

UNIVERSIDAD PRIVADA DE TACNA



FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistema

**Informe de laboratorio 10: Construyendo una
Arquitectura Serverless**

Curso: Inteligencia de negocios

DOCENTE: Ing. Patrick Cuadros Quiroga

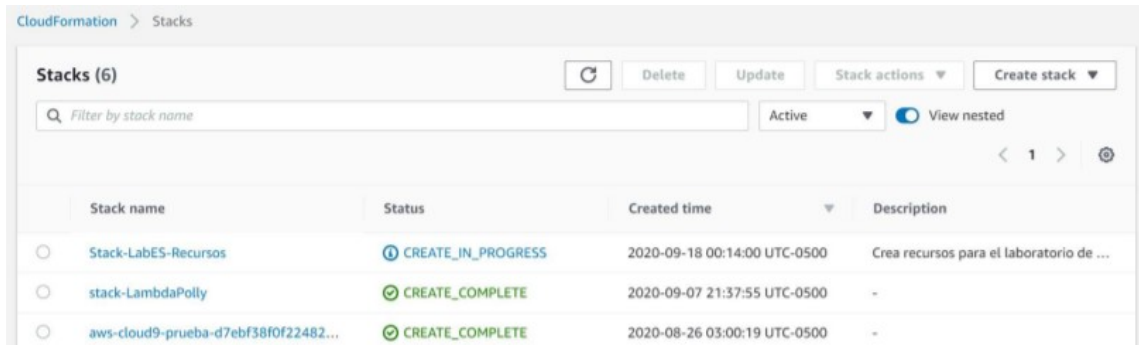
**Alumno: Balcon Coahila, Edwart Juan
(2013046516)**

Tacna – Perú

2021

1. Realizar los siguientes pasos para el laboratorio

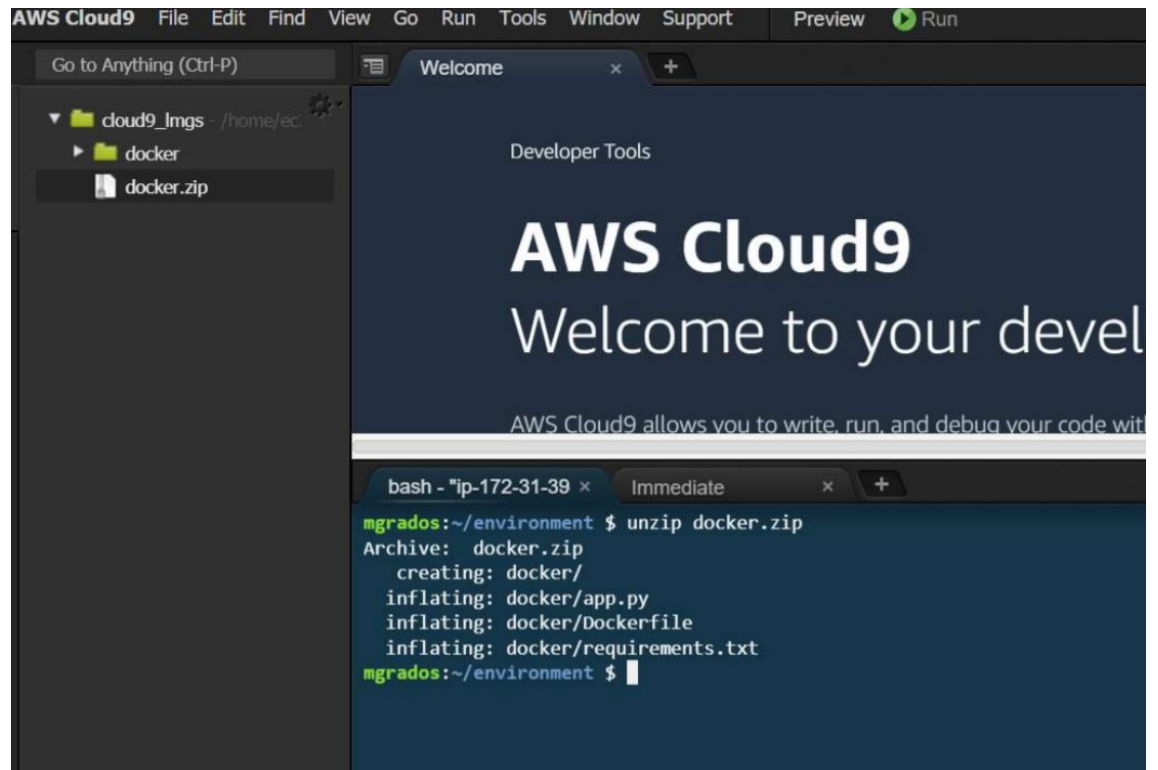
1.1. Visualizamos en CloudFormation que se estén creando los recursos del stack.



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


```
bash - "ip-172-31-52" x bash - "ip-172-31-52" x python3 - "ip-172-31" 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
```


1.3. Entramos a la carpeta docker.




1.4. Entramos al servicio de ECR desde la consola y veremos nuestro repo creado. Copiamos el URI del repositorio.

ECR > Repositories

Repositories (1)  [View push commands](#) [Delete](#) [Edit](#) [Create repository](#)

< 1 > 

	Repository name ▲	URI	Created at ▼	Tag immutability	Scan on push	Encryption type
<input type="radio"/>	ecr-app-get-tweets	 971489366207.dkr.ecr.us-east-1.amazonaws.com/ecr-app-get-tweets	09/18/20, 12:14:07 AM	Disabled	Disabled	AES-256

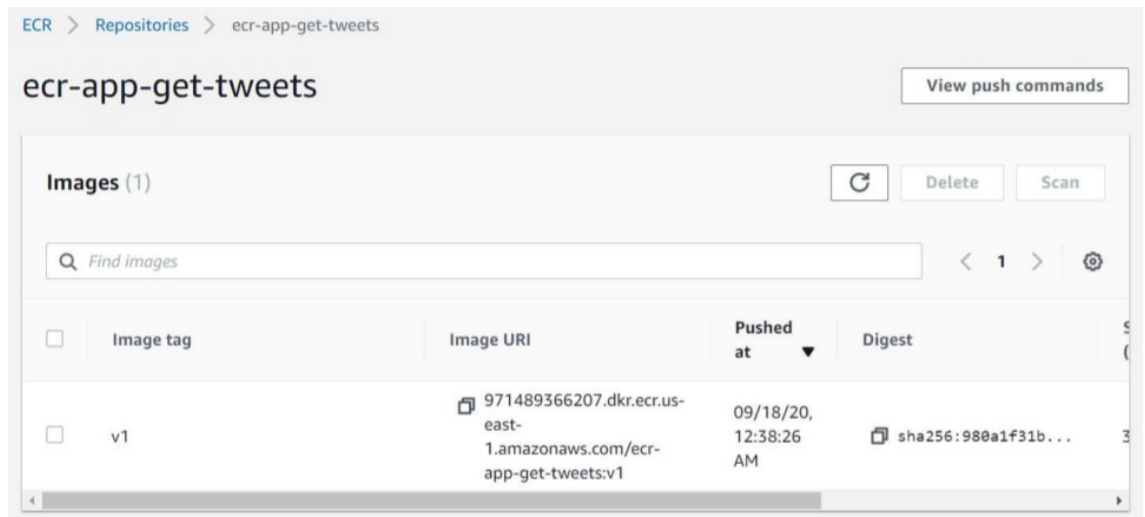
1.5. Construiremos la imagen Docker.

```
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
```

1.6. 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
2e5b4ca91984: Pushed
527ade4639e0: Pushed
c2c789d2d3c5: Pushed
8803ef42039d: Pushed
```

1.7. Regresamos el ECR y obtenemos el URI de la imagen subida.



1.8. Seleccionamos el t pico TopicTweetsNegative y obtenemos su ARN y Clic en la funci n Lambda llamada : DetectSentimentTweets.

Temas (4)		Editar	Eliminar	Publicar mensaje	Crear un tema
<input type="text" value="Buscar"/>		< 1 >			
Nombre		ARN			
<input type="radio"/>	AlertBilling	arn:aws:sns:us-east-1:971489366207:AlertBilling			
<input type="radio"/>	AlertRaspberryHumidity	arn:aws:sns:us-east-1:971489366207:AlertRaspberryHumidity			
<input type="radio"/>	TopicMonitoringIoT	arn:aws:sns:us-east-1:971489366207:TopicMonitoringIoT			
<input type="radio"/>	TopicTweetsNegative	arn:aws:sns:us-east-1:971489366207:TopicTweetsNegative			

es-twitter



Editar dominio

Acciones

Informaci n general

Estado del cl ster

Estado de las instancias

 ndices

Registros

Historial de actualizaciones

Paquetes

Conexiones de b squeda entre cl steres

Estado del dominio Activo

Versi n de Elasticsearch 7.7

Punto de enlace <https://search-es-twitter-s3o54h66ndespandwdznrkpugm.us-east-1.es.amazonaws.com>

ARN del dominio arn:aws:es:us-east-1:971489366207:domain/es-twitter

Kibana https://search-es-twitter-s3o54h66ndespandwdznrkpugm.us-east-1.es.amazonaws.com/_plugin/kibana/

Zonas de disponibilidad 1

Tipo de instancia (datos) t2.small.elasticsearch

N mero de nodos 1

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

Clusters

An Amazon ECS cluster is a regional grouping of one or more container instances on which you can run task requests. Each account receives a default cluster the first time you use the Amazon ECS service. Clusters may contain more than one Amazon EC2 instance type.

For more information, see the [ECS documentation](#).

Create Cluster

Get Started

View



list



card

[view all](#)



0 - 0 of 0



No clusters found

Get Started



1.10. Clic en Next

Getting Started with Amazon Elastic Container Service (Amazon ECS) using Fargate

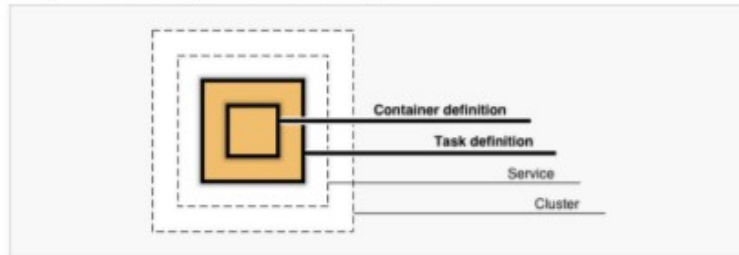
Step 1: Container and Task

Step 2: Service

Step 3: Cluster

Step 4: Review

Diagram of ECS objects and how they relate



Container definition

Edit

Choose an image for your container below to get started quickly or define the container image to use.

sample-app

image : httpd:2.4
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

nginx

image : nginx:latest
memory : 0.5GB (512)
cpu : 0.25 vCPU (256)

tomcat-webserver

image : tomcat
memory : 2GB (2048)
cpu : 1 vCPU (1024)

custom

image : --
memory : --
cpu : --

Configure

Task definition

Edit

A task definition is a blueprint for your application, and describes one or more containers through attributes. Some attributes are configured at the task level but the majority of attributes are configured per container.

Task definition name	first-run-task-definition	?
Network mode	awsvpc	?
Task execution role	Create new	?
Compatibilities	FARGATE	?
Task memory	0.5GB (512)	
Task CPU	0.25 vCPU (256)	

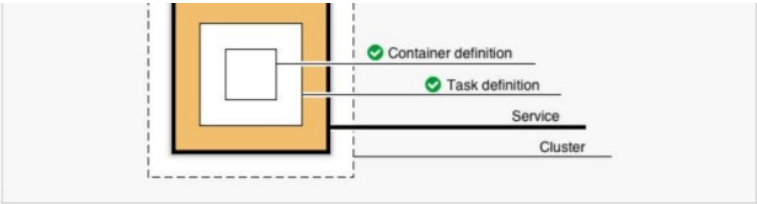
*Required

Cancel

Next

1.11. Clic en Next.

Step 4: Review



Define your service Edit

A service allows you to run and maintain a specified number (the "desired count") of simultaneous instances of a task definition in an ECS cluster.

Service name

Number of desired tasks

Security group
A security group is created to allow all public traffic to your service only on the container port specified. You can further configure security groups and network access outside of this wizard.

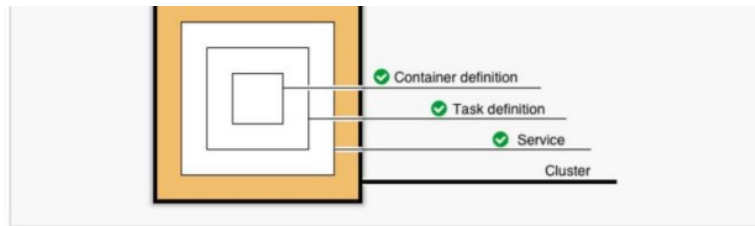
Load balancer type ☒ None ☐ Application Load Balancer

*Required Cancel Previous Next

1.12. Clic en Next.

Step 3: Cluster

Step 4: Review



Configure your cluster

The infrastructure in a Fargate cluster is fully managed by AWS. Your containers run without you managing and configuring individual Amazon EC2 instances.

To see key differences between Fargate and standard ECS clusters, see the [Amazon ECS documentation](#).

Cluster name

Cluster names are unique per account per region. Up to 255 letters (uppercase and lowercase), numbers, and hyphens are allowed.

VPC ID ⓘ

Subnets ⓘ

*Required

[Cancel](#)

[Previous](#)

[Next](#)

1.13. Clic en Create

Task execution role [Create new](#)

Container name [sample-app](#)

Image [httpd:2.4](#)

Memory [512](#)

Port [80](#)

Protocol [HTTP](#)

Service [Edit](#)

Service name [sample-app-service](#)

Number of desired tasks [1](#)

Cluster [Edit](#)

Cluster name [default](#)

VPC ID [Automatically create new](#)

Subnets [Automatically create new](#)

*Required [Cancel](#) [Previous](#) [Create](#)

1.14. Creamos una definición de tarea.

Amazon ECS

Clusters

Task Definitions

Account Settings

Amazon EKS

Clusters

Amazon ECR

Repositories

AWS Marketplace

Discover software

Subscriptions [↗](#)

Task Definitions

Task definitions specify the container information for your application, such as how many containers are part of you they are linked together, and which host ports they will use. [Learn more](#)

[Create new Task Definition](#)
[Create new revision](#)
[Actions ▾](#)
Last updated on September

Status: [ACTIVE](#) [INACTIVE](#)

<input type="checkbox"/>	Task Definition	Latest revision status
No results		

1.15. Clic en Fargate

Create new Task Definition

Step 1: Select launch type compatibility

Step 2: Configure task and container definitions

Select launch type compatibility

Select which launch type you want your task definition to be compatible with based on where your task runs.



1.16. Elegir un nombre para el task definition.

Configure task and container definitions

A task definition specifies which containers are included in your task and how they interact with each other. You can also specify data volumes for your containers to use. [Learn more](#)

Task Definition Name* task-app-tweets ⓘ

Requires Compatibilities* FARGATE

Task Role TweetsTaskRole ⓘ

Optional IAM role that tasks can use to make API requests to authorized AWS services. Create an Amazon Elastic Container Service Task Role in the [IAM Console](#) ⓘ

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.

Task execution IAM role

This role is required by tasks to pull container images and publish container logs to Amazon CloudWatch on your behalf. If you do not have the `ecsTaskExecutionRole` already, we can create one for you.

Task execution role TweetsTaskExecutionRole ?

Task size

The task size allows you to specify a fixed size for your task. Task size is required for tasks using the Fargate launch type and is optional for the EC2 launch type. Container level memory settings are optional when task size is set. Task size is not supported for Windows containers.

Task memory (GB) 0.5GB

The valid memory range for 0.25 vCPU is: 0.5GB - 2GB.

Task CPU (vCPU) 0.25 vCPU

The valid CPU for 0.5 GB memory is: 0.25 vCPU

Task memory maximum allocation for container memory reservation



1.17. 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

Define hard and/or soft memory limits in MiB for your container. Hard and soft limits correspond to the 'memory' and 'memoryReservation'

* Required

Cancel

Add

1.18. Clic en crear.

Launch Status

Task definition status - 2 of 2 completed

Create Task Definition: task-app-tweets

task-app-tweets succeeded

Create CloudWatch Log Group



CloudWatch Log Group created

CloudWatch Log Group [/ecs/task-app-tweets](#)

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

[Task Definitions](#) > [task-app-tweets](#) > 1

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 modify 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.

1.20. Elegir el lanzamiento por Fargate.

Run Task

Select the cluster to run your task definition on and the number of copies of that task to run. To apply configurations, click Advanced Options.

Launch type ☒ FARGATE ☐ EC2 

[Switch to capacity provider strategy](#) 

Task Definition Family
task-app-tweets ▼
Revision
1 ▼

Platform version LATEST ▼ 

Cluster default ▼

VPC and security groups

VPC and security groups are configurable when your task definition uses the awsvpc network mode.

Cluster VPC* vpc-e2b7e098 (172.31.0.0/16) 

Subnets* 

subnet-77d4ef10
(172.31.0.0/20) - us-east-1a
assign ipv6 on creation: Disabled 

subnet-e47978ca
(172.31.80.0/20) - us-east-1b
assign ipv6 on creation: Disabled 



Security groups* task-a-8980  [Edit](#)

Auto-assign public IP ENABLED 

► Advanced Options

1.21. El task definition ya se está ejecutando.

Cluster ARN: `arn:aws:ecs:us-east-1:971489366207:cluster/default`

Status: **ACTIVE**

Registered container instances: 0

Pending tasks count: 1 Fargate, 0 EC2

Running tasks count: 0 Fargate, 0 EC2

Active service count: 0 Fargate, 0 EC2

Draining service count: 0 Fargate, 0 EC2

Services | **Tasks** | ECS Instances | Metrics | Scheduled Tasks | Tags | Capacity Providers

Run new Task | Stop | Stop All | Actions ▾ | Last updated on September 18, 2020 1:34:06 AM (0m ago) | ↺ | ⓘ

Desired task status: **Running** | Stopped

Filter in this page | Launch type: ALL | < 1-1 > | Page size: 50 ▾

<input type="checkbox"/>	Task	Task defi...	Contain...	Last stat...	Desired ...	Started ...	Started ...	Group	Launch t...	Platform...
<input type="checkbox"/>	a6d1df96...	task-app-...	--	PROVISI...	RUNNING			family:tas...	FARGATE	1.3.0

1.22. Ahora acceder al servicio de ElasticSearch. Clic en Acciones
-¿Modificar política de acceso

Panel

Mis dominios

es-twitter

Instancias reservadas

Paquetes

es-twitter | ↺ | ⓘ

Editar dominio | Acciones ▾

Información general | Estado del clúster | Estado de las instancias | Índices | Registros | Historial de actualizaciones

Paquetes | Conexiones de búsqueda entre clústeres

Estado del dominio: **Activo**

Versión de Elasticsearch: 7.4

Punto de enlace: <https://search-es-twitter-gzwx5mq3boksgemnw3u7evuhp4.us-east-1.es.amazonaws.com>

ARN del dominio: `arn:aws:es:us-east-1:599588339263:domain/es-twitter`

Kibana: https://search-es-twitter-gzwx5mq3boksgemnw3u7evuhp4.us-east-1.es.amazonaws.com/_plugin/kibana/

Zonas de disponibilidad: 1

Tipo de instancia (datos): m5.large.elasticsearch

Número de nodos: 1

Tipo de almacenamiento de: FRS