

Actividad: AWS Lambda

Administración de sistemas
operativos

Tarea 1: crear una función de Lambda _____ **3**

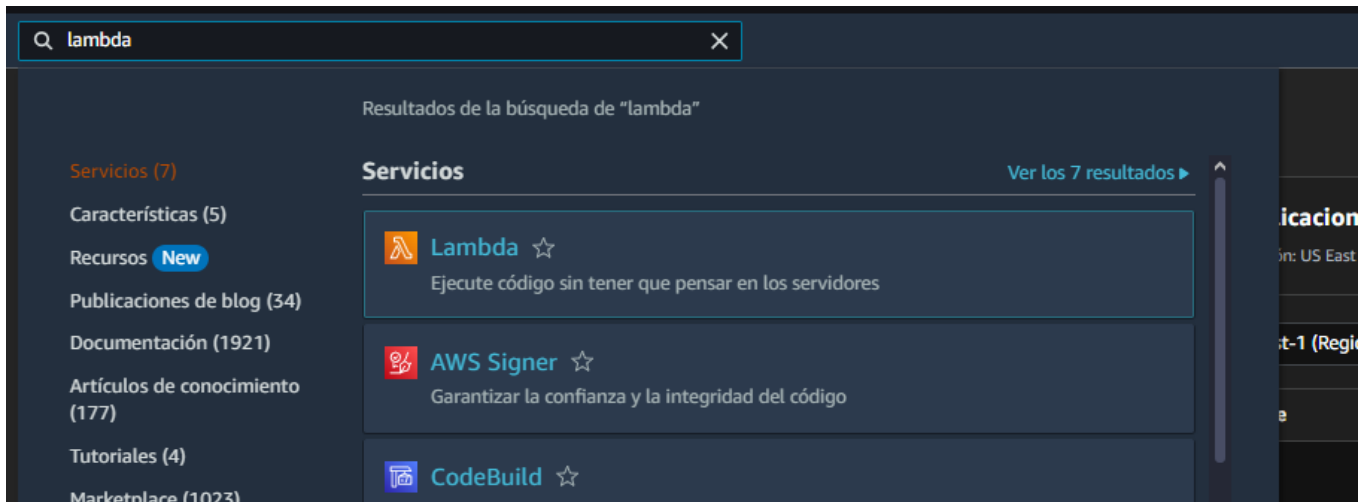
Tarea 2: configurar el desencadenador _____ **5**

Tarea 3: configurar la función de Lambda _____ **6**

Tarea 4: Verificar que la función de Lambda funcionó _____ **9**

Tarea 1: crear una función de Lambda

Primero buscaremos Lambda en el laboratorio



Una vez dentro le daremos a Crear una función y ponemos la configuración tal que así:

Información básica

Nombre de la función

Escriba un nombre para describir el propósito de la función.

myStopinator

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Tiempo de ejecución Información

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Python 3.11

Arquitectura Información

Elija la arquitectura del conjunto de instrucciones que desea para el código de la función.

☒ x86_64

☐ arm64

Permisos Información

De forma predeterminada, Lambda creará un rol de ejecución con permisos para cargar registros en Amazon CloudWatch Logs. Puede personalizar este rol predeterminado más adelante al agregar los disparadores.

▼ Cambiar el rol de ejecución predeterminado

Rol de ejecución

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

☐ 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

Rol existente

Seleccione un rol existente que haya creado para usarlo con esta función de Lambda. El rol debe tener permiso para cargar registros en Amazon CloudWatch Logs.

myStopinatorRole

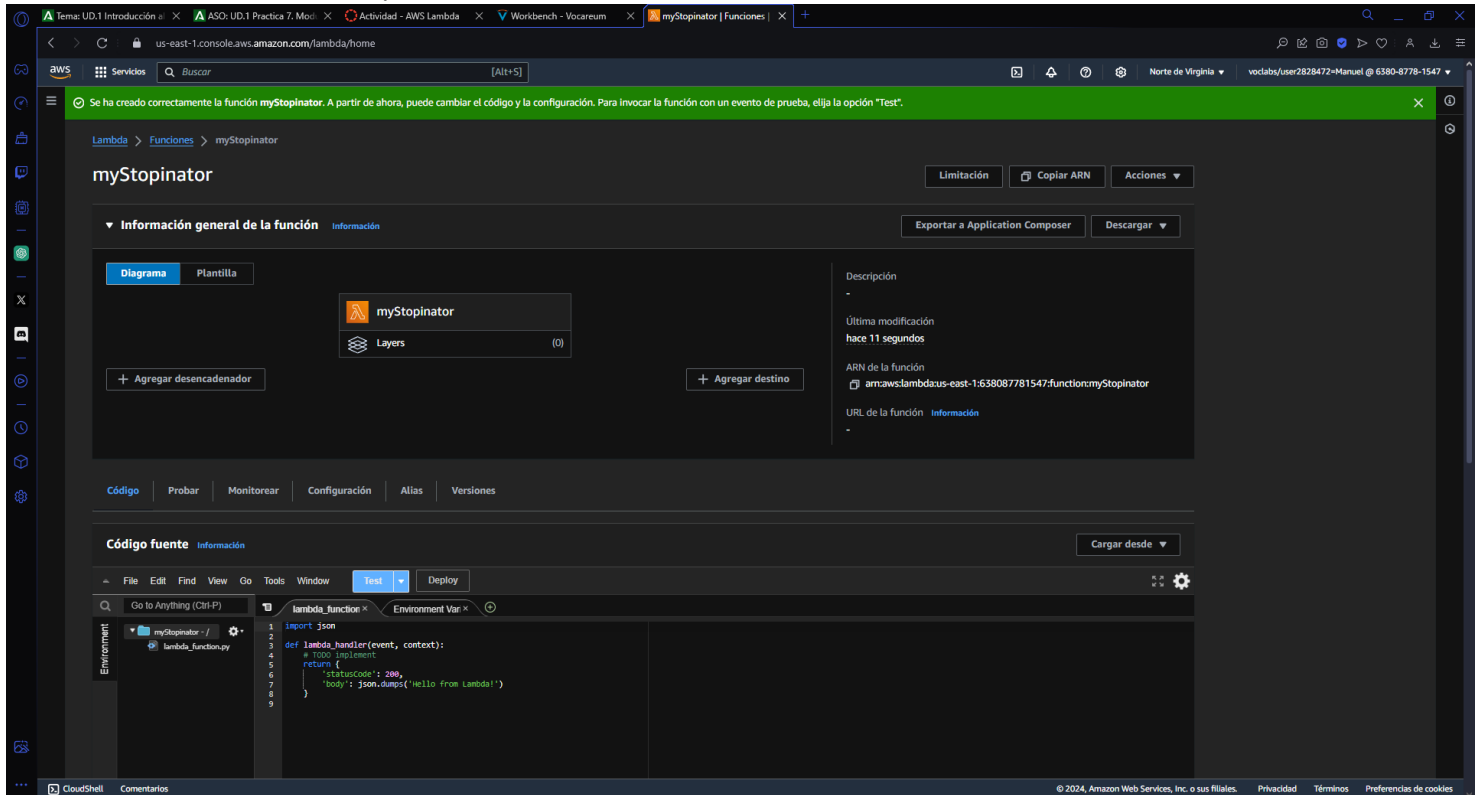
[Consulte el rol myStopinatorRole](#) en la consola de IAM.

► Additional Configurations

Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

4

Le damos a crear y nos saldrá esto:



Tarea 2: configurar el desencadenador

Cliqueamos en agregar desencadenador y ponemos la siguiente configuración:

Lambda > Añadir desencadenadores

Agregar desencadenador

Configuración del desencadenador

Información

EventBridge (CloudWatch Events)

aws asynchronous schedule management-tools

Regla

Seleccione una regla existente o cree una nueva.

Cree una regla nueva

Reglas existentes

Nombre de la regla

Especifique un nombre que identifique la regla de forma inequívoca.

everyMinute

Descripción de regla

Proporcione una descripción opcional para la regla.

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

Desencadene automáticamente el destino según una programación automatizada utilizando expresiones Cron o rate Las expresiones Cron están en UTC.

rate(1 minute)

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

Lambda añadirá los permisos necesarios para Amazon EventBridge (CloudWatch Events) para invocar la función Lambda desde este desencadenador. [Obtenga más información](#) sobre el modelo de permisos de Lambda.

Cancelar Agregar

▼ Información general de la función

Información

Exportar a Application Composer Descargar ▼

Diagrama

Plantilla

myStopinator

Layers (0)

EventBridge (CloudWatch Events)

+ Agregar destino

+ Agregar desencadenador

Descripción

-

Última modificación

hace 49 minutos

ARN de la función

arn:aws:lambda:us-east-1:638087781547:function:myStopinator

URL de la función

Información

-

Código

Probar

Monitorear

Configuración

Alias

Versiones

Configuración general

Desencadenadores

Permisos

Destinos

URL de la función

Variables de entorno

Etiquetas

Desencadenadores (1)

Información

Corregir errores

Editar

Eliminar

Agregar desencadenador

Buscar desencadenadores

< 1 >

Desencadenador

EventBridge (CloudWatch Events): everyMinute

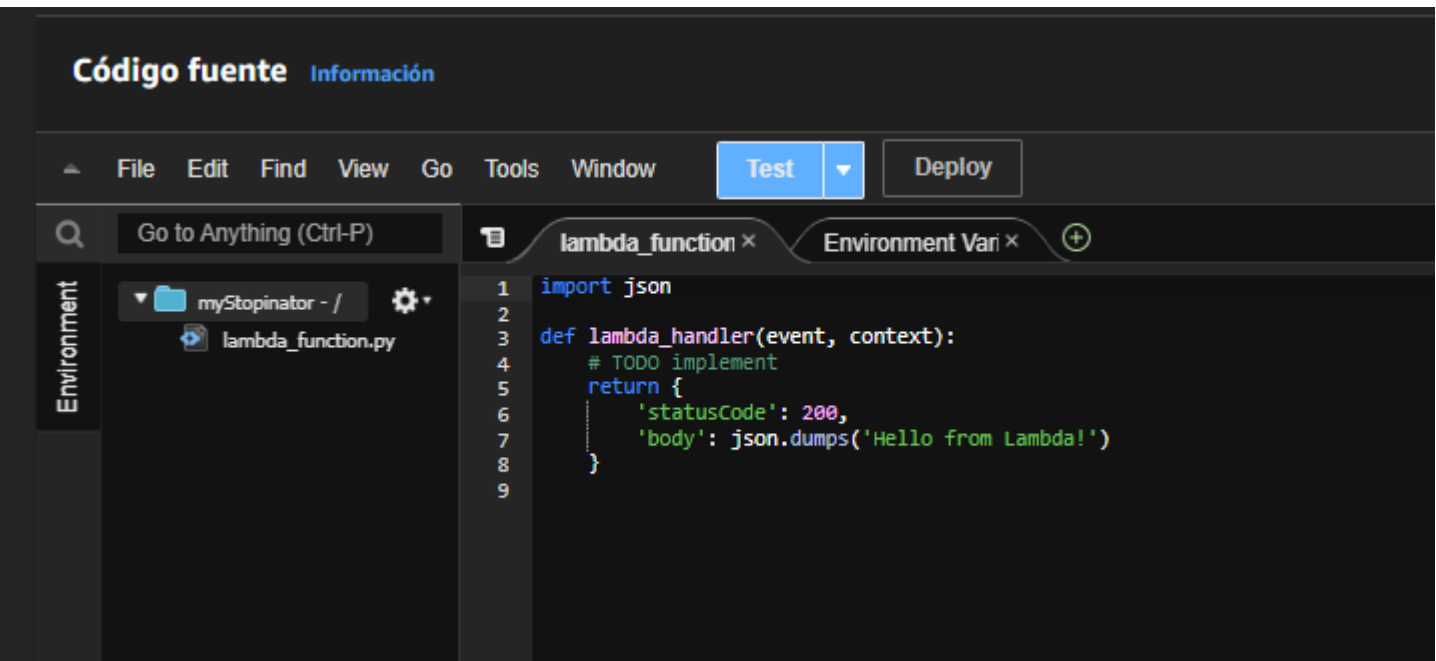
arn:aws:events:us-east-1:638087781547:rule/everyMinute

Estado de la regla: ENABLED

Detalles

Tarea 3: configurar la función de Lambda

Para esta tarea seleccionaremos el código y lo borraremos



Una vez borrado pegaremos lo siguiente:

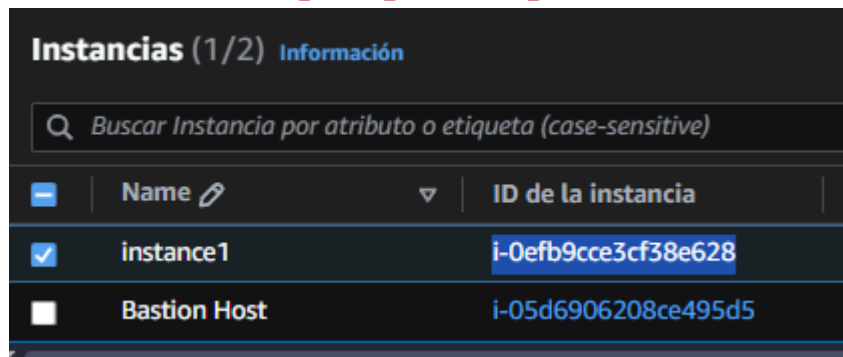
```
import boto3
region = '<REPLACE_WITH_REGION>'
instances = ['<REPLACE_WITH_INSTANCE_ID>']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

Y cambiaremos lo siguiente:

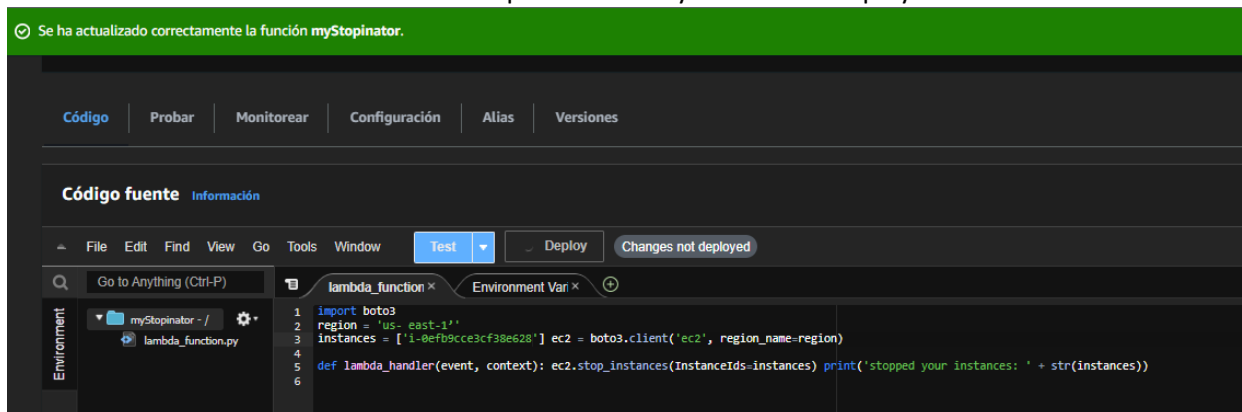
En **<REPLACE_WITH_REGION>** pondremos la región en la que estamos, en este caso **'us-east-1'**

Después comprobaremos si tenemos la instancia llamada instance1 ejecutada, y reemplazaremos la id en **<REPLACE_WITH_INSTANCE_ID>**



	Name	ID de la instancia
<input checked="" type="checkbox"/>	instance1	i-0efb9cce3cf38e628
<input type="checkbox"/>	Bastion Host	i-05d6906208ce495d5

Seleccionamos en el menú de File la opción de Save y le damos a Deploy



Tarea 4: Verificar que la función de Lambda funcionó

Y aquí vemos que se ha detenido la instancia

Instancias (1/2) Información					
<input type="text" value="Buscar Instancia por atributo o etiqueta (case-sensitive)"/>					Todos los e... ▼
<input type="checkbox"/>	Name	ID de la instancia	Estado de la i... ▼	Tipo de inst... ▼	Co
<input checked="" type="checkbox"/>	instance1	i-0efb9cce3cf38e628	Detenida	t2.micro	-
<input type="checkbox"/>	Bastion Host	i-05d6906208ce495d5	En ejecución	t2.micro	