



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

**INFORME DE LABORATORIO N° 10
“CONSTRUYENDO UNA
ARQUITECTURA SERVERLESS PARA EL
ANÁLISIS DE SENTIMIENTOS DE TWEETS EN
TIEMPO REAL”**

Curso: Inteligencia de Negocios

Docente: Ing. Patrick Cuadros Quiroga

Anahua Huayhua, Jenny Karen (2018062150)

**Tacna – Perú
2022**

PRACTICA DE LABORATORIO N° 10**TEMA: ANÁLISIS DE SENTIMIENTOS**

Realizar los siguientes pasos para el laboratorio



1) Entrar a la consola de AWS y acceder al servicio de Cloud9.

Crear una carpeta para el laboratorio y subir los siguientes archivos:

ReadTwitter.py, lambda-es.zip, LabES-Resources.yaml, docker.zip

Creemos un bucket para subir el zip, ejecutamos el comando en el terminal del Cloud9.

aws s3 mb s3://aws-code-lambda-zip-xxxx

Subimos el zip al bucket que hemos creado en S3.

aws s3 cp lambda-es.zip s3://aws-code-lambda-zip-xxxx/zip

Modificaremos el archivo LabES-Resources.yaml Confirmamos la suscripción al tópico de SNS, entramos a nuestro correo. Lanzaremos el stack de CloudFormation . Ejecutamos en el terminal el stack.

aws cloudformation create-stack --stack-name Stack-LabES-Recursos --template-body file://LabES-Resources.yaml --capabilities CAPABILITY_NAMED_IAM

Visualizamos en CloudFormation que se estén creando los recursos del stack.

Instalamos en el terminal la librería tweepy.

pip3 install tweepy --user



Para hacer una prueba de nuestros tokens de Twitter, ejecutaremos lo siguiente, pero antes debemos agregar nuestros tokens dentro el código.

python3 ReadTwitter.py covid

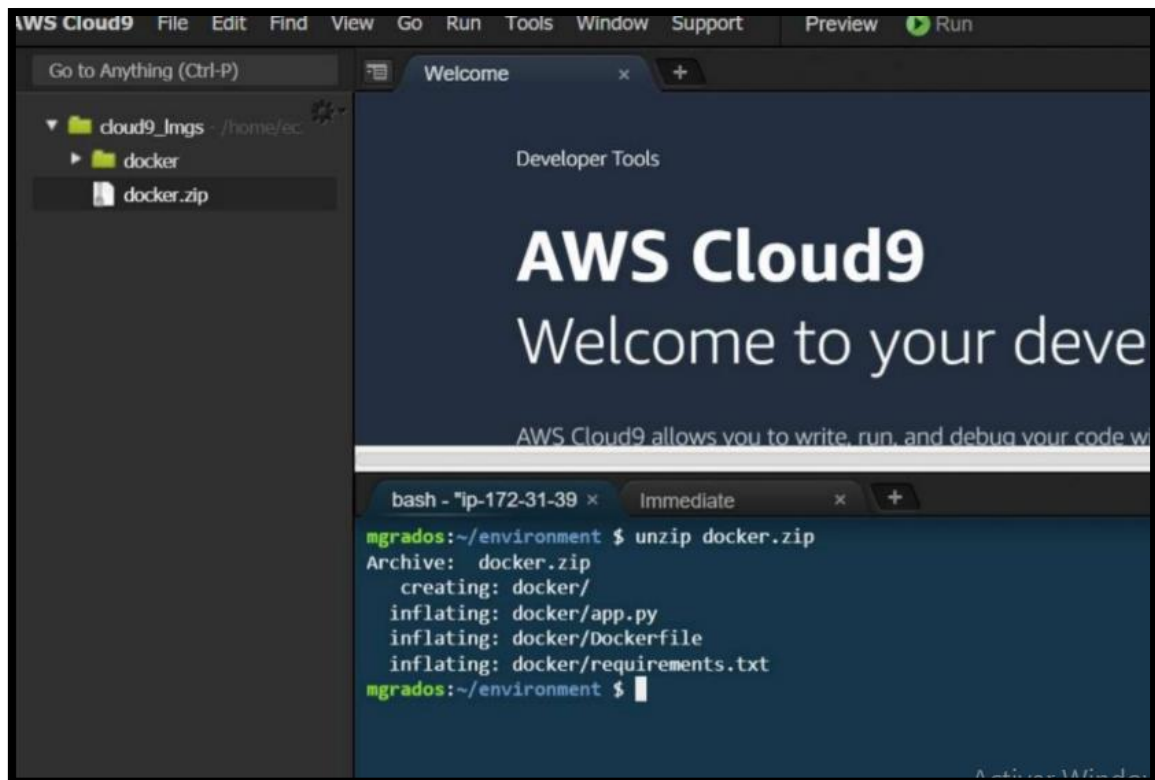
```
bash - "ip-172-31-52" x bash - "ip-172-31-52" x python3 - "ip-172-31-52" x +
ec2-user:~/environment/13ES $ python3 ReadTwitter.py covid
covid
[INFO] El usuario styxNstonz (Tiene 414 seguidores y sigue a 436), vive en None ha comentado sobre el hashtag #covid
[INFO] El usuario Terry Fleischman (Tiene 33 seguidores y sigue a 16), vive en None ha comentado sobre el hashtag #covid
[INFO] El usuario edgar jimenez (Tiene 106 seguidores y sigue a 91), vive en None ha comentado sobre el hashtag #covid
[INFO] El usuario Patrick Henningsen (Tiene 50922 seguidores y sigue a 2608), vive en USA ha comentado sobre el hashtag #covid
```

Descomprimos el archivo docker.zip

unzip docker.zip

cd docker

Entramos a la carpeta docker.



Entramos al servicio de ECR desde la consola y veremos nuestro repo creado.

Copiamos el URI del repositorio.

En mi caso: 971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets

Volvemos a Cloud9

cd docker

Construiremos la imagen Docker.

```
ec2-user:~/environment/13ES/docker $ docker build -t ecr-app-get-tweets:v1
Sending build context to Docker daemon 6.144kB
Step 1/7 : FROM python:3.7
----> e4e55e98f1e0
Step 2/7 : WORKDIR /usr/src/app
----> Using cache
----> 001eacfc696a
Step 3/7 : RUN cd /usr/src/app
----> Using cache
----> 2a8973b4508e
Step 4/7 : COPY requirements.txt /usr/src/app
----> cf3edfa5e1d1
Step 5/7 : COPY app.py /usr/src/app
----> f3ddbbeda191
```

Creamos un tag a nuestro contenedor.

```
docker tag ecr-app-get-tweets:v1 971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets:v1
```

Nos logueamos al ECR

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --password-stdin
971489366207.dkr.ecr.us-east-1.amazonaws.com
```

Subimos la imagen al ECR.

```
ec2-user:~/environment/13ES/docker $ docker push 971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets:v1
The push refers to repository [971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets]
a9262ca13260: Pushed
b4e7d8bf040b: Pushed
6eed352f02a8: Pushed
0a5651c2a552: Pushed
f8d89ba0399b: Pushed
2be8fac4a55f: Pushed
dabd0b48256e: Pushed
98d95bdfa037: Pushed
da9418a2e1b1: Pushing [=====>] 434.5MB/509.9MB
```

2) Regresamos el ECR y obtenemos el URI de la imagen subida.

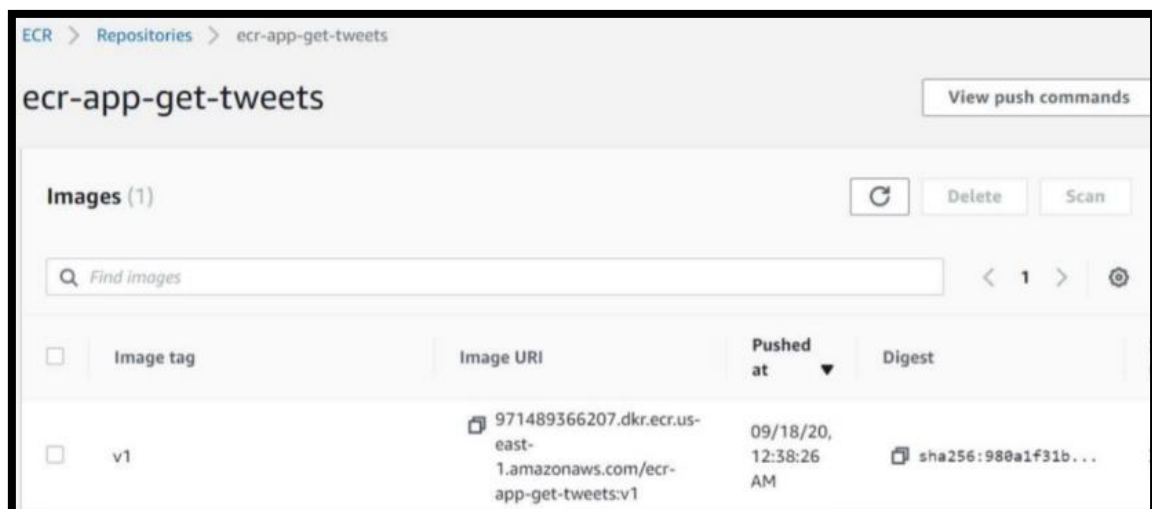
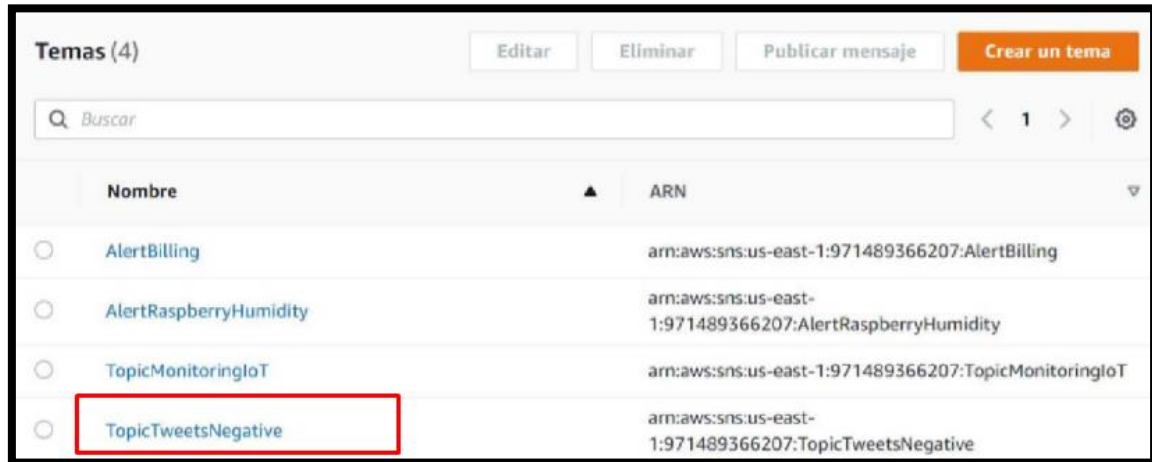


Image tag	Image URI	Pushed at	Digest
<input type="checkbox"/> v1	971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets:v1	09/18/20, 12:38:26 AM	sha256:980a1f31b...



3) Entramos al servicio de SNS.

Seleccionamos el tópic `TopicTweetsNegative` y obtenemos su ARN.



4) Entramos a los servicios de las funciones Lambda

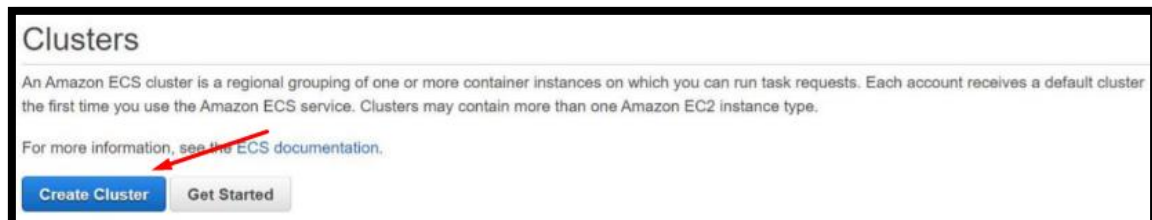
Clic en la función Lambda llamada : `DetectSentimentTweets`.

Modificamos la línea 78 y pegamos el valor del ARN del tópic de SNS.

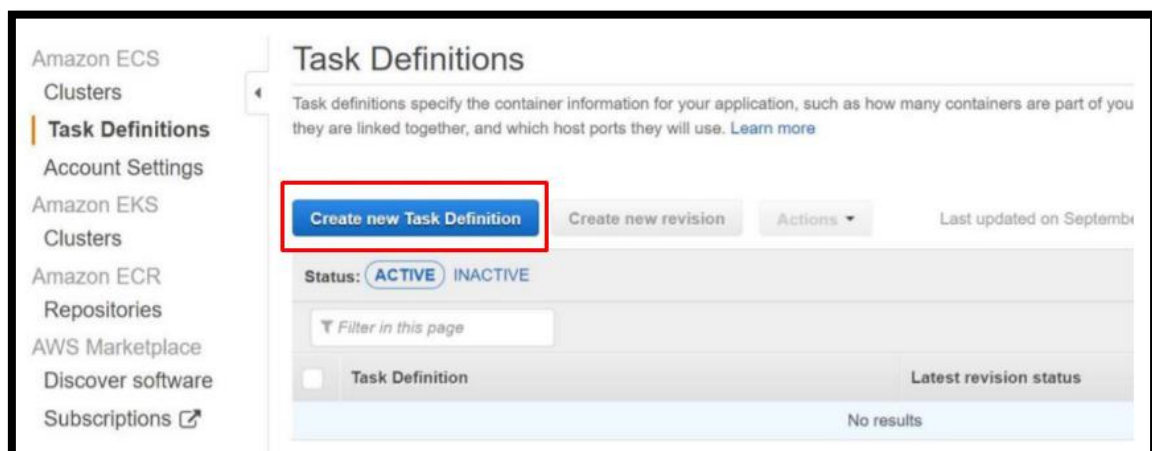
Modificamos también la línea 12, con el valor de la url de nuestro clúster de ElasticSearch.

5) Entramos al servicio de Fargate

Creamos un clúster en ECS (no genera costo), clic en Get Started. Clic en Next.



Creamos una definición de tarea.



Elegir un nombre para el task definition.

Agregar el contenedor a desplegar en Fargate.



Add container

Standard

Container name* container-tweets ⓘ

Image* 971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets:v1 ⓘ

Private repository authentication* ☐ ⓘ

Memory Limits (MiB) Soft limit ▼ 128 ⓘ
[+ Add Hard limit](#)

Ejecutar el task definition, clic en Action -> Run task.

Task Definition: task-app-tweets:1

View detailed information for your task definition. To modify the task definition, you need to create a new revision and then run the task definition

Create new revision Actions ▼

Builder JSON Tags

Task Definition Name task-app-tweets

Task Role TweetsTaskRole

Network Mode awsvpc ⓘ
If you choose <default>, ECS will start your container using Docker's default networking mode, which is Bridge on Linux and NAT on Windows. <default> is the only supported mode on Windows.

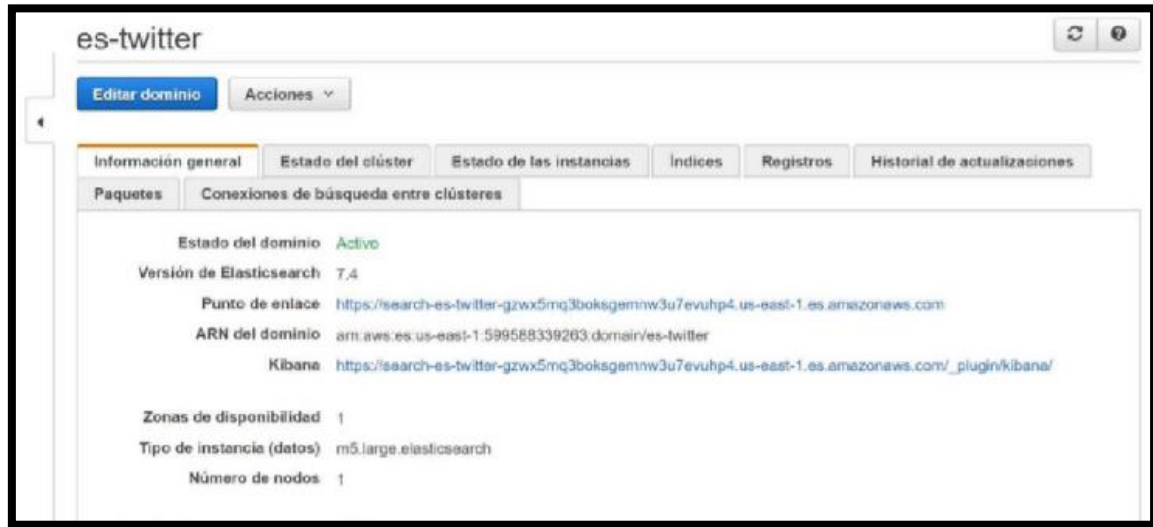
Elegir el lanzamiento por Fargate.

Launch type ☒ FARGATE ☐ EC2 ⓘ



Ahora acceder al servicio de ElasticSearch.

Clic en Acciones -> Modificar política de acceso



Conclusión: se desarrolló el laboratorio según los pasos señalados usando servicios como cloud9, ECR, elastic Search ,SNS entre otros