Uso de la Tecnología AWS Fargate para el despliegue de Sistema Web de Control de Flujo de Capital para una Asociación Ganadera

Rafael Cerino Frías, José Ángel Jesús Magaña, Alejandro Hernández Cadena, José Ney Garrido Vázquez, José Manuel Gómez Zea

Tecnológico Nacional de México Campus Villahermosa, Cd. Industrial. Departamento de Posgrado e Investigación; Carretera Villahermosa Frontera, Km. 3.5, Cd. Industrial, Villahermosa Tabasco, CP: 86010.

Resumen

En la última década, la tecnología de virtualización de servidores ha permitido resolver problemas que se presentaban durante el despliegue del servidor: mensajes de error inesperado, problemas de compatibilidad, renombrado de ruta de directorios, entre otros. En este artículo, analizamos la colaboración entre dos tecnologías de virtualización de servidores: Docker y AWS, que se unieron para ayudar a los desarrolladores a minimizar el tiempo de entrega de aplicaciones modernas. Se tomará como caso de estudio, el despliegue del proyecto ***Sistema Web de Control de Flujo de Capital para una Asociación Ganadera*** en un ambiente de despliegue configurado en AWS Fargate, concluyendo con una estimación de costos mensual.

Abstract

In the last decade, server virtualization technology has made it possible to solve problems that arose during server deployment: unexpected error messages, compatibility problems, directory path renaming, among others. In this article, we look at the collaboration between two server virtualization technologies: Docker and AWS, which came together to help developers minimize delivery time for modern applications. The deployment of the ***Capital Flow Control Web System*** project for a Livestock Association in a deployment environment configured in AWS Fargate will be taken as a case study, concluding with a monthly cost estimate.

Palabras clave: Docker, Contenedor Docker, Máquina virtual, Virtualización, Computación en la nube.
Keywords. Docker, Docker Container, Virtual Machine, Virtualization, Cloud Computing.

1. INTRODUCCIÓN

En la última década, la tecnología ha avanzado a pasos de gigante, sobre todo en el área de virtualización de servidores, ahora se ven desde lejos los días en que se tenían problemas para hacer el despliegue del sistema en el servidor y encontrarse con la sorpresa de mensajes de error por todos lados, por problemas de compatibilidad con el servidor, la ruta de los directorios cambiados, entre otros. Existen una gran variedad de tecnologías de virtualización de servidores; Amazon EC2, Microsoft Azure, Google Cloud, VMware, RKT, OpenVz, VirtualBox, Linux Containers – LXC, LXD de Canonica y Docker por mencionar algunos.

El crecimiento en el uso de estas tecnologías es tan grande que la mayoría de las empresas con sistemas ya hacen uso de alguna de ellas; el motivo es que proporcionan entornos de usuario escalables y seguros. Se pueden clasificar en dos grandes grupos: virtualización basada en contenedores y virtualización basada en hipervisor. Estos dos grupos, se pueden subdividir en Cloud Computing: es una colección de servicios de computación en la nube pública que en conjunto forman una plataforma de computación en la nube [1] y Servidores locales: es básicamente una computadora, parecida en muchos aspectos a la computadora que tienes en tu casa u oficina, aunque claro, tiene sus diferencias y está destinada a ser utilizada con otros fines que las típicas computadoras de casa o del trabajo [2].

En este artículo, analizaremos la colaboración entre dos tecnologías de virtualización de servidores: Docker y AWS, que se unieron para ayudar a los desarrolladores a minimizar el tiempo de entrega de aplicaciones modernas. Se tomará como caso de estudio, el despliegue del proyecto ***Sistema Web de Control de Flujo de Capital para una Asociación Ganadera***.

2. AWS Y DOCKER

Amazon Web Service (AWS) es una de las plataformas más adoptadas y completas del mundo, ofrece más de 180 servicios de centros de datos a nivel mundial. Tiene millones de clientes, incluso empresas como: Pinterest, Capital One y Nextflix, las organizaciones de gobierno están usando AWS para reducir los costos, aumentar su agilidad e innovar de forma rápida. Cuenta con una cantidad de servicios y de características incluidas que supera la de cualquier otro proveedor de la nube, ofreciendo desde tecnologías de infraestructura como: cómputo, almacenamiento y bases de datos, hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas [1].

Docker es una plataforma open source para desarrollar, enviar y ejecutar aplicaciones. Permite separar las aplicaciones de su infraestructura, para que pueda entregar software rápidamente. Con Docker, se puede administrar la infraestructura de la misma manera que administra sus aplicaciones. Al aprovechar las metodologías de Docker para enviar, probar e implementar código rápidamente, se puede reducir significativamente el retraso entre la escritura del código y su ejecución en producción [3].

La ejecución de Docker en AWS, ofrece a desarrolladores y administradores una manera muy confiable y económica de crear, enviar y ejecutar aplicaciones distribuidas en cualquier escala. Docker colabora con AWS, para ayudar a los desarrolladores a acelerar la entrega de aplicaciones modernas a la nube; así mismo, esta combinación estratégica los ayuda a utilizar Docker Compose y Docker Desktop para aprovechar el mismo flujo de trabajo local que utilizan hoy, a fin de implementar sin inconvenientes aplicaciones en Amazon ECS y en AWS Fargate.

3. AMAZON WEB SERVICES FARGATE

Amazon Web Services Fargate (AWS Fargate) es un motor informático sin servidor de pago por uso, que permite centrarse en la creación de aplicaciones sin tener que administrar los servidores [4]. AWS Fargate tiene las siguientes características:

- Permite implementar y administrar aplicaciones, no infraestructura. Fargate elimina los gastos generales operativos del escalado, la instalación de parches, la seguridad y la administración de los servidores.
- Permite monitorear las aplicaciones mediante integraciones incorporadas a los servicios de AWS, como Amazon CloudWatch Container Insights.
- Mejora la seguridad mediante el aislamiento de las cargas de trabajo desde el diseño.
- Solo se paga por lo que utiliza, Fargate escala los recursos para que coincida estrechamente con los requisitos de los recursos especificados. Con Fargate, no habrá un aprovisionamiento excesivo ni tendrá que pagar por servidores adicionales.

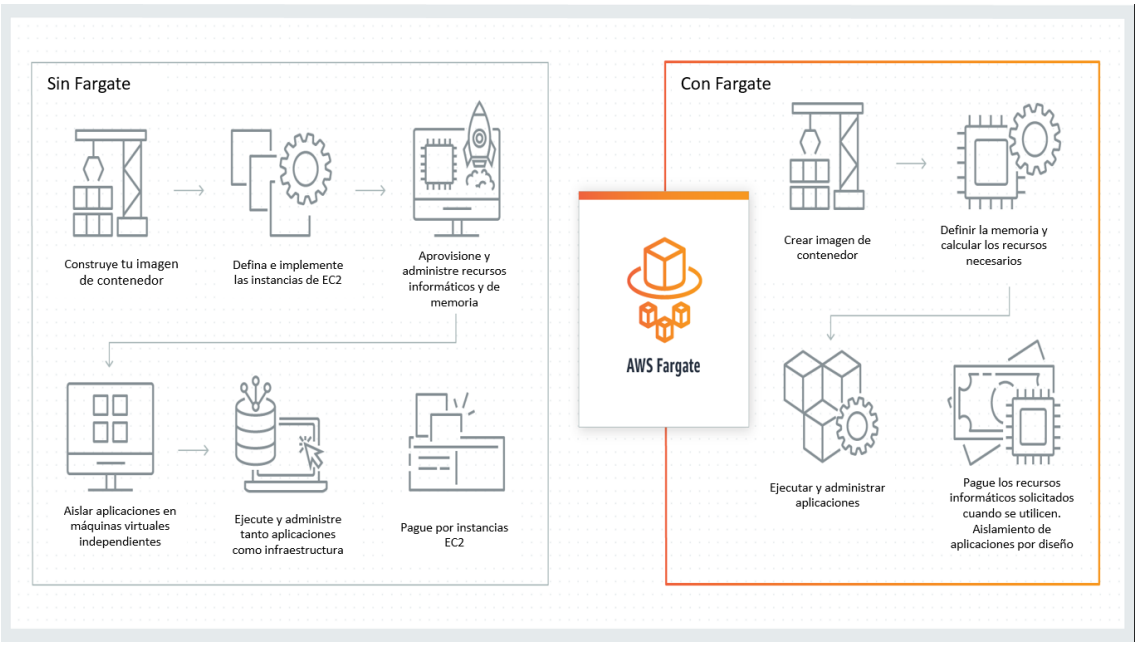

Figura 1. Comparación entre usar y no usar AWS Fargate

En la Figura 1, se explica que usar AWS Fargate permite deslindarse del rol de administrador de la infraestructura de la aplicación y en cuatro pasos puede desplegar los sistemas de forma fácil y sencilla.

4. PROPUESTA

Se propone implementar AWS Fargate para el despliegue de proyecto ***Sistema Web de Control de Flujo de Capital para una Asociación Ganadera,*** dando como justificación que la empresa no cuenta con un Servidor Local, ni con un Área de Informática que esté a cargo de dar mantenimiento y de administrar los sistemas. Ganando beneficios como: seguridad garantizada, se paga por los recursos utilizados, funcionamiento 24h/365 días, fácil de escalar y se olvida del mantenimiento de la infraestructura.

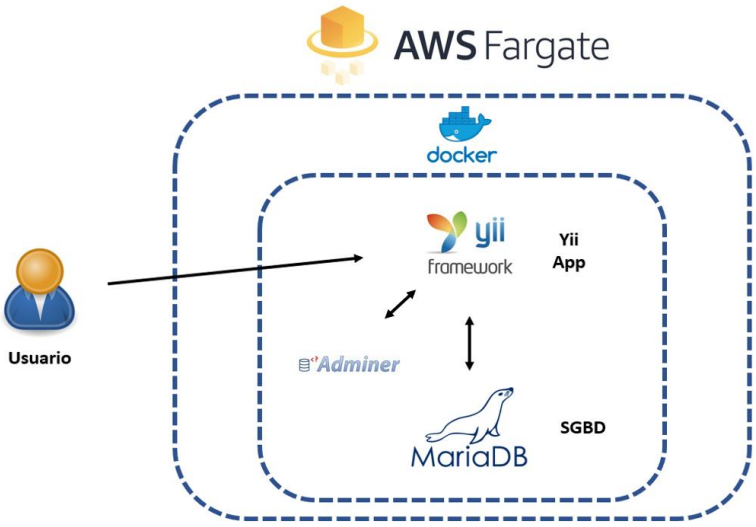

Figura 2. Arquitectura del proyecto

En la Figura 2, se muestra la arquitectura que tiene el proyecto; en la capa superior tenemos a AWS Fargate como plataforma y brindando las herramientas para el despliegue; en la capa media interviene Docker como plataforma que administra los contenedores; por último, los contenedores de Yii App, MariaDB y Adminer.

- **Yii App:** El contenedor está conformado por el Yii2 Framework y el servidor web NGINX.
- **MariaDB:** El contenedor está conformado por el Sistema Gestor de Base de Datos (SGBD) MariaDB.
- **Adminer:** El contenedor está conformado por Adminer la herramienta para administrar contenido de bases de datos de manera gráfica.

5. IMPLEMENTACIÓN

Teniendo en cuenta el problema y la solución propuesta, se hará la configuración del servidor.

a. Configuración de la red

Para configurar la red en AWS, se requiere configurar una Red Virtual Privada en la nube (VPC), uniendo en una sola red todos los recursos que necesites.

Figura 3. Configuración de VPC

Como siguiente paso, se configuran los Grupos de Seguridad que permiten decidir quién tiene acceso al grupo de recursos configurando reglas de entrada y de salida.

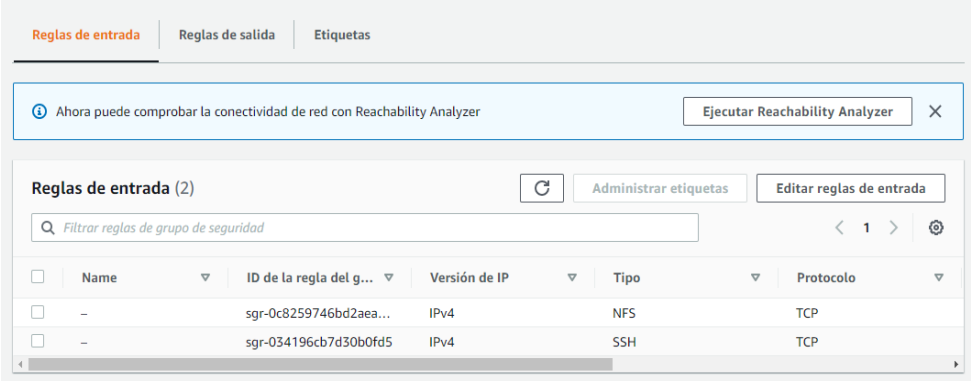

Figura 4. Configuración de las reglas de entrada

b. Configuración del volumen

Los volúmenes son el mecanismo preferido para conservar los datos generados y utilizados por los contenedores de Docker [5]. Pero para AWS Fargate se utiliza un tipo especial, el Elastic File System que es un servicio de almacenamiento en la nube proporcionado por AWS.

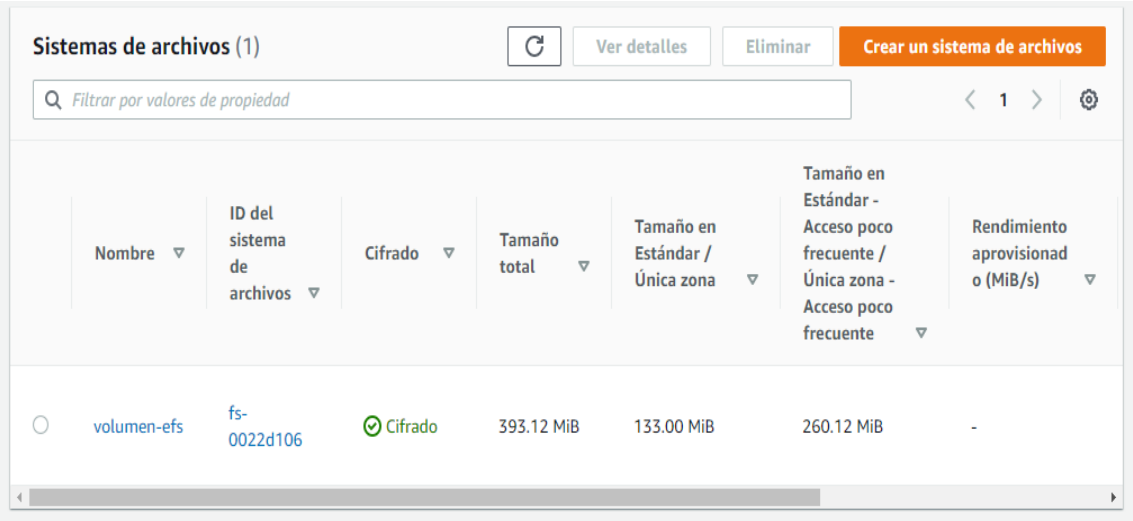

Figura 5. Elastic File System

En el volumen creado se almacenan todos los archivos necesarios para el correcto funcionamiento del sistema; el contenedor Yii App y el contenedor de base de datos MariaDB.

c. Configuración del clúster

Un clúster es una agrupación lógica de tareas o servicios [6], es necesario configurar uno en el cual se almacenará nuestro proyecto, para este caso se elige uno de Solo Redes que creará: un grupo, un VPC y dos subredes.

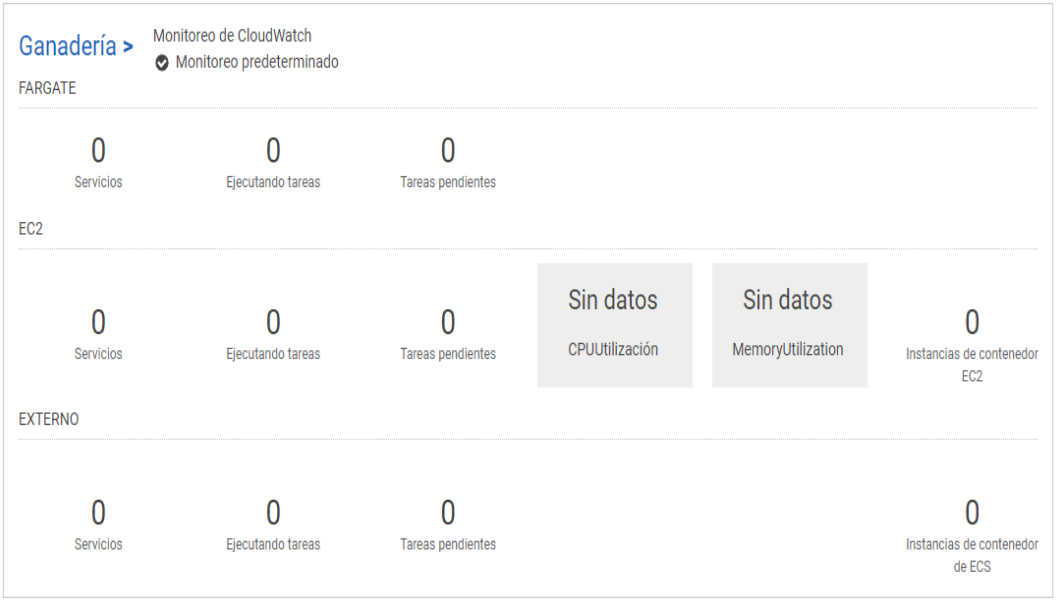

Figura 6. Ejemplo de un clúster configurado

d. Definición de tareas

Esta es la parte más crucial, se configura nuestro servicio web definiendo tareas; son el equivalente al archivo *Docker-composer.yam* que permite desplegar el servicio web con un solo comando, pero la tarea se define en un formato JSON, donde se configuraron los tres contenedores y la interacción que existe entre ellos.

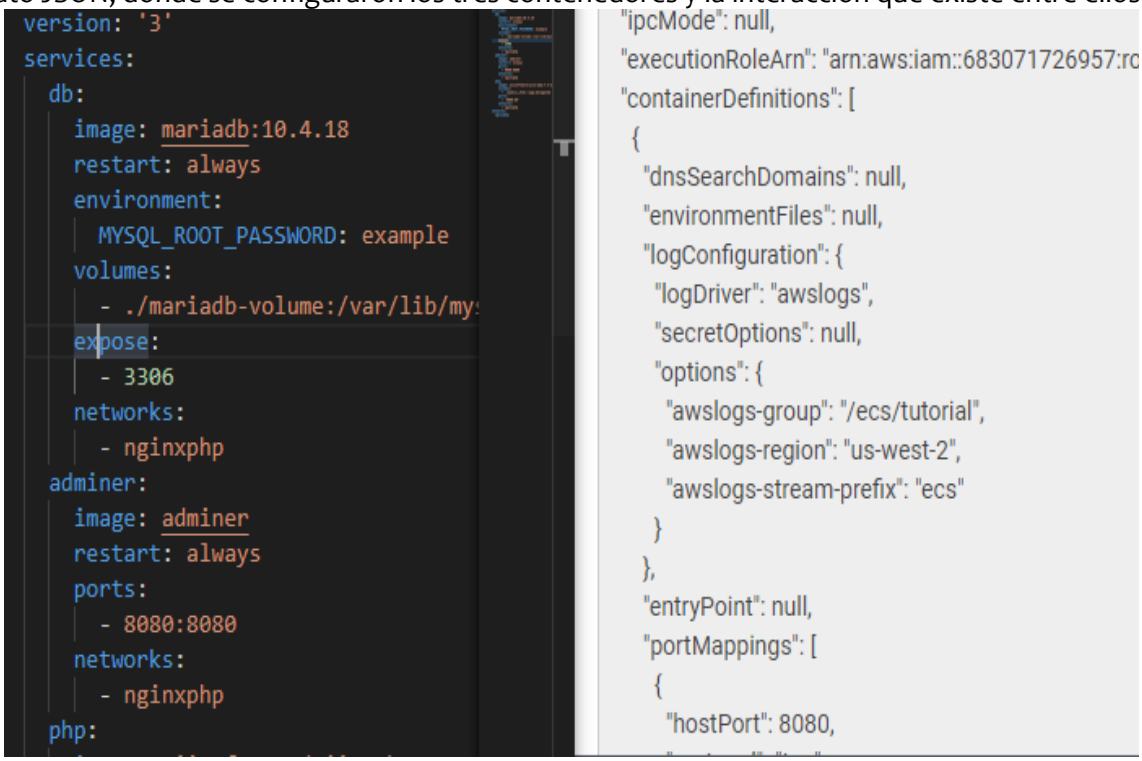

Figura 7. Fragmento de configuración Docker | AWS Fargate

En la figura anterior, se puede apreciar en la parte izquierda la configuración del archivo Docker que se utiliza para el despliegue del servicio y en la parte derecha el archivo JSON que se utiliza para configurar el servicio en AWS Fargate.

e. Creación del servicio

Para la creación del servicio se utiliza la tarea que definimos con anterioridad y se configuran las características que deseamos para nuestro servicio, mencionaremos las más importantes:

- **Tipo de lanzamiento:** Es el tipo de plan que se usa para medir los costos de los servicios AWS, seleccionamos Fargate.
- **Seleccionar el clúster:** En este caso se selecciona el clúster que se creó anteriormente: *Ganaderia*.
- **Nombre del servicio:** La etiqueta con la que se identifica el servicio.
- **VPC de clúster:** Seleccionamos el que se creó al principio.
- **Servicio de auto escalado:** Este ajusta automáticamente la capacidad del servicio de aumentar los recursos o disminuirlos según la demanda y el tráfico de red.

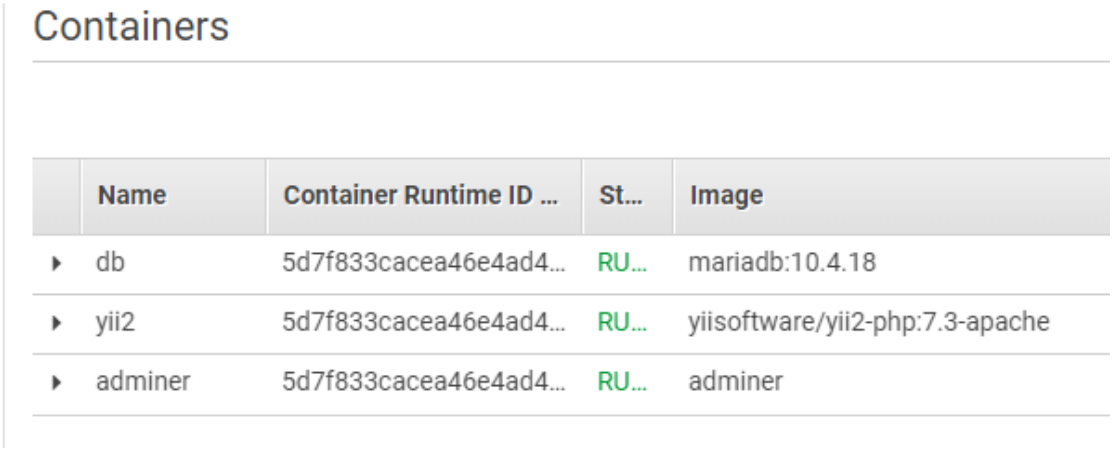

Figura 8. Tabla de contenedores ubicados en el clúster

En la Figura 8, se observa la tabla de contenedores que muestra el estatus de los servicios de la aplicación desplegada.

6. RESULTADOS

Al hacer el despliegue del proyecto ***Sistema Web de Control de Flujo de Capital para una Asociación Ganadera*** utilizando los recursos de AWS Fargate, se puede ver un gran desempeño por parte del servicio contratado.

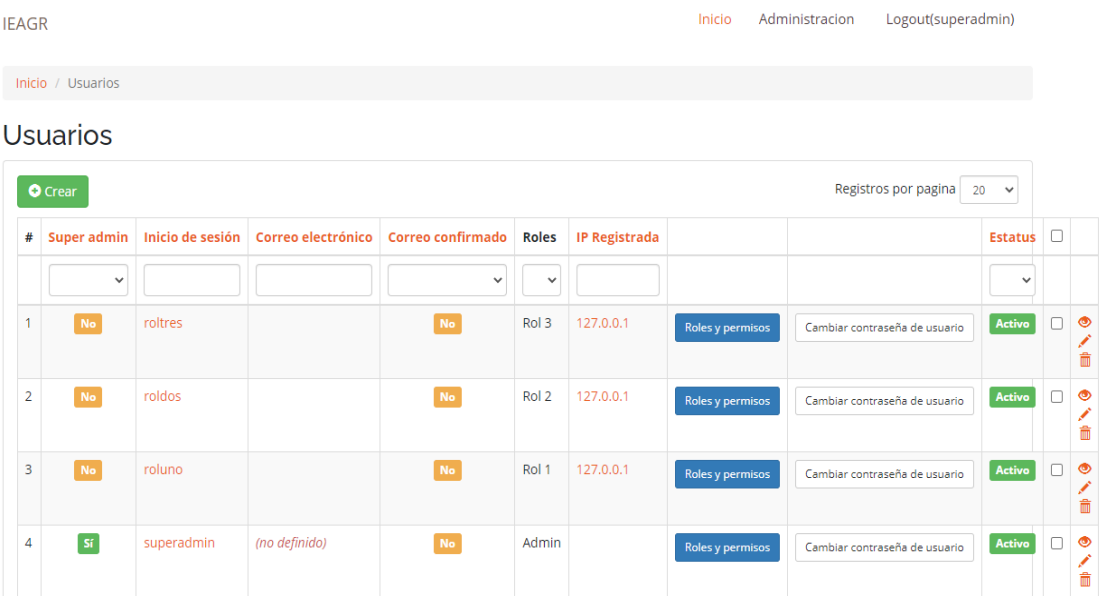

Figura 9. Pantalla del servicio desplegado con AWS Fargate

El cliente ahora puede administrar con facilidad la aplicación, sin tener que preocuparse del mantenimiento del servidor, con la seguridad garantizada y pagando un precio muy accesible por mantenerla activa.

Se estima que para una Asociación Ganadera de entre 100 y 150 socios, con un flujo de operaciones de entre 1200 y 1500 consultas mensuales se puede atender un promedio de hasta 8 socios por hora. AWS Fargate tiene dos tipos de tarifas para los CPU y para los GB de tráfico que tenga el servicio; con precios en dólares con una tasa de cambio al día de hoy de 20.89 pesos por dólar, nos da un total de aproximadamente entre 100 y 120 pesos mexicanos mensuales.

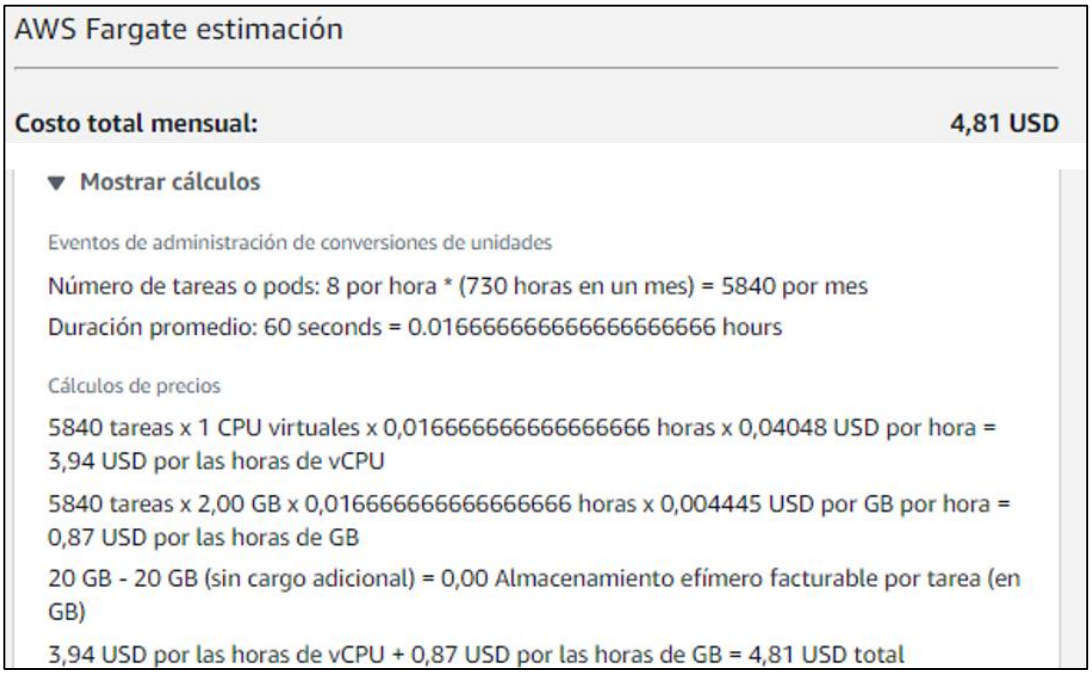

Figura 10. Estimación de costo total mensual de AWS Fargate

7. CONCLUSIÓN

La colaboración entre Docker y AWS proporciona un entorno de desarrollo para un despliegue barato y sencillo, permitiendo que los despliegues posteriores sean más rápidos dado que la configuración queda guardada en la base de datos de AWS.

En trabajos futuros, se pretende configurar con Docker un ambiente para el correcto funcionamiento de un proyecto utilizando una NAS en combinación.

REFERENCIAS

[1] Amazon Web Services, «Informática en la nube con AWS,» 2014. [En línea]. Available: https://aws.amazon.com/es/what-is-aws/. [Último acceso: 11 Diciembre 2021].

[2] Infranetworking, «Servidor local: ¿Qué es y cómo puedo montar uno?,» 2020. [En línea]. Available: https://blog.infranetworking.com/servidor-local/. [Último acceso: 12 Diciembre 2021].

[3] Docker, «Docker es el futuro.,» diciembre 2021. [En línea]. Available: https://docs.docker.com/get-docker/. [Último acceso: 11 Diciembre 2021].

[4] Amazon Web Services, «AWS Fargate, » 2021. [En línea]. [Último acceso: 12 Diciembre 2021].

[5] Docker, «Usar volumenes,» 2021. [En línea]. Available: https://docs.docker.com/storage/volumes/. [Último acceso: 12 Diciembre 2021].

[6] Amazon Web Servicies, «Clústeres de Amazon ECS,» 2021. [En línea]. Available: https://docs.aws.amazon.com/es_es/AmazonECS/latest/developerguide/clusters.html. [Último acceso: 12 Diciembre 2021].

Correo autor: ***m20301149@villahermosa.tecnm.mx***