

CS2032 - Cloud Computing (Ciclo 2024-2)

Event-driven architecture

Semana 13 - Taller 1: SNS - Simple Notification Service

ELABORADO POR: GERALDO COLCHADO

Contenido

Event-driven architecture

1. Objetivo del taller 1
2. SNS - Simple Notification Service
3. Ejercicio 1: Evento Nuevo Alumno en UTEC
4. Ejercicio 2: Ejercicio propuesto
5. Cierre

Event-driven architecture

Objetivo del Taller 1

- Diseño e implementación de una Arquitectura de Solución basada en eventos con el servicio “SNS - Simple Notification Service”

Contenido

Event-driven architecture

1. Objetivo del taller 1
2. **SNS - Simple Notification Service**
3. Ejercicio 1: Evento Nuevo Alumno en UTEC
4. Ejercicio 2: Ejercicio propuesto
5. Cierre

Event-driven architecture

SNS - Simple Notification Service

Amazon Simple Notification Service

Notificaciones push móviles, correo electrónico, SMS y mensajería de publicación/suscripción completamente administrada

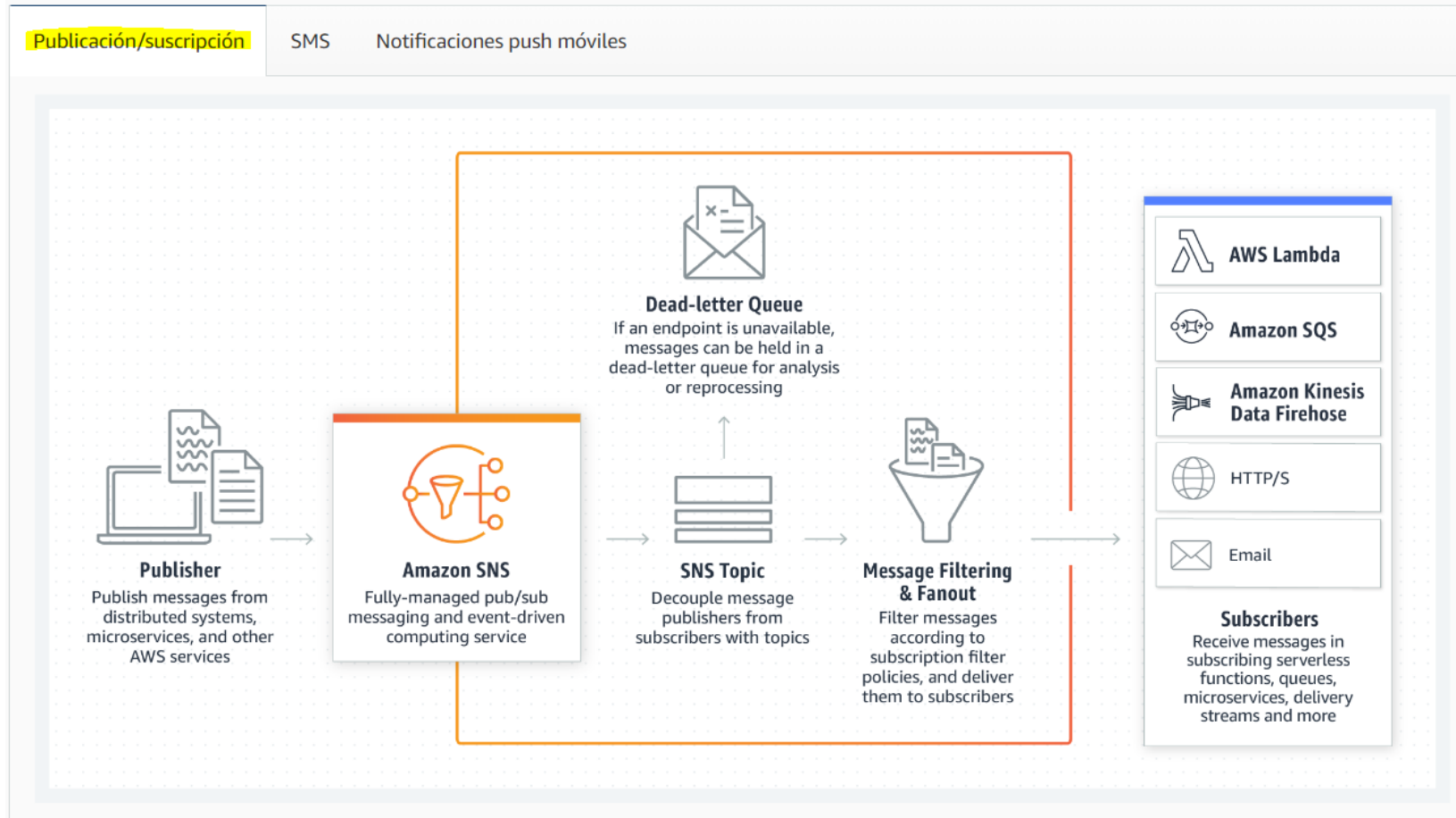
Introducción a Amazon SNS

Amazon Simple Notification Service (Amazon SNS) es un servicio de mensajería completamente administrado para la **comunicación aplicación a aplicación (A2A) y aplicación a persona (A2P)**.

La funcionalidad de publicación y suscripción A2A brinda temas para la mensajería de alto rendimiento, de muchos a muchos, basada en push entre sistemas distribuidos, microservicios y aplicaciones sin servidores controladas por eventos. Mediante el uso de temas de Amazon SNS, los sistemas de publicadores pueden distribuir los mensajes a una gran cantidad de sistemas de suscriptores, entre otros, colas de Amazon SQS, funciones de AWS Lambda, puntos de enlace HTTPS y Amazon Kinesis Data Firehose para procesamiento paralelo. La funcionalidad A2P permite enviar mensajes a usuarios a escala a través de SMS, push móvil y correo electrónico.

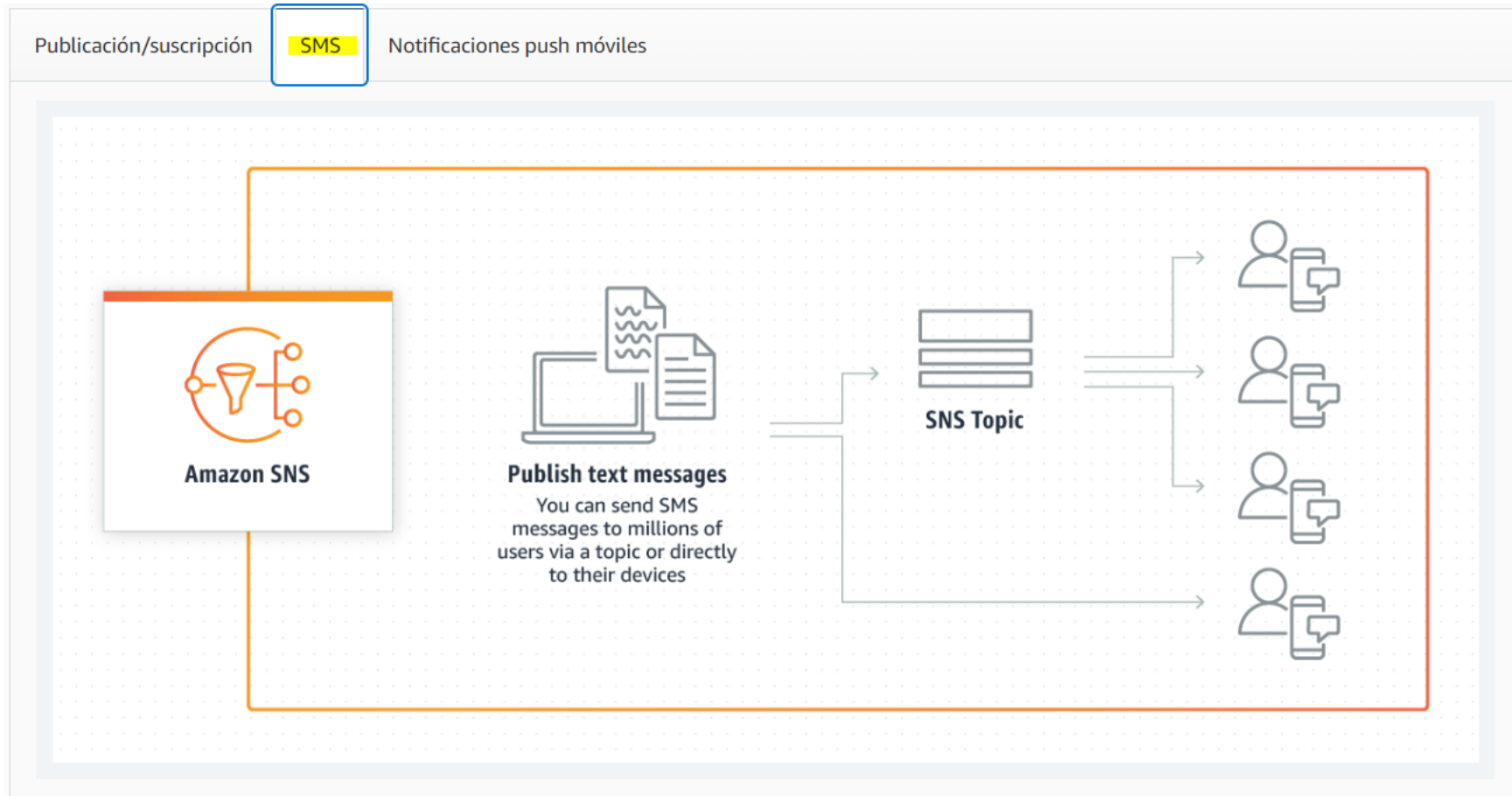
Event-driven architecture

SNS - Simple Notification Service



Event-driven architecture

SNS - Simple Notification Service



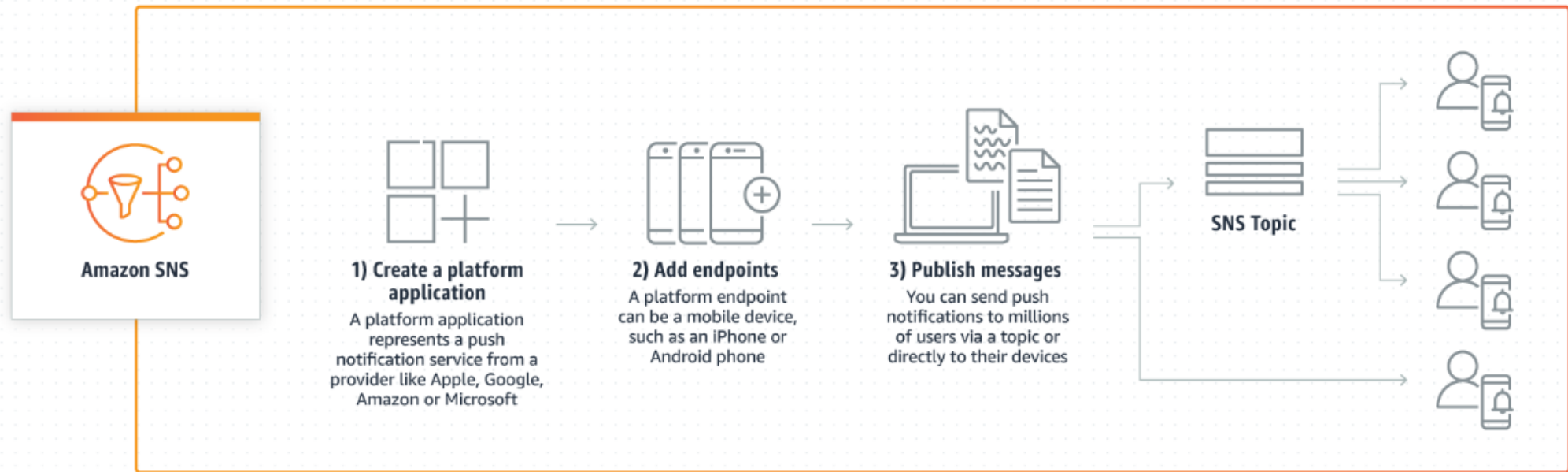
Event-driven architecture

SNS - Simple Notification Service

Publicación/suscripción

SMS

Notificaciones push móviles



Event-driven architecture

SNS - Simple Notification Service

Aplicación a aplicación (A2A)

Amazon SNS es un servicio de mensajería administrado que le permite desacoplar publicadores de suscriptores. Esto es útil para la mensajería entre aplicaciones para microservicios, sistemas distribuidos y aplicaciones sin servidor. [Más información](#)



Event-driven architecture

SNS - Simple Notification Service

Aplicación a persona (A2P)

Amazon SNS le permite enviar notificaciones de inserción a aplicaciones móviles, mensajes de texto a números de teléfonos móviles y correos electrónicos de texto sin formato a direcciones de correo electrónico. Puede distribuir mensajes con un tema o publicar en puntos de enlace móviles directamente. [Más información](#)



Contenido

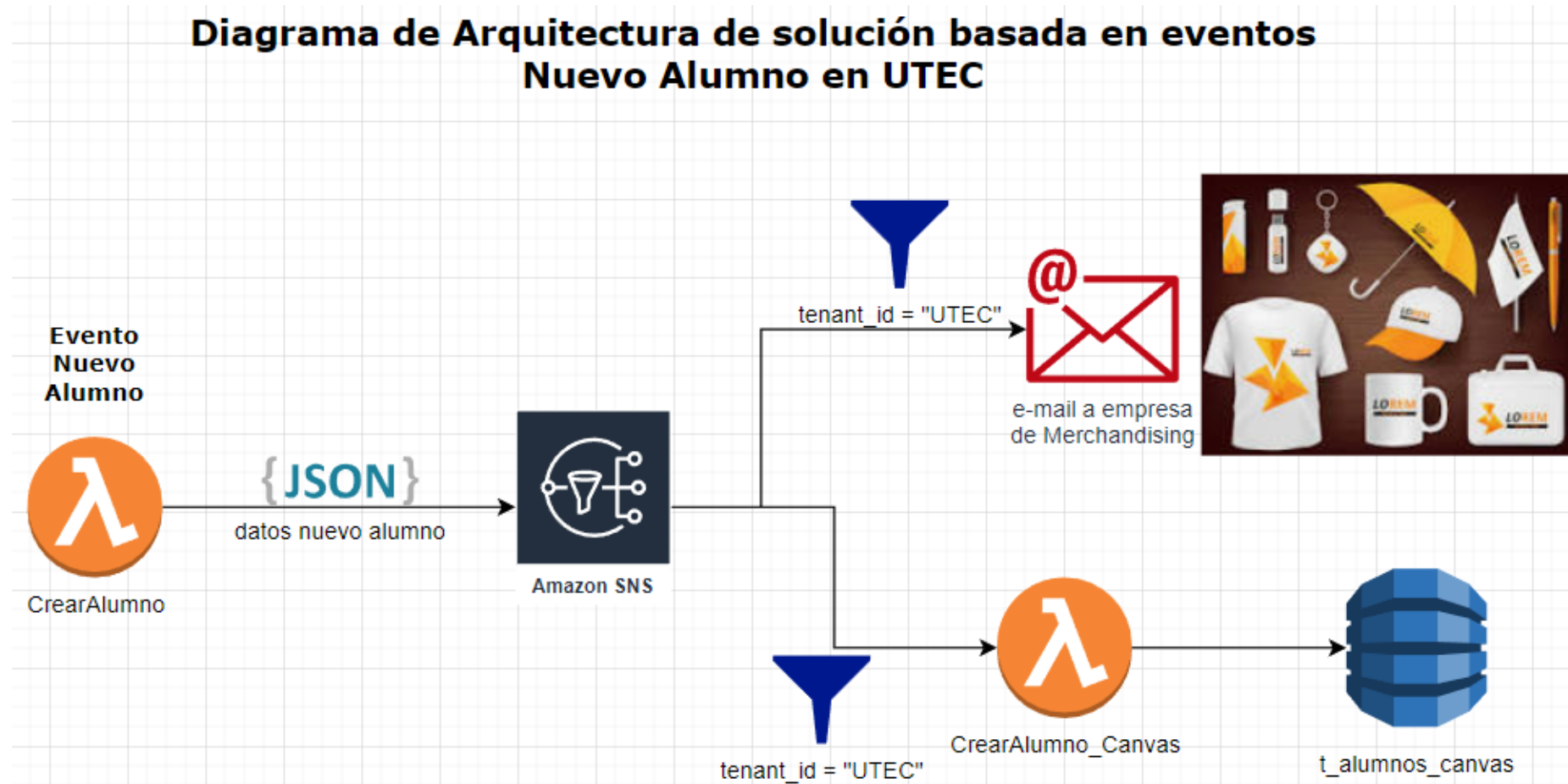
Event-driven architecture

1. Objetivo del taller 1
2. SNS - Simple Notification Service
3. **Ejercicio 1: Evento Nuevo Alumno en UTEC**
4. Ejercicio 2: Ejercicio propuesto
5. Cierre

Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

Implemente la siguiente arquitectura para procesar el evento “Nuevo Alumno en UTEC”



Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

Paso 1: Crear un tema “NuevoAlumno” en SNS

Crear un tema

Crear un tema

Detalles

Tipo Información

El tipo de tema no se puede modificar una vez que se crea el tema

☐ FIFO (primero en entrar, primero en salir)

- Clasificación de mensajes estrictamente conservada
- Entrega única de mensajes
- Rendimiento alto, hasta 300 publicaciones por segundo
- Protocolos de suscripción: SQS

☒ Estándar

- Clasificación de mensajes de mejor esfuerzo
- Entrega de mensajes al menos una vez
- Mayor rendimiento en publicaciones por segundo
- Protocolos de suscripción: SQS, Lambda, HTTP, SMS, correo electrónico, puntos de enlace de aplicaciones móviles

Nombre

NuevoAlumno

Máximo de 256 caracteres. Puede incluir caracteres alfanuméricos, guiones (-) y guiones bajos (_).

Crear un tema

Paso 2: Modificar lambda “CrearAlumno” para que escriba en tema “NuevoAlumno”

```
import json
```

```
import boto3
```

```
def lambda_handler(event, context):
```

```
...
```

```
# Publicar en SNS
```

```
sns_client = boto3.client('sns')
```

```
response_sns = sns_client.publish(
```

```
    TopicArn = 'arn:aws:sns:us-east-1:447891120606:NuevoAlumno',
```

```
    Subject = 'Nuevo Alumno',
```

```
    Message = json.dumps(alumno),
```

```
    MessageAttributes = {
```

```
        'tenant_id': {'DataType': 'String', 'StringValue': tenant_id }
```

```
    }
```

```
)
```

```
print(response_sns)
```

```
# Salida (json)
```

```
return {
```

```
    'statusCode': 200,
```

```
    'response': response
```

```
}
```

Evento
Nuevo
Alumno



CrearAlumno

{JSON}

datos nuevo alumno



Amazon SNS

Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

Paso 3: Crear una suscripción al tema “NuevoAlumno” con un correo electrónico (de preferencia de @gmail) y agregar un filtro para tenant_id = ‘UTEC’. Ingrese a su correo y confirme el enlace.

Crear una suscripción

Crear una suscripción

Detalles

ARN del tema

Protocolo

El tipo de punto de enlace para suscribirse

Punto de enlace

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

[Una vez creada la suscripción, debe confirmarla. Información](#)

Política de filtro de suscripciones - opcional

Esta política filtra los mensajes que recibe un suscriptor. [Información](#)

Política de filtro de suscripciones

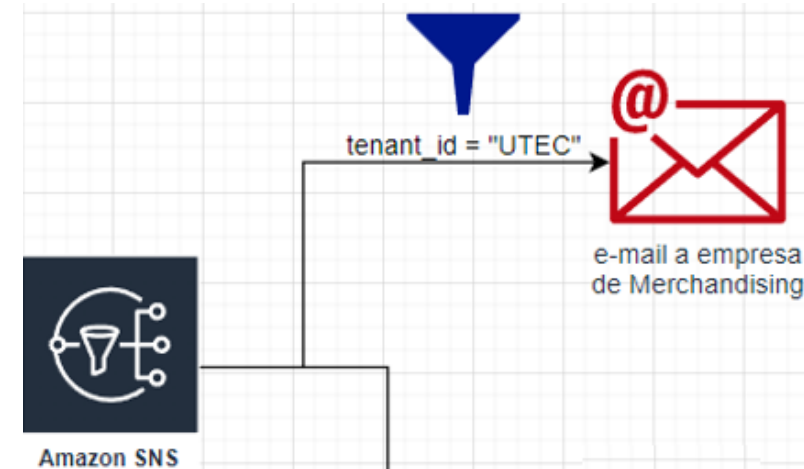
☒ **Habilitado**

☐ Desactivado

Editor de JSON

```
1 {
2   "tenant_id": [
3     "UTEC"
4   ]
5 }
```

Crear una suscripción

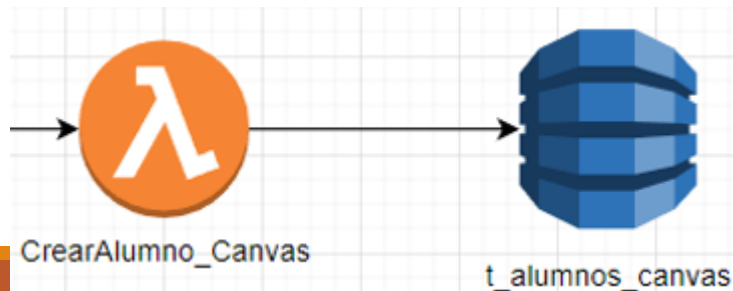


Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

Paso 4: Crear en DynamoDB una tabla **“t_alumnos_canvas”** con campos `tenant_id` (clave de partición), `alumno_id` (clave de ordenación), ambos campos de tipo string o cadena.

Paso 5: Crear un lambda **“CrearAlumno_Canvas”** que reciba un json con los datos del nuevo alumno y los grabe en la tabla **“t_alumnos_canvas”**



```
import json
import boto3

def lambda_handler(event, context):
    # Entrada (json)
    print(event) # Revisar en CloudWatch
    alumno_json = json.loads(event['Records'][0]['Sns']['Message'])
    # Proceso
    dynamodb = boto3.resource('dynamodb')
    table = dynamodb.Table('t_alumnos_canvas')
    alumno = {
        'tenant_id': alumno_json['tenant_id'],
        'alumno_id': alumno_json['alumno_id'],
        'alumno_nombre': alumno_json['alumno_datos']['nombre']
    }
    print(alumno) # Revisar en CloudWatch
    response = table.put_item(Item=alumno)
    # Salida (json)
    return {
        'statusCode': 200,
        'response': response
    }
```

Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

Paso 6: Crear una suscripción al tema SNS “NuevoAlumno” con el lambda “CrearAlumno_Canvas” y agregue el filtro `tenant_id = 'UTEC'`

Crear una suscripción

Detalles

ARN del tema

arn:aws:sns:us-east-1:447891120606:NuevoAlumno

Protocolo

El tipo de punto de enlace para suscribirse

AWS Lambda

Punto de enlace

Una función AWS Lambda que puede recibir notificaciones de Amazon SNS.

arn:aws:lambda:us-east-1:447891120606:function:CrearAlumno_Canvas

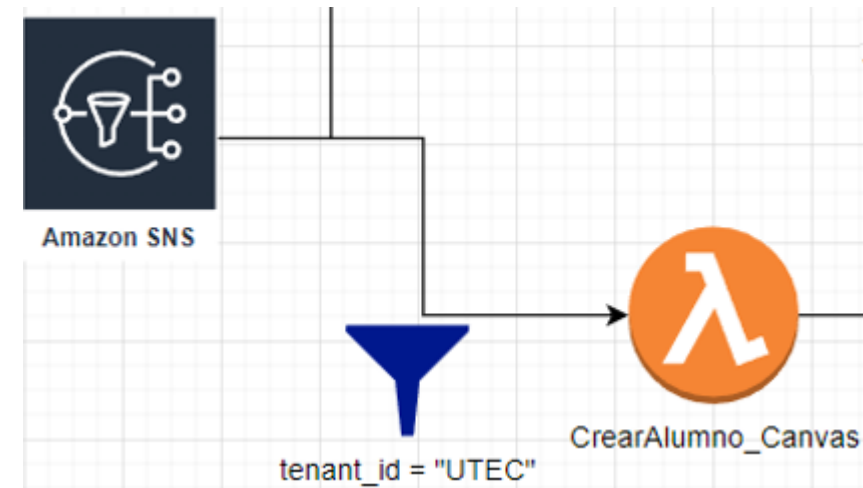
Política de filtro de suscripciones

☒ Habilitado

☐ Desactivado

Editor de JSON

```
1 {  
2   "tenant_id": [  
3     "UTEC"  
4   ]  
5 }
```



Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

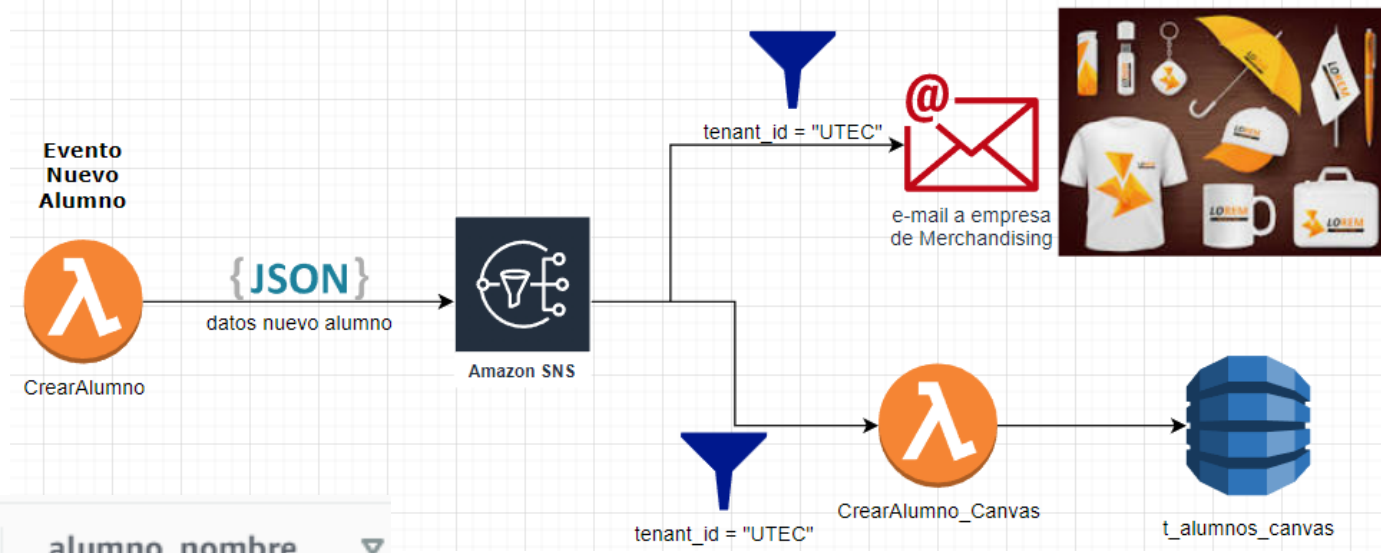
Paso 7: Probar el lambda “CrearAlumno” y verificar que se envíe un correo electrónico y que se grabe un registro en la tabla t_alumnos_canvas

```
{
  "tenant_id": "UTEC",
  "alumno_id": "202199991",
  "alumno_datos": {
    "nombre": "Juan Colchado",
    "sexo": "M",
    "fecha_nac": "2000-09-01",
    "celular": "999736332",
    "domicilio": {
      "direcc": "Av. Javier Prado
158",
      "distrito": "San Isidro",
      "provincia": "Lima",
      "departamento": "Lima",
      "pais": "Perú"
    }
  }
}
```

t_alumnos_canvas

<input type="checkbox"/>	tenant_id	alumno_id	alumno_nombre
<input type="checkbox"/>	UTEC	202199991	Juan Colchado

Diagrama de Arquitectura de solución basada en eventos
Nuevo Alumno en UTEC



Event-driven architecture

Ejercicio 1 - Evento Nuevo Alumno en UTEC

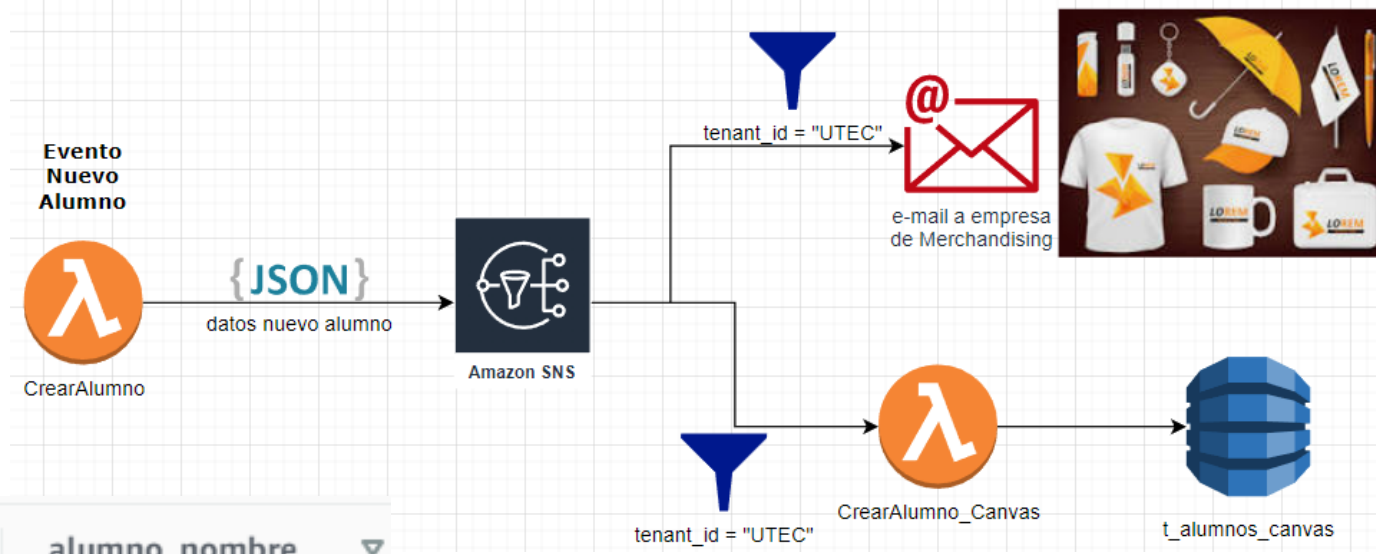
Paso 8: Probar el lambda “CrearAlumno” y verificar que **no** se envíe un correo electrónico y que **no** se grabe un registro en la tabla t_alumnos_canvas por ser de **tenant_id = 'UNIV2'**

```
{
  "tenant_id": "UNIV2",
  "alumno_id": "2021G64747",
  "alumno_datos": {
    "nombre": "Juan Carrasco",
    "sexo": "M",
    "fecha_nac": "2000-09-01",
    "celular": "999736332",
    "domicilio": {
      "direcc": "Av. Javier Prado
158",
      "distrito": "San Isidro",
      "provincia": "Lima",
      "departamento": "Lima",
      "pais": "Perú"
    }
  }
}
```

t_alumnos_canvas

<input type="checkbox"/>	tenant_id	alumno_id	alumno_nombre
<input type="checkbox"/>	UTEC	202199991	Juan Colchado

Diagrama de Arquitectura de solución basada en eventos
Nuevo Alumno en UTEC



Contenido

Event-driven architecture

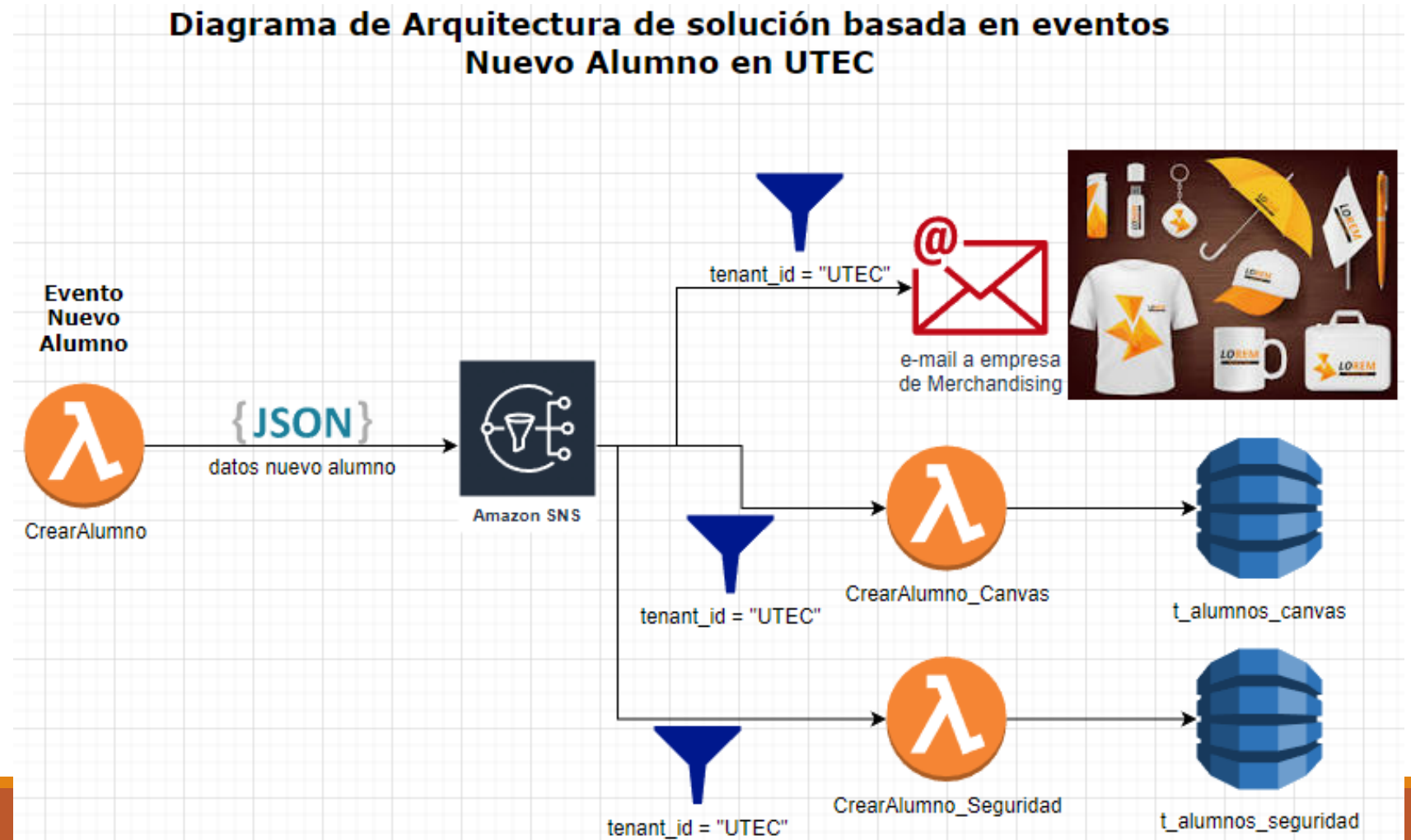
1. Objetivo del taller 1
2. SNS - Simple Notification Service
3. Ejercicio 1: Evento Nuevo Alumno en UTEC
4. **Ejercicio 2: Ejercicio propuesto**
5. Cierre

Event-driven architecture

Propuesto - Evento Nuevo Alumno en UTEC

Adicione un subscriber para notificar al sistema de seguridad de UTEC.

Suba su evidencia al padlet indicado por el docente.



Contenido

Event-driven architecture

1. Objetivo del taller 1
2. SNS - Simple Notification Service
3. Ejercicio 1: Evento Nuevo Alumno en UTEC
4. Ejercicio 2: Ejercicio propuesto
5. **Cierre**

Cierre:

Event-driven architecture - Qué aprendimos?

- Diseño e implementación de una Arquitectura de Solución basada en eventos con el servicio “SNS - Simple Notification Service”

Gracias

Elaborado por docente: Geraldo Colchado