SERVERLESS
 - API Gateway
 - Backups de AMI
 - Notificaciones vía Email de cambios en los grupos de seguridad
 - Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - QWIKSLAB :Using AWS Lambda with Amazon CloudWatch and SNS to Implement a Slack Chat Bot
 - QWIKSLAB: Build a Serverless Text-to-Speech Application with Amazon Polly

Serverless: lambda

Ejercicio: API Gateway

- Objetivo crear una función lambda que recibe ciertos datos y responde con un mensaje
- Se le dará ciertos privilegios (Ec2, Cloudwatch logs) mediante un Role
- Se le llamara desde un API gateway



Serverless: lambda

Ejercicio: API Gateway

Crear función Lambda

- Nombre: prueba1
- Node.js
- Permisos: Nuevo Role
 - Ec2
 - Cloudwatch

```
{  
  "Effect": "Allow",  
  "Action": "logs:*",  
  "Resource": "arn:aws:logs:*:*:*"  
},  
{  
  "Effect": "Allow",  
  "Action": "ec2:*",  
  "Resource": "*" }  
}
```

Crear una función [Información](#)

Seleccione una de las siguientes opciones para crear la función.

Crear desde cero

Empiece con un sencillo ejemplo "Hello World".



Información básica

Nombre de función

Especifique un nombre que describa el propósito de la función.

Utilice exclusivamente letras, números, guiones o guiones bajos. No incluya espacios.

Tiempo de ejecución [Información](#)

Seleccione el lenguaje que quiere utilizar para escribir la función.

Permisos [Información](#)

Lambda creará un rol de ejecución con permiso para cargar registros en Amazon CloudWatch Logs. Le

▼ Seleccionar o crear un rol de ejecución

Rol

Seleccione un rol que defina los permisos de la función. Para crear un rol personalizado, vaya a la [con](#)

☒ Creación de un nuevo rol con permisos básicos de Lambda

☐ Uso de un rol existente

☐ Creación de un nuevo rol desde la política de AWS templates

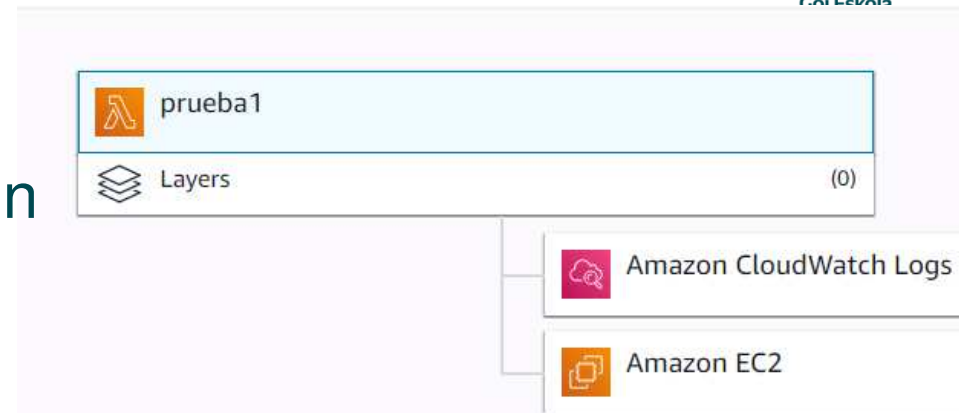
Serverless: lambda

Ejercicio: API Gateway

Implementación de la función

- Ir a código de la función
- Y si se quiere modificar

```
exports.handler = async (event) => {  
  let name = "you";  
  let city = 'World';  
  let responseCode = 200;  
  console.log("request: " + JSON.stringify(event));  
  if (event.queryStringParameters && event.queryStringParameters.name) {  
    console.log("Received name: " + event.queryStringParameters.name);  
    name = event.queryStringParameters.name;  
  }  
  if (event.queryStringParameters && event.queryStringParameters.city) {  
    console.log("Received city: " + event.queryStringParameters.city);  
    city = event.queryStringParameters.city;  
  }  
  let greeting = `Good ${name} of ${city}.`;  
  let responseBody = {  
    message: greeting  
  };  
};
```



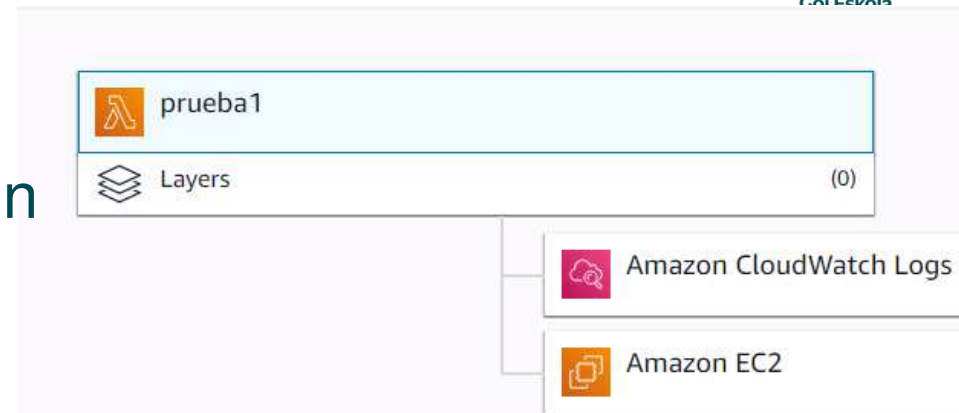
Serverless: lambda

Ejercicio: API Gateway

Implementación de la función

- Ir a código de la función
- Y si se quiere modificar

```
let response = {
  statusCode: responseCode,
  headers: {
    "x-custom-header" : "my custom header value"
  },
  body: JSON.stringify(responseBody)
};
console.log("response: " + JSON.stringify(response))
return response;
```



Serverless: lambda

Ejercicio: API Gateway

Una vez guardada la función crear el API Gateway que relacionará la URL con la función Lambda

- Servicio API Gateway -> Crear recurso

Recursos Acciones ▾ Nuevo recurso secundario

Utilice esta página para crear un nuevo recurso secundario para su recurso .

Configurar como [recurso de proxy](#) ☐

Nombre del recurso*

Ruta de recurso*

Puede agregar parámetros de ruta u
Al configurar `{proxy+}` como un recurso
GET a `/foo`. Para controlar las solicitudes

Habilitar API Gateway CORS ☐

Recursos Acciones ▾

/

/helloworld

ANY

/prueba1

Serverless: lambda

Ejercicio: API Gateway

Una vez guardada la función crear el API Gateway que relacionará la URL con la función Lambda

- Servicio API Gateway -> Crear método en el Recurso
 - Crearemos para cualquier tipo de petición ANY

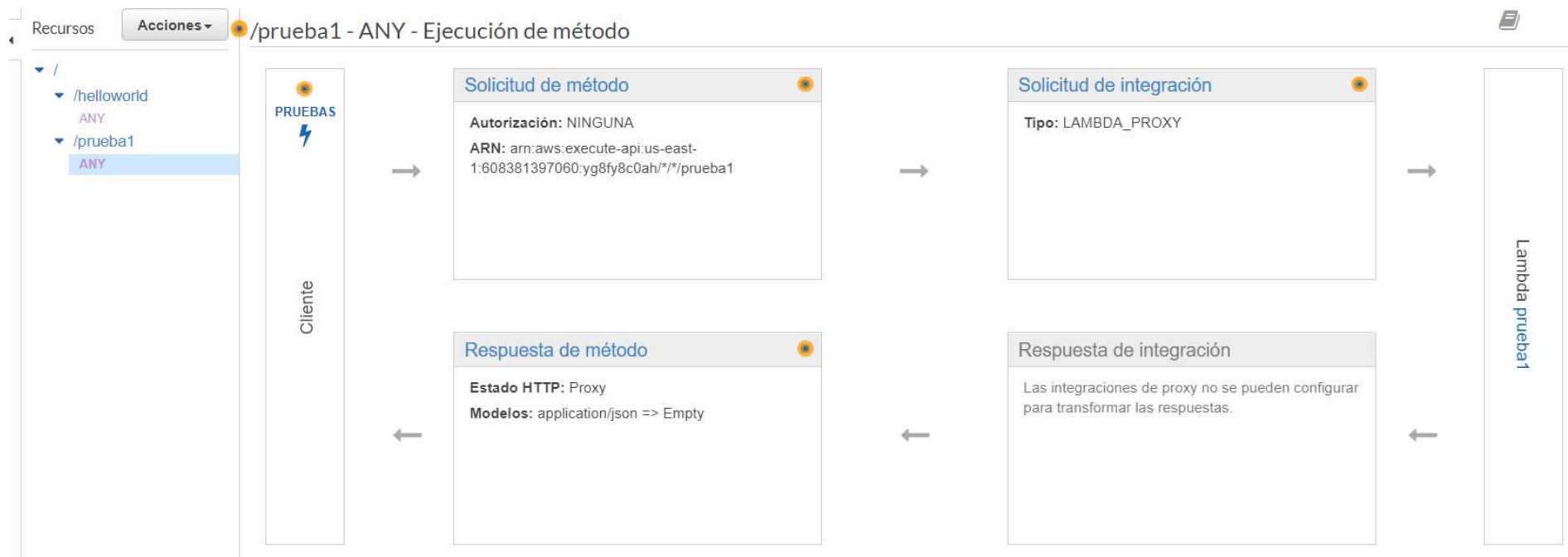
The screenshot shows the AWS API Gateway console interface. On the left, a resource tree shows a resource named `/prueba1` under the `/helloworld` path, with a method type of `ANY`. This resource is highlighted with a yellow circle. In the center, the `/prueba1 - ANY - Configuración` configuration page is displayed. The `Tipo de integración` (Integration type) is set to `Función Lambda`, which is also highlighted with a yellow circle. Below this, the `Usar la integración de proxy Lambda` checkbox is checked and highlighted with a yellow circle. The `Región Lambda` (Lambda region) is set to `us-east-1`. The `Función Lambda` (Lambda function) field is set to `prueba1`, which is also highlighted with a yellow circle. The `Usar tiempo de espera predeterminado` (Use default timeout) checkbox is checked.

Serverless: lambda

Ejercicio: API Gateway

Una vez guardada la función crear el API Gateway que relacionará la URL con la función Lambda

- Servicio API Gateway -> Crear método en el Recurso



Serverless: lambda

Ejercicio: API Gateway

Una vez guardad la función crear el API Gateway que relacionará la URL con la función Lambda

- Y ahora se implementa/Despliega el API

The screenshot shows the AWS API Gateway console. On the left, the 'Recursos' (Resources) pane shows a tree structure with a resource named '/prueba' selected. A dropdown menu for 'Acciones' (Actions) is open, showing options under 'ACCIONES DE RECURSO' (Resource Actions) and 'ACCIONES API' (API Actions). The 'Implementar la API' (Implement API) option is highlighted in yellow. On the right, the 'Implementar la API' (Implement API) step is shown. It includes a heading 'Implementar la API' with a blue icon, a description 'Elija una etapa donde se implementará su API. Por ejemplo, una versión de prueba de API se podría implementar en una etapa denominada beta.', and a form with two fields: 'Etapa de implementación' (Implementation stage) with a dropdown menu showing 'test' (highlighted with a yellow circle) and 'Descripción de implementación' (Implementation description) with a text input field.

Serverless: lambda

Ejercicio: API Gateway

Una vez guardada la función crear el API Gateway que relacionará la URL con la función Lambda

- Probar URL

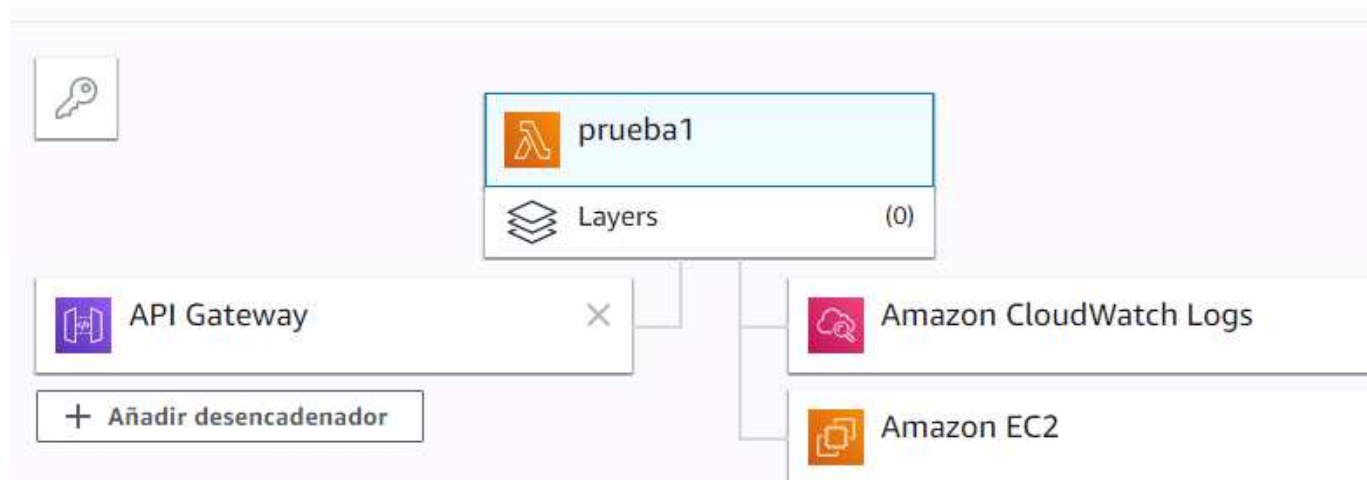
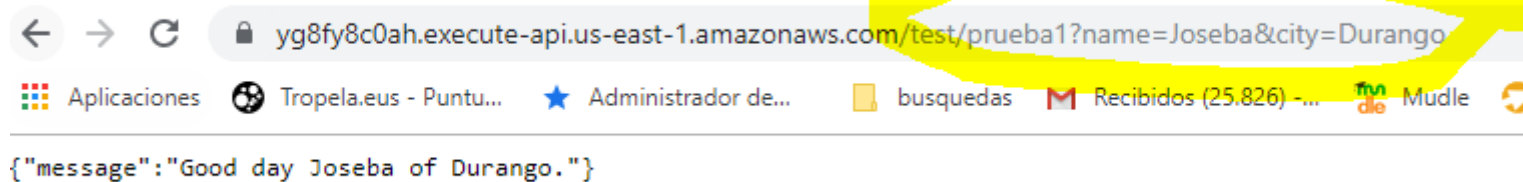


Serverless: lambda

Ejercicio: API Gateway

Una vez guardada la función crear el API Gateway que relacionará la URL con la función Lambda

- Probar URL



Serverless: lambda

- Ejercicio: Backups de AMI

Objetivo: Realizar Backup de aquellas instancias con el tag de Backup=yes cada X utilizando Cloudwatch como Cron

- Starter Account (Por temas de permisos)
- Una vez creada alguna instancia y creado el tag BACKUP=yes
- Crear una función lambda siguiendo los pasos del anterior ejemplo pero con el código de takingAMI.js
- Esta función busca aquellas instancias que tienen el tag BACKUP=yes navegando en los Reservations[] que devuelve la función ***ec2.describeInstances(instanceparams)*** y obtiene el valor del tag Name que utilizara para crear el AMI
- Y crea las instancias mediante ***ec2.createImage(imagep***

Serverless: lambda

- Ejercicio: Backups de AMI

Objetivo: Realizar Backup de aquellas instancias con el tag de Backup=yes

```
var aws = require('aws-sdk');
aws.config.region = 'us-east-1';
var ec2 = new aws.EC2();
var now = new Date();
date = now.toISOString().substring(0, 10)
hours = now.getHours()
minutes = now.getMinutes()
exports.handler = function(event, context) {
    var instanceparams = {
        Filters: [{
            Name: 'tag:Backup',
            Values: [
                'yes'
            ]
        }]
    }
}
```

Serverless: lambda

- Ejercicio: Backups de AMI

Objetivo: Realizar Backup de aquellas instancias con el tag de Backup=yes

```
ec2.describeInstances(instanceparams, function(err, data) {  
  if (err) console.log(err, err.stack);  
  else {  
    for (var i in data.Reservations) {  
      for (var j in data.Reservations[i].Instances) {  
        instanceid = data.Reservations[i].Instances[j].InstanceId;  
        nametag = data.Reservations[i].Instances[j].Tags  
        for (var k in data.Reservations[i].Instances[j].Tags) {  
          if (data.Reservations[i].Instances[j].Tags[k].Key == 'Name') {  
            name = data.Reservations[i].Instances[j].Tags[k].Value;  
          }  
        }  
        console.log("Creating AMIs of the Instance: ", name);  
      }  
    }  
  }  
}
```


Serverless: lambda

- Ejercicio: Backups de AMI

Objetivo: Realizar Backup de aquellas instancias con el tag de Backup=yes

```
console.log("Creating AMIs of the Instance: ", name);
var imageparams = {
  InstanceId: instanceid,
  Name: name + "_" + date + "_" + hours + "-" + minutes,
  NoReboot: true
}
ec2.createImage(imageparams, function(err, data) {
  if (err) console.log(err, err.stack);
  else {
    image = data.ImageId;
    console.log(image);
    var tagparams = {
      Resources: [image],
      Tags: [{
        Key: 'DeleteOn',
        Value: 'yes'
      }]
    };
    ec2.createTags(tagparams, function(err, data) {
      if (err) console.log(err, err.stack);
      else console.log("Tags added to the created AMIs");
    });
  }
});
```

Serverless: lambda

- Ejercicio: Backups de AMI

Estableciendo el Trigger de Cloudwatch events

Crearemos un trigger de cloudwatch que se ejecutara con la periodicidad que queramos

Serverless: lambda

- Ejercicio: Backups de AMI

Objetivo: Realizar Backup de aquellas instancias con el tag de Backup=yes

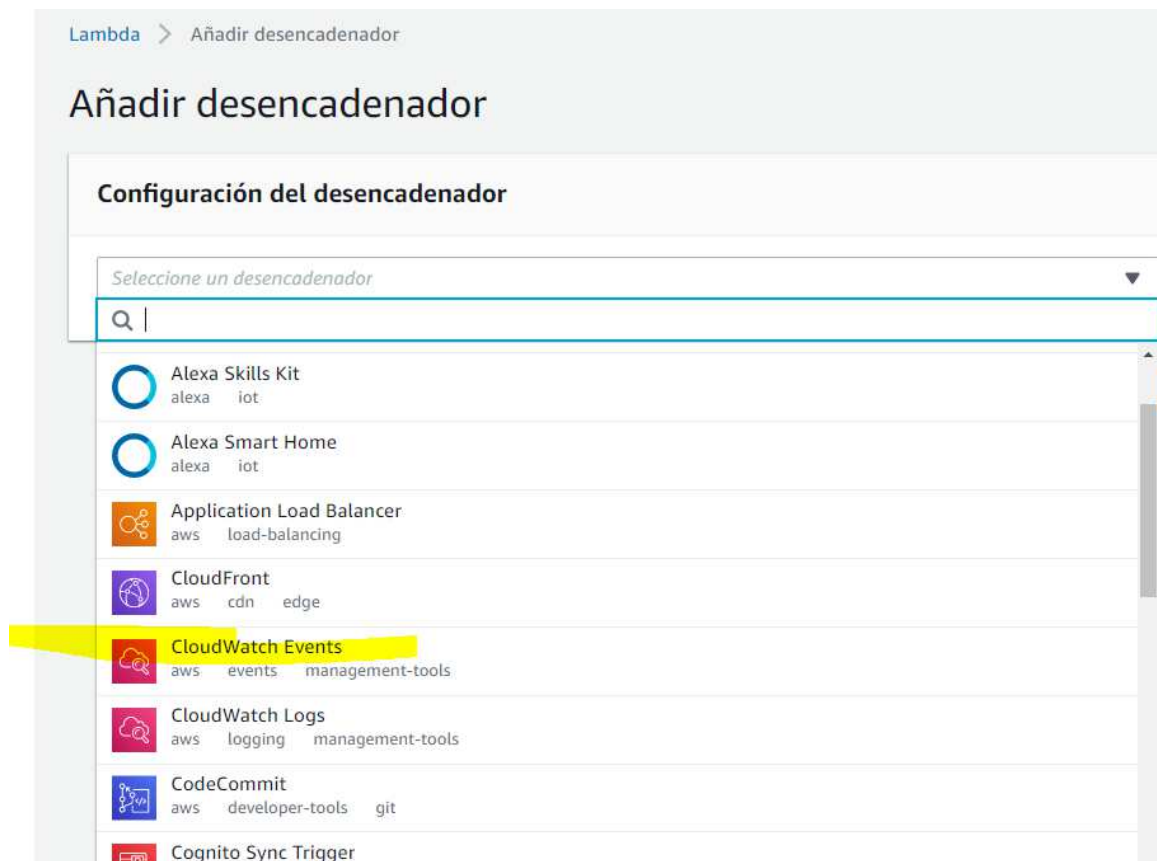
```
var aws = require('aws-sdk');
aws.config.region = 'us-east-1';
var ec2 = new aws.EC2();
var now = new Date();
date = now.toISOString().substring(0, 10)
hours = now.getHours()
minutes = now.getMinutes()
exports.handler = function(event, context) {
    var instanceparams = {
        Filters: [{
            Name: 'tag:Backup',
            Values: [
                'yes'
            ]
        }]
    }
}
```

Serverless: lambda

- Ejercicio: Backups de AMI

Ahora se definirá un Trigger Cloudwatch para la función lambda

- Así se consigue realizar cada X tiempo el backup de la instancia



Serverless: lambda

- Ejercicio: Backups de AMI

Creación de la regla de periodicidad para el evento Cloudwatch -> rate(5 minutes) , permite reglas Cron

- cron(0 17 ? * MON-FRI *)
- Rate(5 minutes)

The screenshot shows the AWS CloudWatch Events console. At the top, there's a header for 'CloudWatch Events' with sub-headers 'aws', 'events', and 'management-tools'. Below this, the 'Regla' section prompts the user to 'Seleccione una regla existente o cree una nueva.' A dropdown menu is open, showing 'Cree una regla nueva' highlighted in yellow. Below this, the 'Nombre de la regla*' field contains 'CW-functionAMI' and is also highlighted in yellow. The 'Descripción de regla' field contains 'Taking AMI of EC2 instances'. In the 'Tipo de regla' section, the 'Expresión de programación' radio button is selected and highlighted with a yellow circle. Below this, the 'Expresión de programación*' field contains 'rate(5 minutes)' and is also highlighted with a yellow circle. At the bottom, there is a note: 'por ejemplo, rate(1 day), cron(0 17 ? * MON-FRI *)'.

CloudWatch Events
aws events management-tools

Regla
Seleccione una regla existente o cree una nueva.

Cree una regla nueva

Seleccione o cree una regla nueva

Nombre de la regla*
Especifique un nombre que identifique la regla de forma inequívoca.

CW-functionAMI

Descripción de regla
Proporcione una descripción opcional para la regla.

Taking AMI of EC2 instances

Tipo de regla
Active el destino con arreglo a un patrón de eventos o una programación automatizada.

☐ Patrón de eventos

☒ Expresión de programación

Expresión de programación*
Active automáticamente el destino con arreglo a una programación automatizada utilizando expresiones Cron o de frecuencia. Las expresiones Cron utilizan el formato UTC.

rate(5 minutes)

por ejemplo, rate(1 day), cron(0 17 ? * MON-FRI *)

Serverless: lambda

- Ejercicio: Backups de AMI

Creación de la regla de periodicidad para el evento
Cloudwatch -> rate(5 minutes)

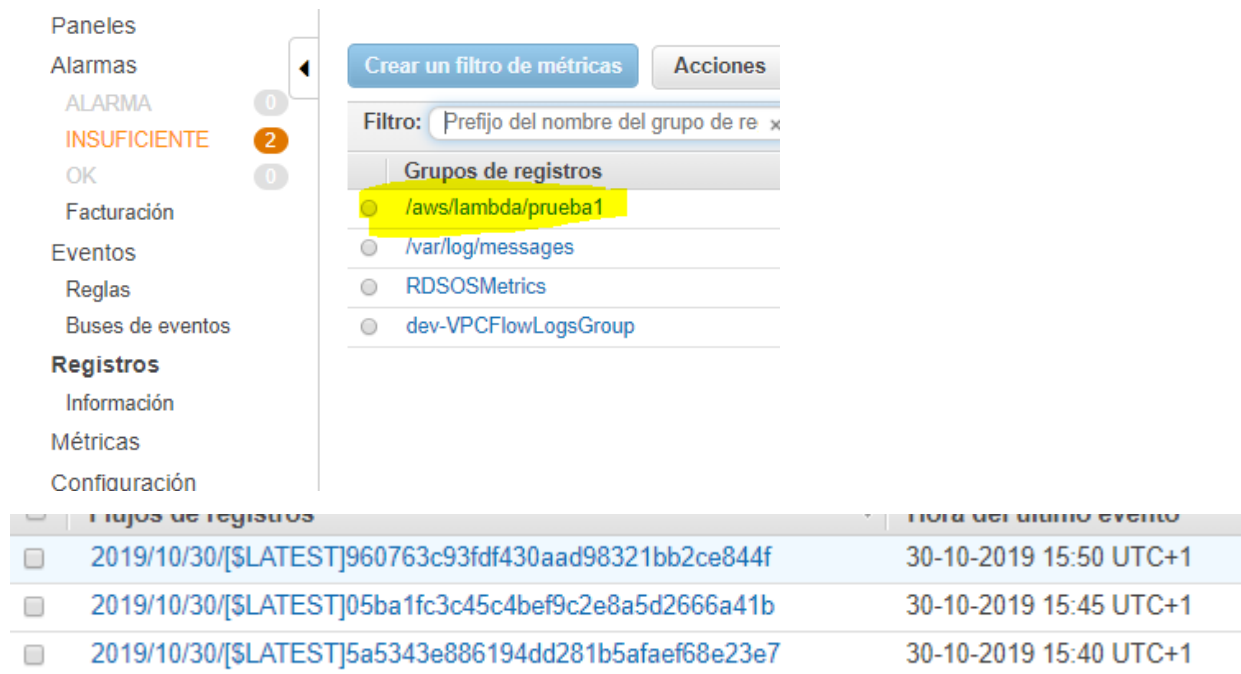


Serverless: lambda

- Ejercicio: Backups de AMI

Creación de la regla de periodicidad para el evento
Cloudwatch -> rate(5 minutes)

- Comprobar los logs de Cloudwatch y AMI de EC2



Grupos de registros

Log Group	Time del último evento
2019/10/30/[\$LATEST]960763c93fdf430aad98321bb2ce844f	30-10-2019 15:50 UTC+1
2019/10/30/[\$LATEST]05ba1fc3c45c4bef9c2e8a5d2666a41b	30-10-2019 15:45 UTC+1
2019/10/30/[\$LATEST]5a5343e886194dd281b5afaef68e23e7	30-10-2019 15:40 UTC+1

Serverless: lambda

- Ejercicio: Backups de AMI

Creación de la regla de periodicidad para el evento Cloudwatch -> rate(5 minutes)

- Comprobar los logs de Cloudwatch generados por el Lambda y AMI de EC2

Filtrar eventos	
Hora (UTC +00:00)	Mensaje
2019-10-30	
No hay eventos anteriores actualmente. Reintente.	
▶ 15:00:06	START RequestId: 65a26e87-3e61-4159-81e2-f64e5919d4f9 Version: \$LATEST
▼ 15:00:07	2019-10-30T15:00:07.693Z 65a26e87-3e61-4159-81e2-f64e5919d4f9 INFO Creating AMIs of the Instance: froga_backup
2019-10-30T15:00:07.693Z 65a26e87-3e61-4159-81e2-f64e5919d4f9 INFO Creating AMIs of the Instance: froga_backup	
▼ 15:00:08	2019-10-30T15:00:08.093Z 65a26e87-3e61-4159-81e2-f64e5919d4f9 INFO ami-06155a2d4326998d4
2019-10-30T15:00:08.093Z 65a26e87-3e61-4159-81e2-f64e5919d4f9 INFO ami-06155a2d4326998d4	
▶ 15:00:08	2019-10-30T15:00:08.396Z 65a26e87-3e61-4159-81e2-f64e5919d4f9 INFO Tags added to the created AMIs
▶ 15:00:08	END RequestId: 65a26e87-3e61-4159-81e2-f64e5919d4f9
▶ 15:00:08	REPORT RequestId: 65a26e87-3e61-4159-81e2-f64e5919d4f9 Duration: 1898.29 ms Billed Duration: 1900 ms Memory Size: 128 MB Max Mem
No hay eventos recientes actualmente. Reintente.	

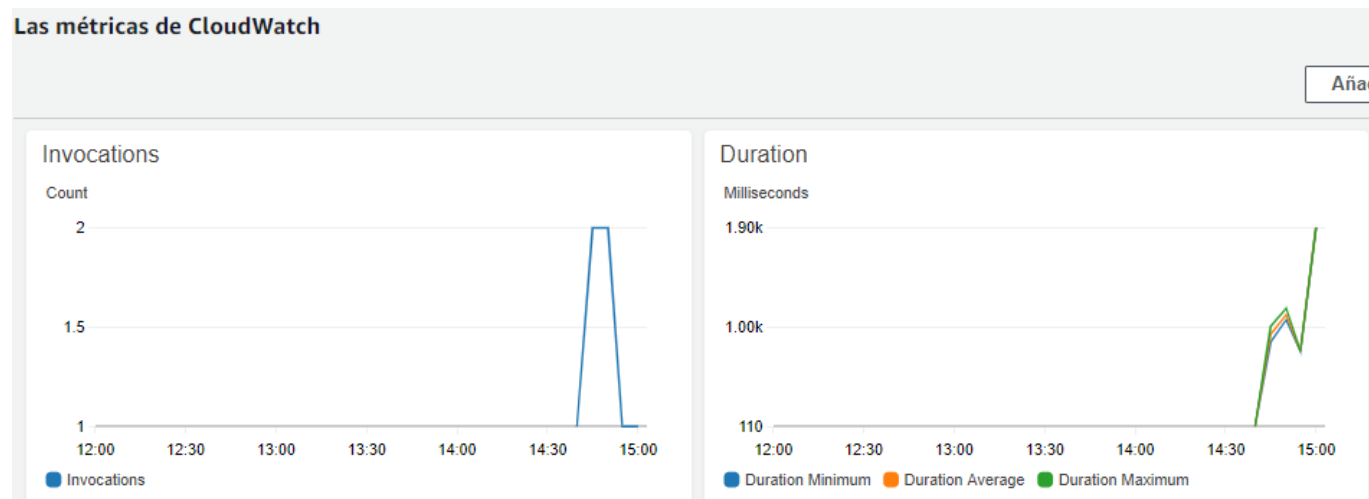
Owned by me		Filter by tags and attributes or search by keyword						
<input checked="" type="checkbox"/>	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
<input checked="" type="checkbox"/>	froga_backup_...		ami-06155a2d4326998d4	420693608596/f...	420693608596	Private	available	October 30, 2019 at 4:00:08

Serverless: lambda

- Ejercicio: Backups de AMI

Creación de la regla de periodicidad para el evento
Cloudwatch -> rate(5 minutes)

- Comprobar la monitorización Cloudwatch del Lambda



Serverless: lambda

- Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con **AWS Config**
 - Se creara un Topic SNS para el administrador que recibirá notificaciones de Email cada vez que haya un cambio en los recursos mencionados en **AWS Config**
 - Validar un email como el responsable de un grupo de seguridad. Este email solo recibirá mensajes cuando el grupo de seguridad se vea modificado
 - Se creara una regla de AWS Config
 - Se creara una función Lambda que enviara notificaciones cuando ocurran ciertos eventos , en este caso el cambios en el grupo de seguridad

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Creación de un Tema/Topic SNS

Detalles

Nombre

Máximo de 256 caracteres. Puede incluir caracteres alfanuméricos, guiones y guiones bajos.

Nombre para visualización - *opcional*

Para utilizar este tema con suscripciones a SMS, escriba un nombre para el tema.

Máximo de 100 caracteres, incluidos guiones (-) y guiones bajos (_).

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Ahora crearemos una suscripción al Topic
- Y confirmaremos desde el email

Crear una suscripción

Detalles

ARN del tema

Protocolo

El tipo de punto de enlace para suscribirse

Correo electrónico

▼

Punto de enlace

Una dirección de correo electrónico que puede recibir notificaciones de Amazon SNS.

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config


- Y confirmaremos desde el email

SG-TEST

Berria

AWS Notification - Subscription Confirmation - You h

AWS Notification - Subscription Confirmation

amazon web services

Simple Notification

SG-TEST <no-reply@sns.amazonaws.com>
hartzaileak: ni

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:608381397060:SNS-SG-Test

To confirm this subscription, click or visit the link below (If this was in error no a
[Confirm subscription](#)

Please do not reply directly to this email. If you wish to remove yourself from receiving all future

Subscription confirmed!

You have subscribed jaagibas@gmail.com to the topic:
SNS-SG-Test

Your subscription's id is:
arn:aws:sns:us-east-1:608381397060:SNS-SG-Test:550e1865-a611-43bf-a:
8adba1258681

If it was not your intention to subscribe, [click here to unsubscribe](#).

ID	Punto de enlace	Estado	Protocolo	Tema
550e1865-a611-43bf-a21f-8adba1258681	jaagibas@gmail.com	Confirmada	EMAIL	SNS-SG-Test

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Ahora estableceremos un email externo de confianza para enviar email en su nombre



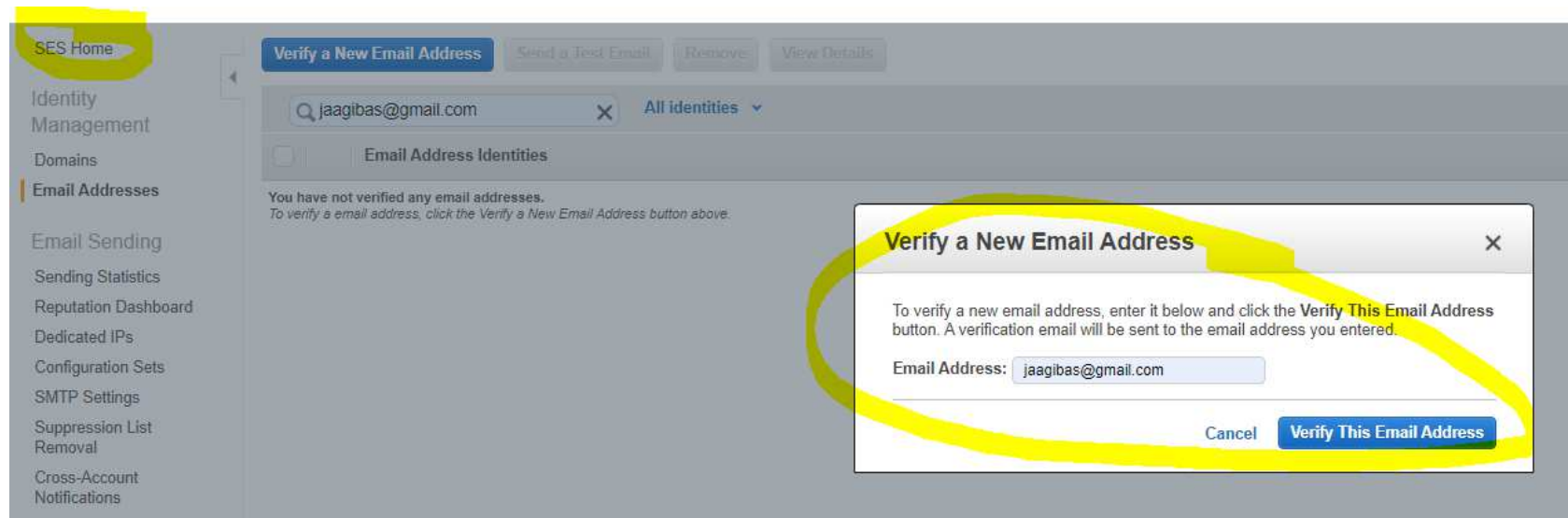
Amazon Simple Email Service

Amazon Simple Email Service enables you to send and receive email using a reliable and scalable email platform.

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

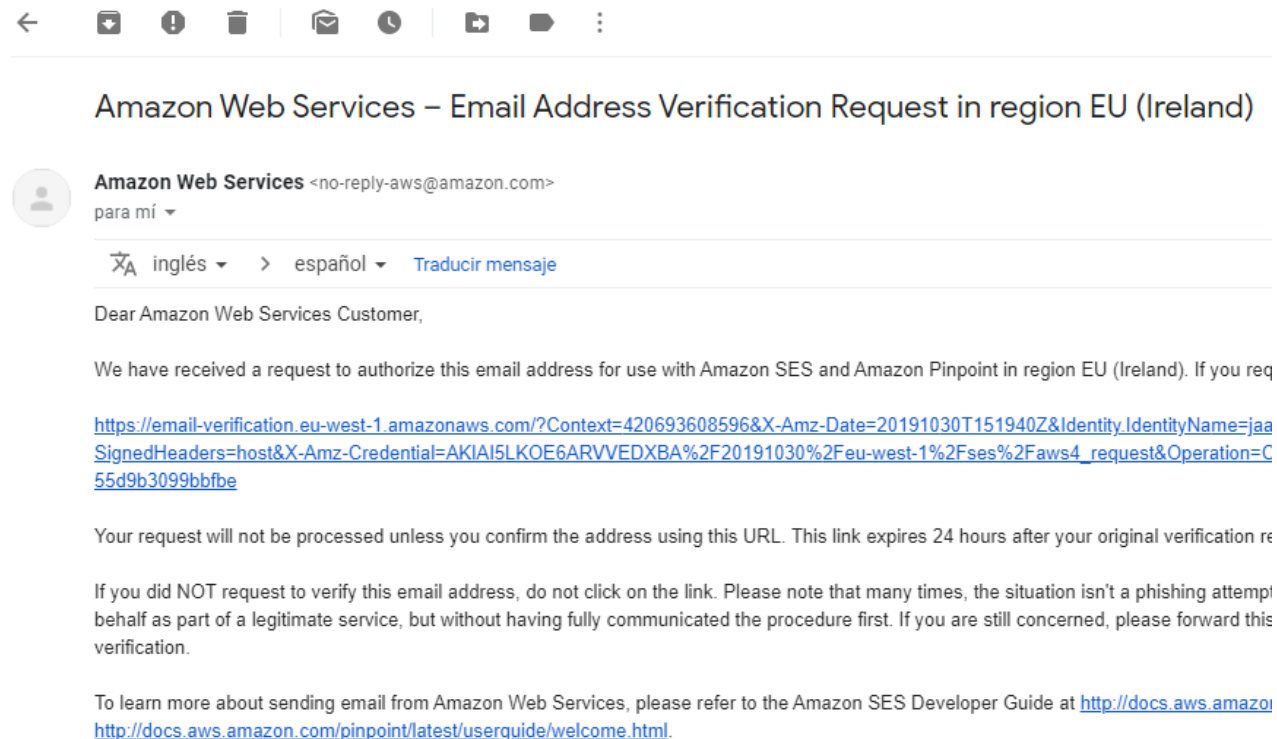
- Ahora estableceremos un email externo de confianza para enviar email en su nombre, en este caso supondremos que el dueño es jaagirre@mondragon.edu



Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Ahora estableceremos un email externo de confianza para enviar email en su nombre, en este caso supondremos que el dueño es jaagirre@mondragon.edu



Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Esto permite enviar emails en nombre de jaagirre@mondragon.edu (también validar jaagibas@gmail.com)

¡Enhorabuena!

Ha verificado con éxito una dirección de correo electrónico. Ya puede empezar a enviar correos electrónicos desde esta dirección.

Para los nuevos usuarios de Amazon SES – Si aún no ha solicitado un aumento de límite de envío, siga estando en el [entorno de pruebas](#) y solo puede enviar correos electrónicos a direcciones que haya verificado previamente. Para verificar una nueva dirección de correo electrónico o dominio, consulte la sección **Administración de identidades de la [consola de Amazon SES](#)**.

Para los nuevos usuarios de Amazon Pinpoint – Si aún no ha solicitado un aumento de límite de envío sigue estando en el [entorno de pruebas](#) y solo puede enviar correos electrónicos a direcciones que haya verificado previamente. Para verificar una nueva dirección de correo electrónico o dominio, consulte la página **Configuración > Canales** de la [consola de Amazon Pinpoint](#).

Si ya ha obtenido la aprobación para el aumento de límite de envío, puede comenzar a enviar correos

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Esto permite enviar emails en nombre de jaagirre@mondragon.edu (también validar jaagibas@gmail.com)

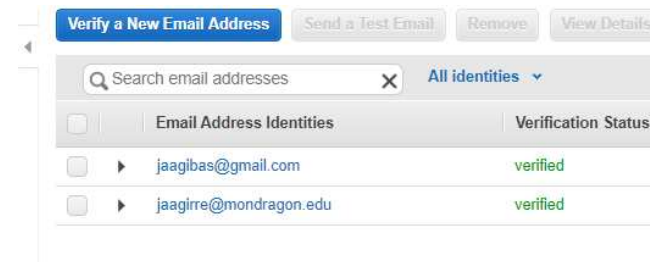
¡Enhorabuena!

Ha verificado con éxito una dirección de correo electrónico. Ya puede empezar a enviar correos electrónicos desde esta dirección.

Para los nuevos usuarios de Amazon SES – Si aún no ha solicitado un aumento de límite de envío, siga estando en el [entorno de pruebas](#) y solo puede enviar correos electrónicos a direcciones que haya verificado previamente. Para verificar una nueva dirección de correo electrónico o dominio, consulte la sección **Administración de identidades de la [consola de Amazon SES](#)**.

Para los nuevos usuarios de Amazon Pinpoint – Si aún no ha solicitado un aumento de límite de envío sigue estando en el [entorno de pruebas](#) y solo puede enviar correos electrónicos a direcciones que haya verificado previamente. Para verificar una nueva dirección de correo electrónico o dominio, consulte la página **Configuración > Canales** de la [consola de Amazon Pinpoint](#).

Si ya ha obtenido la aprobación para el aumento de límite de envío, puede comenzar a enviar correos



Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Configuraremos AWS Config para Recursos AWS
- AWS Config permite registrar los cambios de configuración de los recursos
- Es perfecto para un aPYME con pocos administradores, pero hay que combinar con Lambda en organizaciones mayores



AWS Config

AWS Config proporciona un inventario de sus recursos de AWS y un historial de cambios de configuración en estos recursos. Puede utilizar AWS Config para definir las reglas que evalúan la conformidad de estas configuraciones.

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Crearemos una configuración que notifique utilizando el Topic anteriormente creado

Tipos de recursos que se van a registrar

Seleccionar los tipos de recursos de AWS para los que desea que AWS Config registre los cambios en la configuración. De forma predeterminada, AWS Config registra los recursos globales admitidos en esta región.

Todos los recursos ☒ Registrar todos los recursos admitidos en esta región ⓘ
☐ Incluir recursos globales (por ejemplo, recursos de AWS IAM) ⓘ

Tipos específicos

Bucket de Amazon S3*

El bucket recibe el historial de configuración y los archivos de instantáneas de configuración, que contienen detalles de los recursos que registra AWS Config.

☒ Crear un bucket
☐ Elegir un bucket de su cuenta
☐ Elegir un bucket de otra cuenta ⓘ

Nombre del bucket* / / AWSLogs/420693608596/Config/eu-west-1

Tema de Amazon SNS

☒ Transmite los cambios de configuración y las notificaciones a un tema de Amazon SNS.
⚠ Si elige el correo electrónico como punto de enlace para las notificaciones de su tema de SNS, es posible que reciba una gran cantidad de correos electrónicos.

☐ Crear un tema
☒ Elegir un tema de su cuenta
☐ Elegir un tema de otra cuenta ⓘ

Nombre del tema*

Rol de AWS Config*

Conceda a AWS Config acceso de solo lectura a los recursos de AWS para que pueda registrar la información de configuración y concédale permiso para enviar esta i

☒ Crear un rol vinculado al servicio de AWS Config
☐ Elegir una función de su cuenta

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- A partir de ahora cualquier cambio se notificara al administrador

The screenshot displays the AWS Config console interface. On the left, a sidebar contains navigation links: Panel, Reglas, Recursos, Consulta avanzada, Configuración, and Novedades. The main content area features a blue header with a welcome message and a disclaimer. Below this, the 'Panel' section is active, showing a table of 'Recursos totales (63)'. The table lists various AWS resources and their counts. To the right of the table, the 'Estado de conformidad' section shows that all rules and resources are compliant. Below this, a section titled 'Reglas no conformes por recuento de recursos no conformes' indicates that there are no non-compliant rules.

Recursos totales (63)

Tipo	Recuento
EC2 SecurityGroup	20
S3 Bucket	12
EC2 Subnet	8
EC2 InternetGateway	3
EC2 VPC	3
RDS DBSnapshot	3
EC2 RouteTable	3
EC2 NetworkAcl	3
CloudWatch Alarm	2
EC2 Instance	1

[Ver los 63 recursos](#)
Ejecute consultas avanzadas con respecto a los datos de configuración de recursos.

Estado de conformidad

Reglas	Recursos
⚠️ 0 regla(s) no conforme(s) ✅ 0 regla(s) conforme(s)	⚠️ 0 Recurso(s) no conforme(s) ✅ 0 Recurso(s) conforme(s)

Reglas no conformes por recuento de recursos no conformes

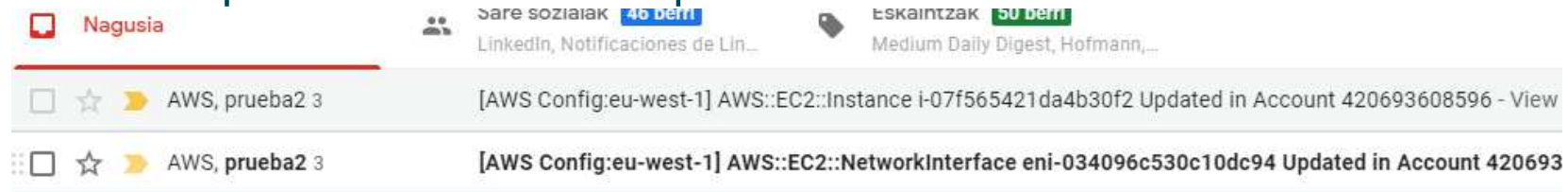
Nombre	Conformidad
No hay reglas no conformes.	

[Ver todas las reglas no conformes](#)

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- A partir de ahora cualquier cambio se notificara al administrador



prueba2 no-reply@sns.amazonaws.com amazonses.com domeinuaren bidez
hartzaileak: ni

🌐 ingeles > euskara Itzuli mezua

View the Timeline for this Resource in AWS Config Management Console:

<https://console.aws.amazon.com/config/home?region=eu-west-1#timeline/AWS::EC2::Instance/i-07f565421da4b30f2?>

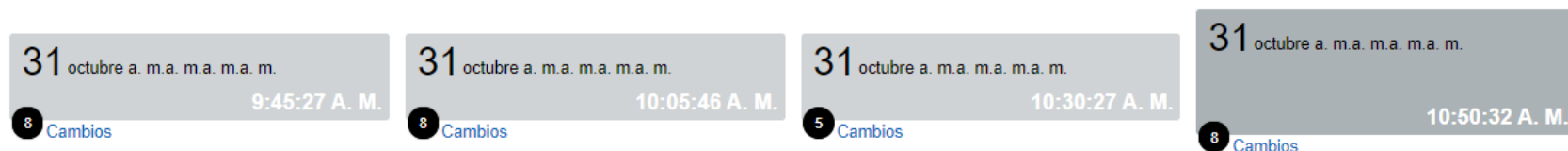
New State and Change Record:

```
{
  "configurationItemDiff": {
    "changedProperties": {
      "Configuration.NetworkInterfaces.0": {
        "previousValue": {
          "association": {
            "ipOwnerId": "amazon",
            "updatedValue": null,
            "changeType": "DELETE"
          },
          "Configuration.NetworkInterfaces.1": {
            "previousValue": null,
            "updatedValue": {
              "association": {
                "attachment": {
                  "attachTime": "2019-10-31T08:27:27.000Z",
                  "attachmentId": "eni-attach-01cb1daf8b20c9115",
                  "attachmentName": "eni-attach-01cb1daf8b20c9115"
                }
              }
            }
          }
        }
      }
    }
  }
}
```


Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Y también desde la consola AWS Config mediante la escala de tiempo



Panel de configuración

Recursos	
Recuento total de recursos	101
Los 10 tipos de recursos principales	Total
EC2 SecurityGroup	19
S3 Bucket	16
IAM Role	15
IAM User	9
EC2 Subnet	8
IAM Group	5

Campo	De
Configuration.NetworkInterfaces.0	<pre> Object association: Object ipOwnerId: "amazon" publicDnsName: "ec2-34-244-6-50.eu-west-1.compute.amazonaws.com" publicIp: "34.244.6.50" attachment: Object attachTime: "2019-10-31T08:27:27.000Z" attachmentId: "eni-attach-01cb1daf8b20c9115" deleteOnTermination: true deviceIndex: 0 status: "attached" description: "" groups: Array [1] 0: Object groupName: "launch-wizard-13" groupId: "sg-0857b522e6557f112" ipv6Addresses: Array [0] [] macAddress: "06:cf:03:1c:6d:12" networkInterfaceId: "eni-034096c530c10dc94" ownerId: "420693608596" privateDnsName: "ip-172-31-24-198.eu-west-1.compute.internal" </pre>

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Se cobra 0,003 por cambio en configuración de recurso

Evaluaciones de reglas de AWS Config	Precio
Primeras 100 000 evaluaciones de reglas	0,001 USD por evaluación de regla por región
Las siguientes 400 000 evaluaciones de regla (100 001-500 000)	0,0008 USD por evaluación de regla por región
500 001 evaluaciones de regla y más	0,0005 USD por evaluación de regla por región

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Podríamos utilizar un Lambda para filtrar los mensajes y enviar ciertos mensajes a ciertas personas
 - Objetivo: Enviar desde un email verificado a otro email
 - Pondremos un trigger SNS a un Lambda
 - Daremos privilegios al Lambda para crear eventos de LOG y enviar emails
 - Cuando reciba un mensaje SNS de AWS Config analizara si el mensaje se refiere a un SG y si el SG tiene un TAG_NotifyOnChange=yes y el TAG_OWNERDL=emailDelResponsbaleSG

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
var fromAddress = "";
var aws = require('aws-sdk');
var ses = new aws.SES({ apiVersion: '2010-12-01' });

exports.handler = function (event, context) {
  // Validate the incoming event
  if (!event.Records
    || event.Records.length == 0
    || !event.Records[0].Sns
    || !event.Records[0].Sns.Message) {
    context.done();
    return;
  }
  // The "message" from the event is stringified - it needs to be parsed into an object
  var message = JSON.parse(event.Records[0].Sns.Message);
  // Validate the format of the SNS message
  if (!message || !message.configurationItemDiff || !message.configurationItem) {
    console.log("The 'message' format is invalid. Stopping execution...")
    context.done();
    return;
  }
}
```

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
// Make sure the change notification is for a Security Group
if (message.configurationItem.resourceType !== 'AWS::EC2::SecurityGroup') {
    console.log('Change notifications are only supported for Security Groups.');
```

context.done();

return;

}


```
// Check to make sure notifications are enabled for this SG
var notifyOnChange = message.configurationItem.tags["NotifyOnChange"];

if (!notifyOnChange || notifyOnChange == 'No' || notifyOnChange == 'False') {
    console.log('Notifications are not enabled for the Security Group in question.');
```

context.done();

return;

}

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
// Make sure the change notification is for a Security Group
if (message.configurationItem.resourceType !== 'AWS::EC2::SecurityGroup') {
    console.log('Change notifications are only supported for Security Groups.');
```

context.done();

return;

}


```
// Check to make sure notifications are enabled for this SG
var notifyOnChange = message.configurationItem.tags["NotifyOnChange"];

if (!notifyOnChange || notifyOnChange == 'No' || notifyOnChange == 'False') {
    console.log('Notifications are not enabled for the Security Group in question.');
```

context.done();

return;

}

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
// Retrieve tags that need to be included in email body
var owner = message.configurationItem.tags["Owner"];
var ownerDL = message.configurationItem.tags["OwnerDL"];
var securityGroupId = message.configurationItem.resourceId;
var securityGroupName = message.configurationItem.tags["Name"];

// Verify email address in the OwnerDL tag is a valid email address
var emailRegex = /([\w-\+\.\.]+(?:\.\[\w-\+\.\.]+\])*)@((?:[\w-\+\.\.]+\w[\w-]{0,66})\.[a-z]{2,6}(?:\.\[a-z]{2})?)/i;
if (!ownerDL || ownerDL === "" || !emailRegex.test(ownerDL)) {
  console.log('The email address provided in the OwnerDL tag is either missing or invalid.');
```

context.done();

```
  return;
}
```

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
console.log("Owner: " + owner);
console.log("Owner DL: " + ownerDL);
console.log("Security Group ID: " + securityGroupId);

var sgIdentifiers = [securityGroupId];
if (securityGroupName && securityGroupName != "") {
    sgIdentifiers.push(securityGroupName);
}

// Construct the subject line
var subject = "Security Group " + securityGroupId + " Was Changed"

// Construct the email body (simple right now)
var emailMessage = 'The security group ' + sgIdentifiers.join('/') + ' was changed.\n\nDetails:\n';
emailMessage += JSON.stringify(message.configurationItemDiff, null, 2);

console.log("Email Body: " + emailMessage);

// Send the notification email and end function on callback
sendEmail(ownerDL, subject, emailMessage, function () {
    context.done(null, { success: true }); // SUCCESS with message
});
```

Serverless: lambda

Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
function sendEmail(toAddress, subject, message, callback) {  
    // Construct the parameters for the email  
    var emailParams = {  
        Destination: {  
            ToAddresses: [toAddress]  
        },  
        Message: {  
            Body: {  
                Text: {  
                    Data: message  
                }  
            },  
            Subject: {  
                Data: subject  
            }  
        },  
        Source: fromAddress  
    };  
    // Send the email using the parameters
```

Serverless: lambda

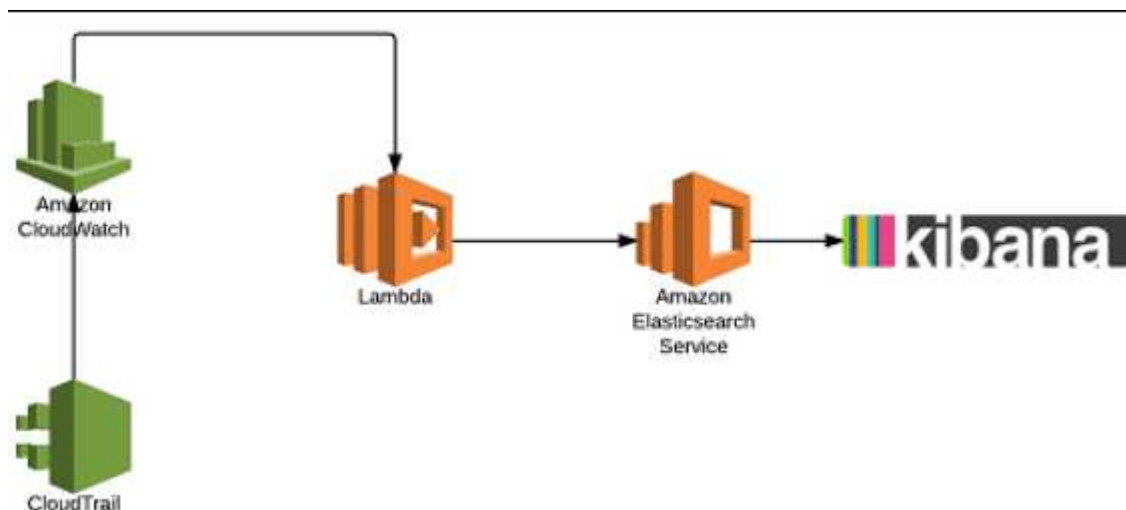
Ejercicio: Notificaciones vía Email de cambios en los grupos de seguridad combinado con AWS Config

- Código de la función Lambda

```
// Send the email using the parameters
var email = ses.sendEmail(emailParams, function (err, data) {
  if (err) {
    console.log('Error: ' + err);
    throw new Error(err);
  }
  else {
    console.log('Email sent successfully.');
```


Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Mediante CloudTrail se detectarán los cambios en los recursos
 - Logea todo el uso del API AWS
 - La visualización en CloudTrail es complicada
 - Mediante Cloudwatch recibiremos comunicaciones de CloudTrail y las enviaremos a una función Lambda que reenviara la información a Kibana mediante Elastic Search



Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - CloudTrail logea todas las acciones ocurridas en el API

	Event time	User name	Event name	Resource type
▶	2019-10-31, 11:01:48 AM	root	TerminateInstances	EC2 Instance
▶	2019-10-31, 11:01:18 AM	root	StopConfigurationR...	
▶	2019-10-31, 10:45:02 AM	root	StopInstances	EC2 Instance
▶	2019-10-31, 10:41:38 AM	root	StartInstances	EC2 Instance
▶	2019-10-31, 10:37:10 AM	root	StopInstances	EC2 Instance
▶	2019-10-31, 10:37:01 AM	root	AuthorizeSecurityGr...	EC2 SecurityGro...
▶	2019-10-31, 10:33:55 AM	root	SetTopicAttributes	SNS Topic
▶	2019-10-31, 10:33:55 AM	root	SetTopicAttributes	SNS Topic
▶	2019-10-31, 10:32:25 AM	root	AuthorizeSecurityGr...	EC2 SecurityGro...
▶	2019-10-31, 10:32:08 AM	root	RevokeSecurityGro...	EC2 SecurityGro...
▶	2019-10-31, 10:27:04 AM	root	StartInstances	EC2 Instance
▶	2019-10-31, 10:26:11 AM	root	StopInstances	EC2 Instance
▶	2019-10-31, 10:23:26 AM	root	StartInstances	EC2 Instance
▶	2019-10-31, 10:20:38 AM	root	StopInstances	EC2 Instance
▶	2019-10-31, 10:18:31 AM	root	RevokeSecurityGro...	EC2 SecurityGro...

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - CloudTrail logea todas las acciones ocurridas en el API

Event time	User name	Event name	Resource type	Resource name
2019-10-31, 11:01:48 AM	root	TerminateInstances	EC2 Instance	i-07f565421...

AWS access key	ASIAWD42LBCKLS4HEXWF	Event time	2019-10-31, 11:01:48 AM
AWS region	eu-west-1	Read only	false
Error code		Request ID	60fa0c54-3aef...
Event ID	9b531b18-b6f5-4724-a8d7-a52ed895f55d	Source IP address	193.146.78.97
Event name	TerminateInstances	User name	root
Event source	ec2.amazonaws.com		

You don't have any trails. With a trail, you can retain a record of events for your organization by logging in with the master account.

View Event

```

{
  "awsRegion": "eu-west-1",
  "sourceIPAddress": "193.146.78.97",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "groupId": "sg-0936b51c43ebbc732",
    "ipPermissions": {
      "items": [
        {
          "ipProtocol": "tcp",
          "fromPort": 80,
          "toPort": 80,
          "groups": {},
          "ipRanges": {
            "items": [
              {
                "cidrIp": "0.0.0.0/0"
              }
            ]
          }
        }
      ],
      "ipv6Ranges": {},
      "prefixListIds": {}
    }
  }
}

```

Event history

Your event history contains the activities taken by people, groups, or roles in your account. You can view the last 90 days of events. Choose an event to view more details.

Filter: Read only false

Event time	User name	Event name
2019-10-31, 10:33:55 AM	root	SetTrails
2019-10-31, 10:32:25 AM	root	AuthorizeSecurityGroupsInboundRules

Resources Referenced (1)

Resource type	Resource name
EC2 SecurityGroup	sg-0936b51c43ebbc732

[View event](#)

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Crear un Trail para todas las regiones
 - Los trail permitir crear métricas, lanzar eventos
 - En la sección de DataEvents logear todos las acciones sobre S3, recordar que CloudTrail logea todas las operaciones en un Bucket S3

CREATE TRAIL

Create trail

Trail name	Home region	Multi-region trail	Organization trail	S3 bucket	Log file prefix
CT-AWS	EU (Ireland)	Yes	No	ct-aws	

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Una vez creado el Trail , lo relacionaremos con CloudWatch
 - Seleccionamos el Trail y vamos a la sección CloudWatch Logs y configuramos
 - Y creamos un LogGroup : Ej. CloudTrail/CT-AWS
 - Concedemos permisos

▼ CloudWatch Logs		▼	CloudWatch Logs Log group	▼	Status
Log group	CloudTrail/CT-AWS		CloudTrail/CT-AWS		✓
IAM role	CloudTrail_CloudWatchLogs_Role				

[Create CloudWatch /](#)

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Ahora crearemos el recurso Amazon ES
 - Clickamos “Crear nuevo Dominio”

Nombre del dominio

El nombre del dominio debe comenzar con una letra minúscula y debe tener entre 1 y 28 caracteres (solo minúsculas, 0-9 y - (guion)).

S

El tipo de instancia que se corresponda con las necesidades de computación, memoria y almacenamiento de su aplicación, el número de fragmentos y réplicas, el tipo de consultas y el volumen de solicitudes. Para ver un resumen de los requisitos de cada instancia, consulte la documentación de Amazon ES.

Tipo de instancia

El tipo de instancia r5.large.elasticsearch necesita almacenamiento de EBS.

Número de instancias

Serverless: lambda

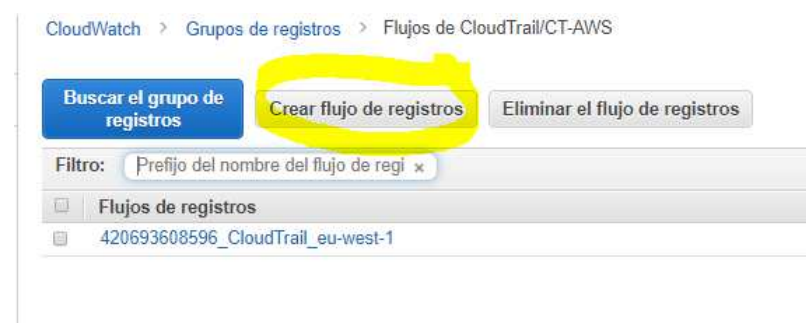
- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Una vez creado el Dominio ElasticSearch se activara el flujo de losg Cloudwatch a ElasticSearch

The screenshot displays the AWS Management Console interface for an ElasticSearch domain. On the left sidebar, under 'Panel', the 'Mis dominios' section is active, showing 'es-aws' with a status of 'Instancias reservadas'. The main content area is titled 'es-aws' and contains three buttons: 'Configurar el clúster' (highlighted in blue), 'Modificar la política de acceso', and 'Administrar'. Below these buttons are three tabs: 'Información general' (selected), 'Estado del clúster', and 'Estado de las instancias'. The 'Información general' tab displays the following details:

Estado del dominio	Activo
Versión de Elasticsearch	7.1
Punto de enlace	https://search-es-aws-kuc7v5y4snjbaupqtxmx
ARN del dominio	arn:aws:es:eu-west-1:420693608596:domain
Kibana	https://search-es-aws-kuc7v5y4snjbaupqtxmx


Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Una vez creado el Dominio ElasticSearch se activara el flujo de losg Cloudwatch a ElasticSearch
 - En CloudWatch crearemos un flujo de registro a ElastciSearch



Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - El flujo de registro a ElasticSearch se realiza mediante una función Lambda que la crea AWS automáticamente



The screenshot shows the AWS CloudTrail console. On the left, under 'Filtro: Prefijo del nombre del', there is a list of log groups: '/aws/lambda/prueba1', '/var/log/messages', 'CloudTrail/CT-AWS' (selected), 'RDSOSMetrics', and 'dev-VPCFlowLogsGroup'. A dropdown menu 'Acciones' is open, showing options: 'Crear un grupo de registros', 'Eliminar el grupo de registros', 'Exportar', 'Exportar datos a Amazon S3', 'Ver todas las exportaciones a Amazon S3', 'Suscripciones', 'Transmitir a AWS Lambda', 'Transmitir a Amazon Elasticsearch Service' (highlighted in yellow), and 'Eliminar el filtro de suscripción'.

Iniciar la transmisión de CloudTrail/CT-AWS a Amazon Elasticsearch Service

Está a punto de empezar a transmitir datos del grupo de registros "CloudTrail/CT-AWS" a un clúster de Amazon Elasticsearch Service. Todos los datos de registro nuevos enviados a este grupo de registros se enviarán al clúster que elija.

Seleccionar una cuenta ☒ Esta cuenta ☐ Otra cuenta

Clúster de Amazon ES* ⓘ

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Se crea un rol de ejecución

Amazon CloudWatch Logs is requesting permission to create an AWS Lambda

AWS Lambda uses an IAM role that grants your custom code permissions to access

▼ Ocultar detalles

Resumen de roles ?

Descripción del rol Lambda execution role permissions

Rol de IAM

Nombre de rol

► Ver documento de política

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Se selecciona el formato CloudTrail

Configurar el formato y los filtros de registros

Elija el formato de registro para obtener un patrón de filtro recomendado para los datos de registro o seleccione "Otro" para especificar un patrón de filtro personalizado. Un patrón de filtro vacío coincide con todos los eventos de registro.

Formato de registro*

Patrón del filtro de suscripción

Especifique la estructura de eventos de registro y las condiciones de filtro que se aplican a los datos de registro cuando se transmiten a Amazon Elasticsearch Service.

Seleccionar los datos de registro para probar

[Borrar](#) [Probar patrón](#)

```
{ "eventVersion": "1.05", "userIdentity": { "type": "Root", "principalId": "420693608596", "arn": "arn:aws:iam::420693608596:root" }, "eventTime": "2017-08-15T16:00:00Z", "eventSource": "aws:cloudtrail", "eventName": "CreateTrail", "awsRegion": "eu-west-1", "sourceIPAddress": "192.168.1.1", "userAgent": "aws-cli/2.0.0", "requestParameters": { "trailName": "CloudTrail-eu-west-1", "trailVersion": "1" }, "responseElements": { "trailName": "CloudTrail-eu-west-1", "trailVersion": "1" }, "errorMessage": null, "errorCode": null, "errorType": null }
```

Serverless: lambda

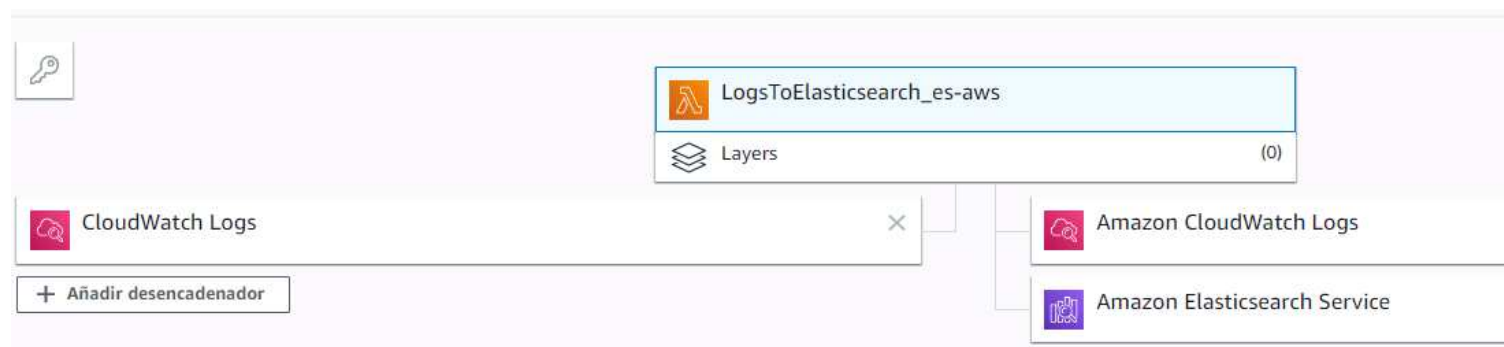
- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Se selecciona el formato CloudTrail
 - Y se crea el flujo junto con una función Lambda

▼ /var/log/messages	Explor:	Ninguna
● CloudTrail/CT-AWS	Explor:	Lambda (LogsToElasticsearch_es-aws)

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Ahora vemos en el Servicio Lambda la función creada automáticamente

Nombre de la función	Descripción
prueba1	
LogsToElasticsearch_es-aws	CloudWatch Logs to Amazon ES streaming



Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Por ultimo configuraremos Kibana para la visualización
 - Para ello abrimos el dominioElasticSearch creado y clickamos en el endpoint de Kibana

Estado del dominio Activo

Versión de Elasticsearch 7.1

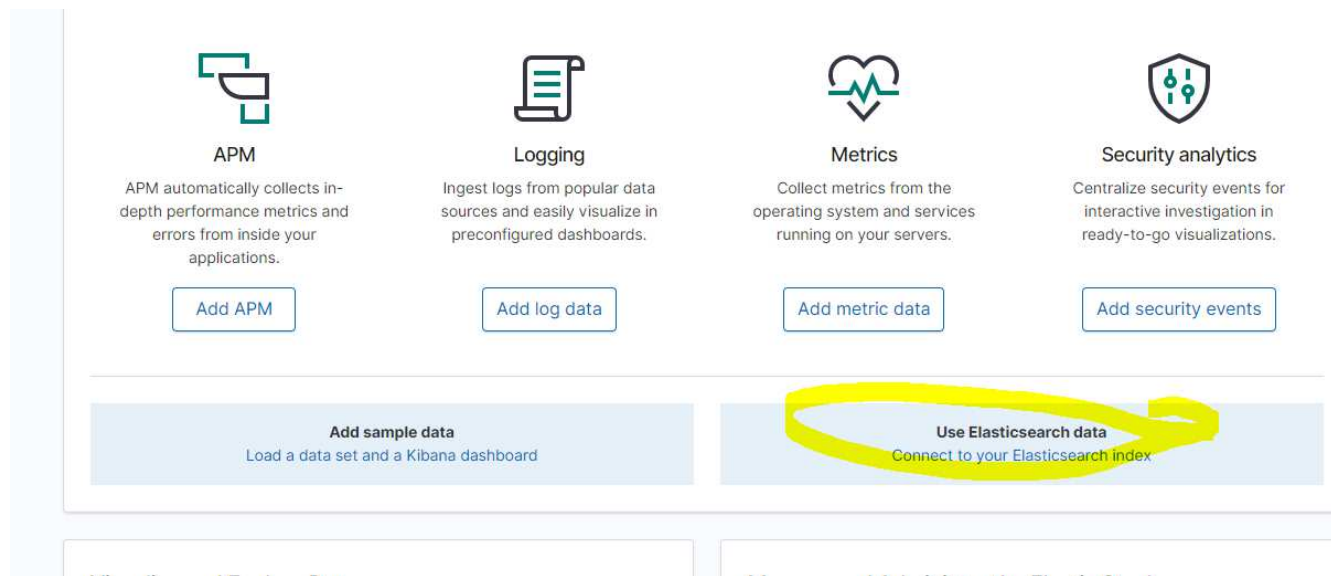
Punto de enlace <https://search-es-aws-kuc7v5y4snjbaupqtxmxr3ofva.eu-1>

ARN del dominio arn:aws:es:eu-west-1:420693608596:domain/es-aws

Kibana <https://search-es-aws-kuc7v5y4snjbaupqtxmxr3ofva.eu-1>

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Ahora conectaremos Kibana al index de ElasticSearch



Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Definir el patrón de índice cwl-*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch

Step 1 of 2: Define index pattern

Index pattern

cwl-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ **Success!** Your index pattern matches **1 index**.

cwl-2019.10.31

Rows per page: 10 ▾

Step 2 of 2: Configure settings

You've defined **cwl-*** as your index pattern. Now you can

Time Filter field name

Refresh

@timestamp ▾

The Time Filter will use this field to filter your data by time.

You can choose not to have a time field, but you will not be able to

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Definir el patrón de índice cwl-*

Create index pattern

Kibana uses index patterns to retrieve data from Elasticsearch

Step 1 of 2: Define index pattern

Index pattern

cwl-*

You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.

✓ **Success!** Your index pattern matches **1 index**.

cwl-2019.10.31

Rows per page: 10 ▾

Step 2 of 2: Configure settings

You've defined **cwl-*** as your index pattern. Now you can

Time Filter field name

Refresh

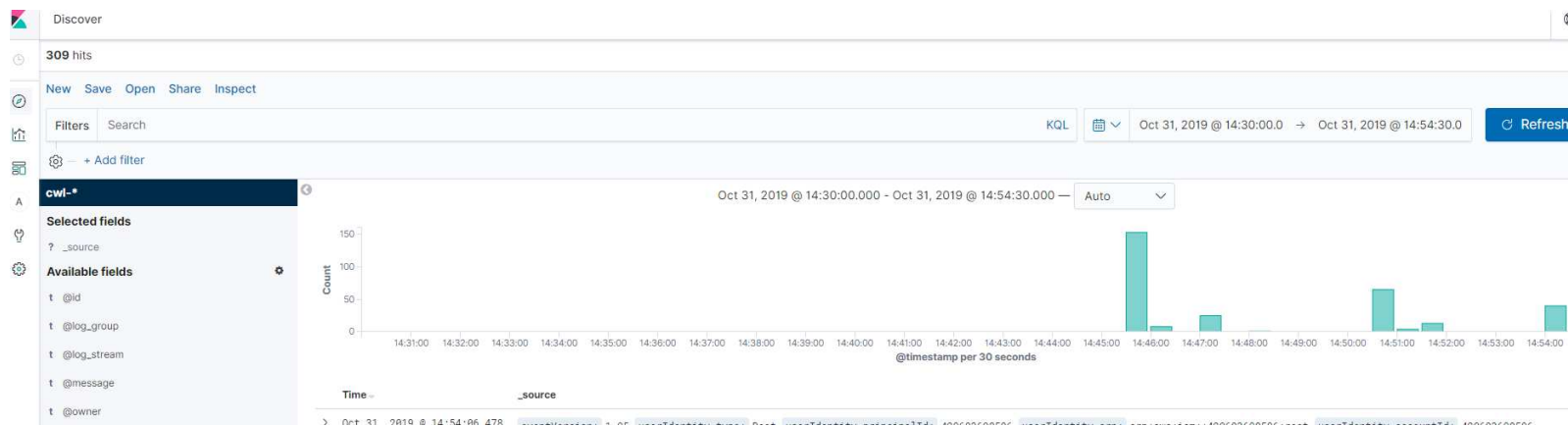
@timestamp ▾

The Time Filter will use this field to filter your data by time.

You can choose not to have a time field, but you will not be able to

Serverless: lambda

- Ejercicio: Monitorización de uso de API AWS en Kibana vía ElasticSearch
 - Una vez creado solo queda ir Discover



Serverless: lambda

- Ejemplos en QWIKSLAB

Serverless
Architectures with
Amazon DynamoDB
and Amazon Kinesis
Streams with AWS
Lambda

1 hour 10 Credits ★★★★★ Rate Lab
 training and certification

Monitoring Security
Groups with AWS
Config

57 minutes 10 Credits ★★★★★ Rate Lab
 training and certification

Esta es muy buena , un lambda para qu
emande mensjaes cuando se crea una
maquina

Serverless
Architectures using
Amazon CloudWatch
Events and Scheduled
Events with AWS
Lambda

1 hour 15 minutes 8 Credits ★★★★★ Rate Lab
SPL-119 Version 2.2.9


Hybrid Storage and
Data Migration with
AWS Storage Gateway
File Gateway

Creating an Amazon
Virtual Private Cloud
(VPC) with AWS
CloudFormation

1 hour 1 Credit ★★★★★ Rate Lab
 training and certification

Using AWS Lambda
with Amazon
CloudWatch and SNS
to Implement a Slack
Chat Bot

Creating a Serverless
Video Conversion
Watchfolder Workflow
for MediaConvert

2 hours 30 minutes 8 Credits ★★★★★ Rate Lab
 training and certification



**Mondragon
Unibertsitatea**

Goi Eskola
Politeknikoa

**Eskerrik asko
Muchas gracias
Thank you**