

# Taller de Modularización con Virtualización e Introducción a Docker y a AWS

## Sebastián Villamarín Rodríguez

### AREP- LAB 5

#### Descripción

En el Taller de Modularización con Virtualización e Introducción a Docker y a AWS se realizó primero el aprendizaje del manejo de contenedores en Docker, el cual se desplegó la primera instancia en Docker. Después, se realizó el respectivo aprendizaje de AWS para poder con el contenedor creado en Docker, desplegarlo en AWS en la máquina virtual, para así tener base para poder iniciar la tarea la cual consiste en una implementación de una arquitectura la cual consiste en un balanceador de carga en donde se usa el método de RoundRobin. Este balanceador de carga se encarga de realizar las respectivas peticiones a los LogService, los cuales se encargan de realizar la conexión con las bases de datos montada en MongoDB, que almacena todos los mensajes entrantes. En la arquitectura del programa se encuentran en total cinco contenedores, uno para el RoundRobin, tres para el LogService y uno para las bases de datos en Mongo.

#### Prerrequisitos

Para la realización y ejecución tanto del programa como de las pruebas de este, se requieren ser instalados los siguientes programas:

- Maven. Herramienta que se encarga de estandarizar la estructura física de los proyectos de software, maneja dependencias (librerías) automáticamente desde repositorios y administra el flujo de vida de construcción de un software.
- GIT. Sistema de control de versiones que almacena cambios sobre un archivo o un conjunto de archivos, permite recuperar versiones previas de esos archivos y permite otras cosas como el manejo de ramas (branches).
- Docker. Programa encargado de crear contenedores ligeros y portables para las aplicaciones software que puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo, facilitando así también los despliegues.

Para asegurar que el usuario cumple con todos los prerrequisitos para poder ejecutar el programa, es necesario disponer de un Shell o Símbolo del Sistema para ejecutar los siguientes comandos para comprobar que todos los programas están instalados correctamente, para así compilar y ejecutar tanto las pruebas como el programa correctamente.

- mvn -version
- git --version
- java -version
- docker version

### Instalación

Para descargar el proyecto de GitHub, primero debemos clonar este repositorio, ejecutando la siguiente línea de comando en GIT.

```
https://github.com/svillamarin27/AREP-LAB5.git
```

### Ejecución

Para compilar el proyecto utilizando la herramienta Maven, nos dirigimos al directorio donde se encuentra alojado el proyecto, y dentro de este ejecutamos en un Shell o Símbolo del Sistema el siguiente comando:

```
mvn package
```

### Localhost

Para probar ahora el correcto funcionamiento del Docker de manera local o localhost del programa RoundRobin, primero ejecutamos los siguientes comandos en orden.

```
docker build --tag firstsparkjavarepo/roundrobin .
docker images
docker run -d -p 35000:6000 --name firstcontainerweb firstsparkjavarepo/roundrobin
```

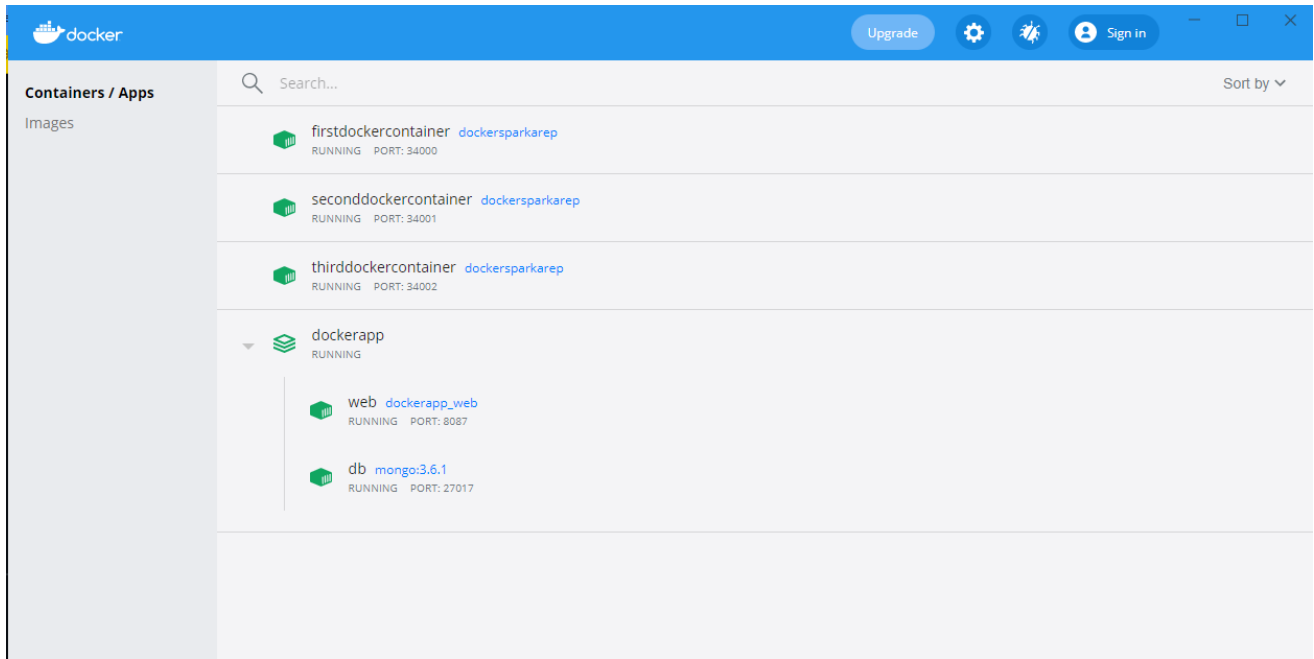
Luego de ejecutarlos en exactamente ese mismo orden, tenemos el siguiente resultado en pantalla.

```
[+] Building 8.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.7s
=> => transferring dockerfile: 32B                                              0.0s
=> [internal] load .dockerignore                                                1.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/openjdk:8                    1.3s
=> [auth] library/openjdk:pull token for registry-1.docker.io                 0.0s
=> [internal] load build context                                              0.6s
=> => transferring context: 5.68kB                                             0.0s
=> [1/4] FROM docker.io/library/openjdk:8@sha256:fc73b8ac2832f27d852a380b3d5fa6e051c4f38d27da6b81bc609f126feff84  0.0s
=> CACHED [2/4] WORKDIR /usrapp/bin                                           0.0s
=> [3/4] COPY /target/classes /usrapp/bin/classes                             0.9s
=> [4/4] COPY /target/dependency /usrapp/bin/dependency                       1.4s
=> exporting to image                                                         2.1s
=> => exporting layers                                                         1.3s
=> => writing image sha256:5eb0eb967a33770b76cbd9da316a0303774a5f8bc5e26bf3a5689b64bc91dfe5  0.1s
=> => naming to docker.io/arep-lab5/logservice                               0.1s
```

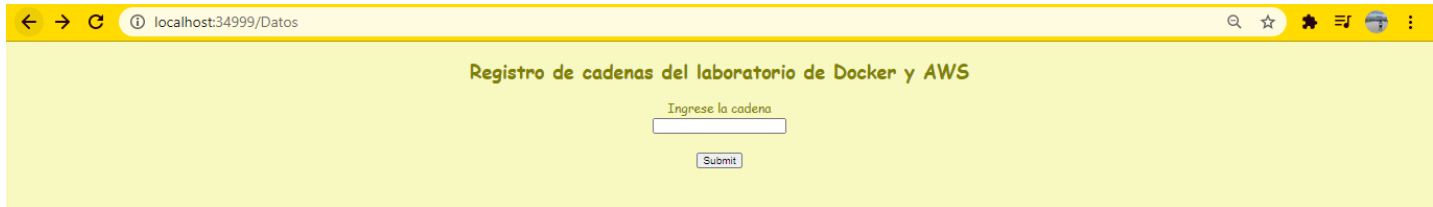
Ahora, para correr los tres logs en puertos diferentes, se ejecutan los siguientes comandos en orden.

```
docker run -d -p 34000:6000 --name firstdockercontainer firstsparkjavarepo/logservice
docker run -d -p 34001:6000 --name seconddockercontainer firstsparkjavarepo/logservice
docker run -d -p 34002:6000 --name thirddockercontainer firstsparkjavarepo/logservice
```

Ahora, para verificar que en la aplicación Docker se hayan desplegado con éxito los contenedores LogService y RoundRobin en sus respectivos puertos, se abre la aplicación de Docker de escritorio y se hace la verificación que todos los contenedores estén corriendo en sus respectivos puertos. Como se ve en la siguiente imagen, todos los contenedores están corriendo satisfactoriamente.

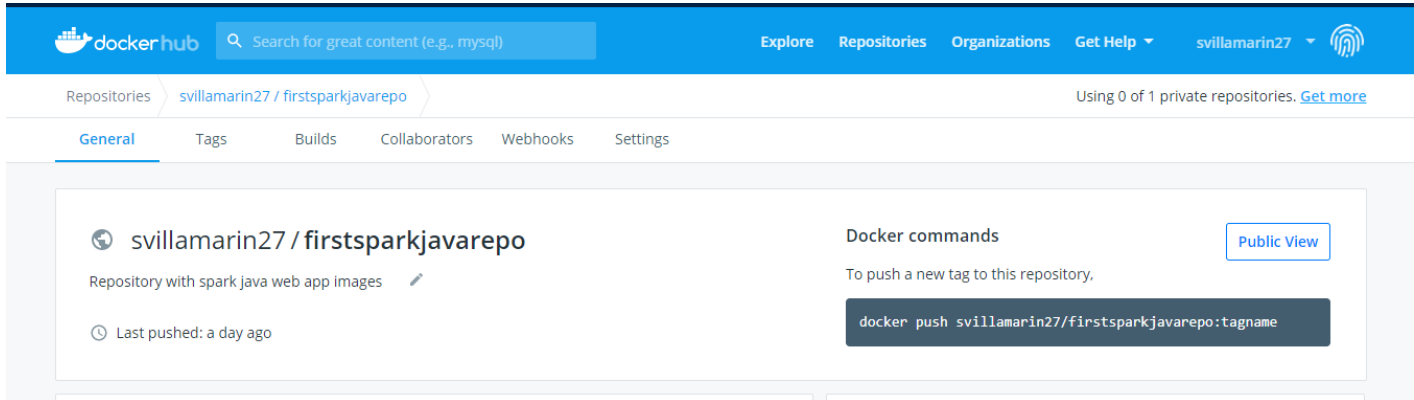


Para comprobar que la página web ha sido desplegada con éxito, se ingresa en el navegador la siguiente URL: localhost:34999/Datos . Luego de ingresar la URL en el navegador, se obtiene el siguiente resultado.



## AWS

Antes de iniciar a utilizar AWS, primero se debe subir cada uno de los contenedores creados a un repositorio. Para realizar esto, primero se creó el primer repositorio en Docker Hub llamado firstsparkjavarepo, como se ve a continuación.



Luego, se ejecutaron los siguientes comandos en orden para poder subir los contenedores.

```
docker tag firstsparkjavacontainer/roundrobin svillamarin27/firstsparkjavacontainer
docker push svillamarin27/firstsparkjavacontainer:latest
```

Para iniciar a desplegar el contenedor en una máquina virtual alojada en AWS, primero se selecciona el tipo de máquina virtual que se utilizará, en este caso, se utilizará Amazon Linux 2 AMI (HVM), SSD Volume Type. Para utilizarla, se realiza clic en el botón Seleccionar.

Paso 1: Elegir una imagen de Amazon Machine (AMI)

Cancelar y salir

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS, nuestra comunidad de usuarios o AWS Marketplace, o puede seleccionar una de sus propias AMI.

Para buscar una AMI, escriba un término de búsqueda; por ejemplo, "Windows" ✕

Buscar por parámetro de Systems Manager

Inicio rápido

Mis AMI

AWS Marketplace

AMI de la comunidad

☐ Solo capa gratuita ?

Amazon Linux

Apto para la capa

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-038f1ca1bd58a5790 (64 bits x86) / ami-01437b695ea9225cd (64 bits Arm)

Amazon Linux 2 incluye cinco años de soporte. Proporciona el kernel de Linux 4.14 adaptado para un rendimiento óptimo en Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1 y en los últimos paquetes de software a través de complementos.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

macOS Big Sur 11.2.1 - ami-0af92ed5f8a74cda3

The macOS Big Sur AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of multiple AWS packages included in the AMI.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

Seleccionar

64 bits (x86)

64 bits (Arm)

Seleccionar

64 bits (Mac)

Ahora se selecciona el tipo de instancia. Para esta máquina virtual, se selecciona t2.micro, la cual es apta para la capa gratuita. luego de seleccionarla, se realiza clic en Revisar y lanzar.

Paso 2: Página Choose an Instance Type

Amazon EC2 proporciona una amplia selección de tipos de instancias optimizados para adaptarse a diferentes casos de uso. Las instancias son servidores virtuales que pueden ejecutar aplicaciones. Tienen distintas combinaciones de CPU, memoria, almacenamiento y capacidad de red, lo que proporciona una gran flexibilidad para elegir la combinación de recursos adecuada para las aplicaciones. [Más información](#) acerca de los tipos de instancias y cómo pueden satisfacer sus necesidades de computación.

Filtrar por: 

Todas las familias de instancias

Generación actual

Mostrar/ocultar columnas

Seleccionada actualmente: t2.micro (- ECU, 1 vCPU, 2.5 GHz, -, 1 GiB memoria, EBS solo)

	Familia	Tipo	vCPU	Memoria ( GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red	Compatibilidad con IPv6
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS solo	-	De bajo a moderado	Sí
<input checked="" type="checkbox"/>	t2	t2.micro Apto para la capa gratuita	1	1	EBS solo	-	De bajo a moderado	Sí
<input type="checkbox"/>	t2	t2.small	1	2	EBS solo	-	De bajo a moderado	Sí
<input type="checkbox"/>	t2	t2.medium	2	4	EBS solo	-	De bajo a moderado	Sí
<input type="checkbox"/>	t2	t2.large	2	8	EBS solo	-	De bajo a moderado	Sí

Cancelar

Anterior

Revisar y lanzar

Siguiente: Página Configuración de los detalles de la instancia

A continuación se muestra la instancia para verificar la máquina virtual que está a punto de ser lanzada. Para lanzarla, se realiza clic en el botón Lanzar.

Paso 7: Página Review Instance Launch

la capa en los últimos paquetes de software a través de complementos.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm

Tipo de instancia

Editar tipo de instancia

Tipo de instancia	ECU	vCPU	Memoria ( GiB)	Almacenamiento de la instancia (GB)	Optimizado para EBS disponible	Desempeño de la red
t2.micro	-	1	1	EBS solo	-	Low to Moderate

Grupos de seguridad

Editar grupos de seguridad

Nombre del grupo de seguridad

launch-wizard-3

Descripción

launch-wizard-3 created 2021-03-12T18:49:22.969-05:00

Tipo	Protocolo	Rango de puertos	Origen	Descripción
SSH	TCP	22	0.0.0.0/0	

Detalles de la instancia

Editar detalles de la instancia

Cancelar

Anterior

Lanzar

Luego, se procede a crear un nuevo par de llaves para poder acceder a la máquina virtual desde el computador en cuestión. Para esto se selecciona la opción Crear un nuevo par de llaves y se escribe el nombre del par de claves. Para descargar la llave, se realiza clic en el botón Descargar par de llaves.

×

Seleccione un par de claves existente o cree un nuevo par de claves

Un par de claves consta de una **clave pública** que AWS almacena y un **archivo de claves privadas** que usted almacena. Juntos, le permiten conectarse a su instancia de forma segura. Para las AMI de Windows, el archivo de claves privadas es necesario para obtener la contraseña usada para iniciar sesión en la instancia. Para las AMI de Linux, el archivo de claves privadas le permite realizar una conexión SSH segura con su instancia.

Nota: El par de claves seleccionado se añadirá al conjunto de claves autorizadas para esta instancia. Obtenga más información sobre [cómo eliminar pares de claves existentes de una AMI pública](#).

Crear un nuevo par de claves

Nombre del par de claves

awsthird

Descargar par de claves

...

Tiene que descargar el **archivo de claves privadas** (archivo \*.pem) para poder continuar. **Guárdelo en un lugar seguro y accesible**. No podrá descargar el archivo de nuevo después de crearlo.

Después de descargar el par de llaves, ahora se procede a realizar clic en el botón Lanzar instancias.

Un par de claves consta de una **clave pública** que AWS almacena y un **archivo de claves privadas** que usted almacena. Juntos, le permiten conectarse a su instancia de forma segura. Para las AMI de Windows, el archivo de claves privadas es necesario para obtener la contraseña usada para iniciar sesión en la instancia. Para las AMI de Linux, el archivo de claves privadas le permite realizar una conexión SSH segura con su instancia.

Nota: El par de claves seleccionado se añadirá al conjunto de claves autorizadas para esta instancia. Obtenga más información sobre [cómo eliminar pares de claves existentes de una AMI pública](#).

Crear un nuevo par de claves

Nombre del par de claves

awsthird

Descargar par de claves

...

Tiene que descargar el **archivo de claves privadas** (archivo \*.pem) para poder continuar. **Guárdelo en un lugar seguro y accesible**. No podrá descargar el archivo de nuevo después de crearlo.

Cancelar

Lanzar instancias

Ahora, se muestra que la instancia ha sido lanzada con éxito. Para verificar que esta ha sido lanzada, se realiza clic en el botón Ver instancias.

## Página Launch Status

### Cómo conectarse a la instancia

Se está lanzando su instancia. Pueden transcurrir unos minutos hasta que tenga el estado **en ejecución**, momento en el cual estará lista para poder usarla. Las horas de uso de la nueva instancia comenzarán inmediatamente y seguirán devengando gastos hasta que detenga o termine la instancia.

Haga clic en **Ver las instancias** para monitorizar el estado de su instancia. Cuando la instancia tenga el estado **en ejecución**, podrá **conectarse** a ella desde la pantalla Instancias. [Más información](#) cómo conectarse a la instancia.

#### ▼ Aquí tiene algunos recursos útiles que le ayudarán a comenzar

- [Cómo conectarse a la instancia Linux](#)
- [Amazon EC2: Guía del usuario](#)
- [Más información sobre la capa de uso gratuita de AWS](#)
- [Amazon EC2: Foro de debate](#)

Mientras se están lanzando sus instancias, también puede

- [Crear alarmas de comprobación de estado](#) recibir notificaciones cuando estas instancias no superen las comprobaciones de estado. (Podrían aplicarse cargos adicionales)
- [Crear y asociar volúmenes de EBS adicionales](#) (Podrían aplicarse cargos adicionales)
- [Administrar grupos de seguridad](#)

Ver instancias

Para conectarse a la instancia, se realiza clic en el botón Acciones, para posteriormente realizar clic en el botón Conectar.

Servicios

Buscar servicios, características, productos del Marketplace

vocstartsoft/user1326867=sebastian.villamarin@mail.escuelaing... Norte de Virginia Soporte

New EC2 Experience

Panel de EC2

Eventos

Etiquetas

Limites

Instancias

Instancias

Tipos de instancia

Plantillas de lanzamiento

Solicitudes de spot

Savings Plans

Instancias reservadas

Hosts dedicados

Instancias programadas

Reservas de capacidad

Imágenes

AMI

Le damos la bienvenida a la nueva experiencia de instancias

Rediseñamos la consola de EC2 para que sea más fácil de usar. Para cambiar entre la consola antigua y la nueva, use el conmutador de la Nueva Experiencia de EC2 que se encuentra arriba del panel de navegación. Lanzaremos actualizaciones continuamente basadas en los comentarios de los clientes.

Instancias (1/3)

Información

Conectar

Estado de la instancia

Acciones

Lanzar instancias

Conectar

Ver detalles

Administrar el estado de la instancia

Configuración de la instancia

Redes

Seguridad

Imagen y plantillas

	Name	ID de la instancia	Estado de la i...	Tipo de inst...	Com
<input type="checkbox"/>	-	i-027afaef5368f5a71	En ejecución	t2.micro	2
<input type="checkbox"/>	-	i-0d935cccd9de41250	En ejecución	t2.micro	2
<input checked="" type="checkbox"/>	-	i-0f481d9e4efac2324	En ejecución	t2.micro	1

Instancia: i-0f481d9e4efac2324

Detalles

Seguridad

Redes

Almacenamiento

Comprobaciones de estado

Monitoreo

Etiquetas

Resumen de instancia

Información

ID de la instancia

Dirección IPv4 pública

Direcciones IPv4 privadas

Para realizar la respectiva conexión con la instancia, se realiza clic en el botón Cliente SSH, que es el medio en el cual se realizará la conexión con la instancia.

Conectarse a la instancia

Información

Conéctese a la instancia i-0f481d9e4efac2324 mediante cualquiera de estas opciones

Conexión de la instancia EC2

Administrador de sesiones

Cliente SSH

ID de la instancia

i-0f481d9e4efac2324

1. Abra un cliente SSH.

2. Localice el archivo de clave privada. La clave utilizada para lanzar esta instancia es awstthird.pem

3. Ejecute este comando, si es necesario, para garantizar que la clave no se pueda ver públicamente.

chmod 400 awstthird.pem

4. Conéctese a la instancia mediante su DNS público:

ec2-54-163-34-86.compute-1.amazonaws.com

Ejemplo:

ssh -i "awstthird.pem" ec2-user@ec2-54-163-34-86.compute-1.amazonaws.com

Ahora, se ejecuta el SSH desde el computador con el cual se desea realizar la conexión con la instancia, y se ejecuta el siguiente comando. ssh -i "awstthird.pem" ec2-user@ec2-54-163-34-86.compute-1.amazonaws.com  
Como se puede observar, el contenedor ha sido desplegado satisfactoriamente desde la máquina virtual montada en AWS.

ec2-54-174-237-126.compute-1.amazonaws.com:35000/Datos

Registro de cadenas del laboratorio de Docker y AWS

Ingrese la cadena

Submit

# Autor Sebastián Villamarín Rodríguez

# Licencia

Este proyecto esta licenciado bajo la licencia General Public License v3.0, revise el archivo LICENSE para más información.