# Orquestación de Contenedores

Amazon Web Services

# Introducción

En esta práctica vais a desplegar un servicio desde la consola de AWS. Esto es un pequeño tutorial de cómo orquestar un servicio de prueba.

Recordemos los pasos que hay que hacer:

1. Crear un Cluster para el servicio
2. Crear un tipo de tarea para el servicio
3. Crear un balanceador de carga (para saber a qué máquina apuntar)
4. Crear el servicio

Ahora vamos a ir paso por paso viendo cómo podemos hacerlo. Os animo a que miréis que significa cada opción. Aquí no vamos a estudiar en profundidad que significa cada cosa, tiraremos mucho del por defecto. Pero para los siguientes pasos deberías entender que haceis.

# 1. Crear un Cluster

Recordemos que un cluster es un conjunto de instancias/máquinas que trabajan conjuntamente con un objetivo. Vamos a empezar creando ese grupo.

1. Entramos en “Elastic Container Service”
2. Veremos un botón que dice “Create Cluster”. Clicamos en él
3. Como vamos a usar instancias Amazon EC2 instances, así que marcamos solo esa opción
4. Nos va a decir que creamos un nuevo ASG (Auto Scaling Group). Sirve para crear grupos de instancias EC2.
    a. Usaremos de tipo On-Demand. Las spot son más baratas, pero te las pueden tirar en cualquier momento y eso no estaría bien para producción
    b. Sistema operativo: Amazon Linux 2023
    c. Tipo de instancia: t3.micro (la gratuita)
    d. Instance Role: LabInstanceProfile
    e. Capacidad: Minimo 1 máximo 1 (queremos siempre tener 1)
    f. SSH Key pair: vockey
    g. Root EBS Volume: 30gb
5. Para el tema de redes
    a. La vpc por defecto
    b. Las subredes todas las que vienen
    c. Security group el que viene
    d. Auto assign public ip lo que viene
6. Le damos a create

7. Ahora el cluster se estará creando. Puede que tarde un rato, así que esperamos un poco
8. Una vez esté, clicamos en el cluster creado

Ya tenemos el cluster creado. Ahora vamos a por el siguiente paso

# 2. Crear un tipo de tarea para el servicio (task definition)

Ahora le tenemos que explicar al cluster que tarea va a ejecutar. Para eso la tenemos que definir. Recordemos que (por simplificar) tarea=contenedor. Vamos a definir esta tarea.

1. Viendo el cluster que acabáis de crear, a la izquierda tendréis una columna con diferentes opciones. Clicad en “task definitions”
2. Le dais a “Create new Task Definition”. Así definiremos la tarea
3. Le poneis nombre “test-gtio”
4. Seleccionais Amazon EC2 instances
    a. OS/Architecture: Linux/X86_64
    b. Network mode: awsvpc
    c. Task size: 0.85cpu 0.85gb (esto es cuantos recursos tendrá esta tarea, un poco menos de lo que tiene la máquina)
    d. Task role: LabRole
    e. Task Execution Role: LabRole
5. Ahora tenemos que definir el primer contenedor que se ejecutará. Este será un contenedor essential, es decir, que sin él la tarea no funcionará.
    a. Nombre: test-gtio
    b. Image URI: La imagen de vuestro contenedor. Tendrá que estar en ECR. Estudiad como se sube un contenedor a ECR, os lo dejo a vosotros ;)
    c. Essential Container: Yes
    d. Port Mappings: Esto dependerá de la imagen que uséis, pero son los puertos que tenéis que abrir. Pongamos por ejemplo el 8080, protocolo TCP, app protocol HTTP y port name por defecto
    e. Recursos: cuantos recursos le vamos a dar a ese contenedor en concreto . Recordemos que puede haber varios contenedores en una misma tarea. Así que esto tendrá que ser igual o menos que antes. En este caso 0.85 cpu y 0.85 gb
    f. No tocaremos las variables de entorno
    g. Dejaremos por defecto que los logs vayan a cloudwatch, que nos vendrá bien para el futuro.
    h. El resto podemos dejar por defecto, así que le damos a create!

Así tendremos la task definition creada. Ahora podemos pasar al siguiente paso, crear el balanceador de carga

# 3. Crear un balanceador de carga

Ya tenemos un cluster (conjunto de máquinas) y una task definition (la tarea que va a hacer ese cluster). Pero ahora necesitamos alguien que nos indique a qué máquina del cluster va a llegar nuestra petición. Por eso necesitamos un balanceador de carga. Para ello vamos a hacer lo siguiente:

1. Vamos al servicio “EC2”
2. En la columna izquierda buscamos “Load Balancers” y clicamos en él. Estará en la subsección “Load Balancing”
3. Clicamos en “Create Load Balancer” y luego en “Application Load Balancer”
4. Rellenamos la configuración así:
    a. Load Balancer Name: test-load-balancer
    b. Scheme: Internet-facing -> Queremos que se pueda acceder desde internet
    c. IP address type: IPv4
    d. VPC: Seleccionamos la de por defecto
    e. En Mappings seleccionamos todas las zonas que haya y dejamos la subnet por defecto -> Queremos que pueda acceder a las redes por defecto de todas las zonas
    f. Security groups dejamos el de por defecto, con las cuentas de invitados de AWS no podemos controlar mucho más
5. Y ahora veremos la sección de Listeners and Routing.

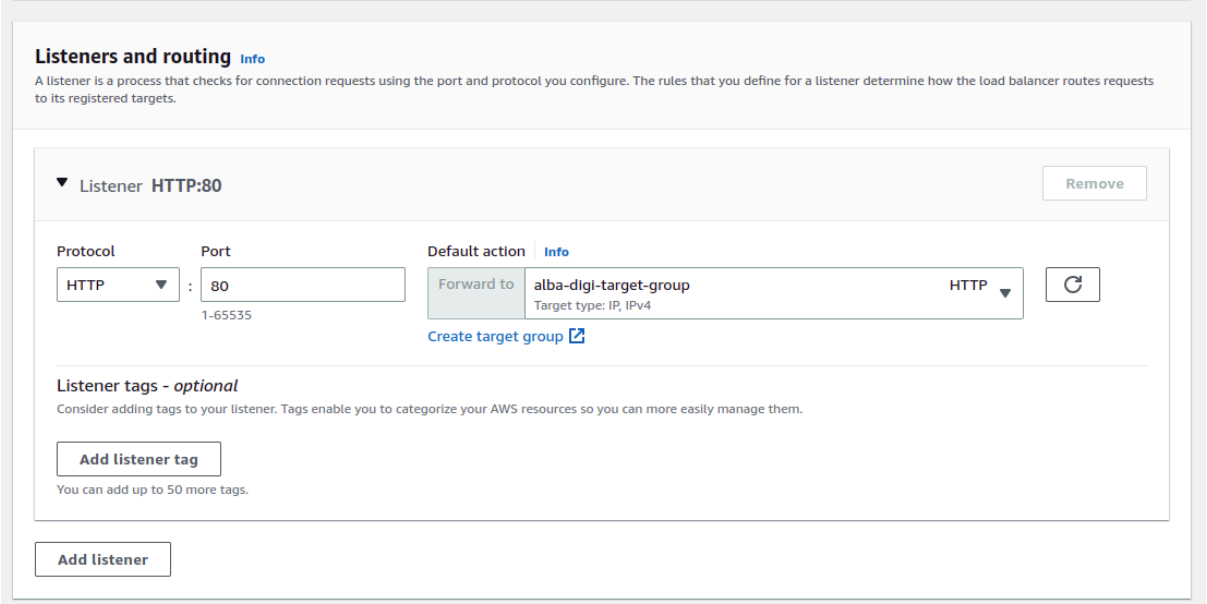
Ahora le vamos a explicar al balanceador a qué grupo de máquinas puede redirigir. Y para ello tenemos que crear ese grupo de máquinas. Será un grupo inteligente, así todo será automático. Será un “Target group”, así que vamos a clicar donde pone “Create target group”


Esto nos llevará a otra pantalla donde configuraremos el target group. Para ello rellenaremos la web con esta configuración:

1. Choose a target type: Elegimos IP Addresses
2. Target Group Name: test-target-group
3. Protocol: HTTP
4. Port: 8080 -> Más arriba hemos abierto el puerto 8080, que es donde funciona este contenedor. Cuando el balanceador apunte a este grupo de instancias, queremos que apunte en el puerto 8080. Pero si habeis elegido otro, será otro
5. Tipo de IP: IPv4
6. VPC: Seleccionamos la que hemos elegido anteriormente en al cluster
7. Protocol Version: HTTP1
8. Ahora sí que vamos a configurar el healthcheck
    a. Health Check Protocol: HTTP

    b. Health Check Path: por ejemplo, /alive Dependerá de vuestro contenedor
9. Clicamos en Next.
10. lo dejamos todo igual, no añadimos nada al balanceador y luego clicamos en Create Target Group

Con esto ya tenemos nuestro target group. Ahora podemos seguir configurando nuestro balanceador de carga. Volvemos a la pestaña anterior, clicamos en el círculo más a la izquierda y en el desplegable seleccionamos el target group que acabamos de crear. Lo dejamos tal que así. El puerto será en el que queremos escuchar. Si siempre ha sido 8080, será 8080




Y le damos a Create Load Balancer. Le damos a “view Load Balancer”. Ahora tenemos que esperar un poco hasta que en “State” cambie de “provisioning” a “Active”

Ya tenemos nuestro balanceador de carga, por lo que podemos empezar con nuestro último paso

# 4. Crear el servicio

Ahora tenemos un cluster (conjunto de máquinas), una task definition (una tarea tipo) y un balanceador de carga (quien sabe a donde apuntar). Ahora necesitamos juntar todo esto, y para ello vamos a crear un servicio. Para ello haremos lo siguiente:

1. Volveremos al cluster que hemos creado anteriormente. Para ello vamos al servicio “Elastic Container Service” y clicamos en él.
2. Abajo veremos que está seleccionada la pestaña “Services”. Clicaremos en el botón de abajo, donde pone “Create”. Así empezaremos a crear el servicio
3. El environment lo configuraremos tal que así:
    a. Opción Capacity Provider Strategy

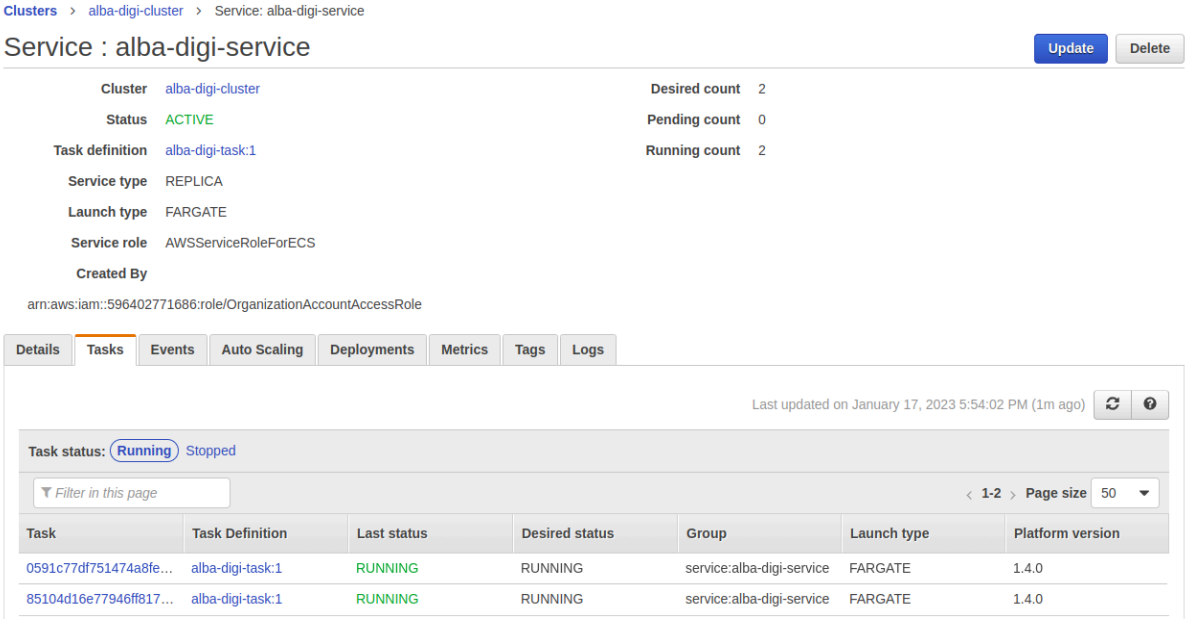
    b. Strategy: use cluster default
4. La configuración de despliegue tal que así:
    a. Tipo: service
    b. Task definition - Family: test-gtio (la creada anteriormente)
    c. Task Definition - Revision: La que ponga (latest)
    d. Service name: test-gtio
    e. Service type: Replica
    f. Desired tasks: 1 (vamos a poner más de una, que ya que creamos el servicio…)
    g. El resto lo dejamos igual
5. Service connect: Lo dejamos igual
6. Service discovery: Lo dejamos igual
7. Networking:
    a. La vpc de siempre
    b. Con todas las subnets
    c. El security group que podemos elegir
8. Load balancing. Aquí vamos a unir nuestro balanceador que hemocreado anteriormente:
    a. Seleccionamos Application Load Balancer
    b. Elegimos el contenedor que nos sale
    c. Y elegimos usar un balanceador existente
    d. Seleccionamos el elegido anteriormente
    e. Le damos un periodo de gracia de 30 segundos (espera 30 segundos a que el balanceador vea si está vivo el contenedor o no, así le dejamos que se levante tranquilo y que el endpoint de health empiece a funcionar
    f. Listener elegimos el anteriormente creado
    g. Y el target group lo mismo
9. El resto lo vamos a dejar igual (podeis curiosear) y le damos a CREATE

¡Ya tenemos nuestro servicio funcionando! Vamos a probarlo y podemos dar la práctica por terminada

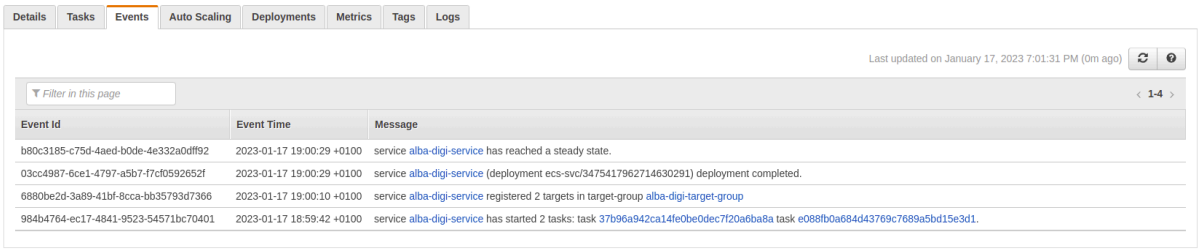
# 5. Probar el servicio

Para poder dar por buena la práctica tenéis que revisar esto

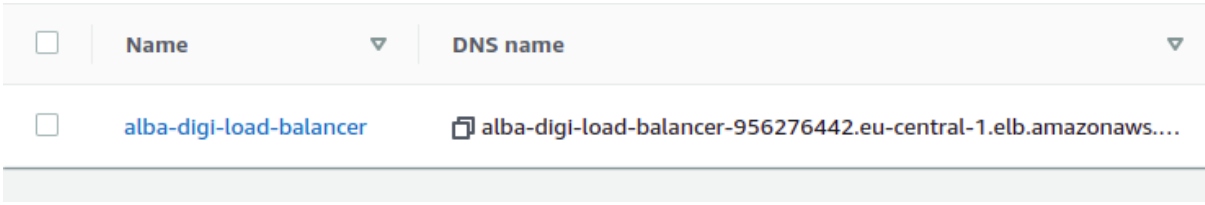
1. Servicio con sus 2 tasks en running, tal que así.



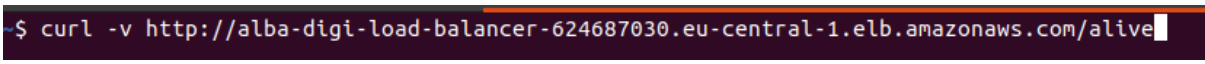

2. Revisar los eventos del servicio, donde se vea que el servicio está steady (puede que tarde un poco en aparecer)




3. Probar que el servicio está ejecutándose. Para ello vamos a hacer lo siguiente:
   a. Ir al servicio EC2
   b. Ir al apartado “Load Balancers”
   c. Copiar el dns name, lo podeis encontrar aquí

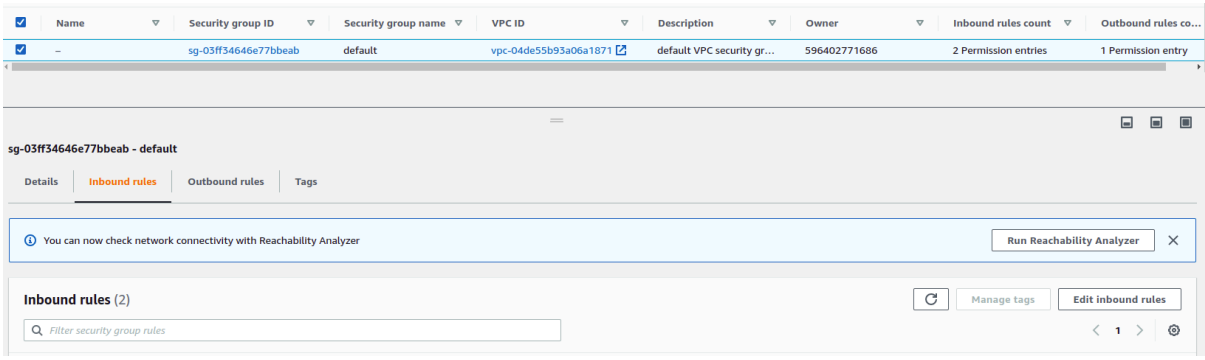



   d. Abrir una terminal y escribir “curl -v http://{SUSTITUYE}/alive” donde la palabra SUSTITUYE es el DNS Name de arriba. Sería algo así




   e. Como veis falla. Es porque al crear una VPC tenéis que abrir esa red a vuestro internet. Para ello vais al servicio EC2, Security groups, seleccionamos el que pone “default” y le dais a Inbound rules, edit Inbound

rules




f. Le dais a add rule y lo dejáis así


g. Le dais a save rule y volveis a probar a hacer la petición en la terminal. Os debería salir lo siguiente


# Conclusiones

En esta práctica hemos levantado un servicio con múltiples máquinas. Hemos dejado de lado muchísimos temas de seguridad y buenas prácticas por simplificar. Os animamos a que investigueis estos temas, ya que son imprescindibles si queremos implantar estas tecnologías en una empresa. Para la entrega final algunos de estos temas serán necesarios que los estudieis, como por ejemplo subir imágenes al registry o los grupos de seguridad.

Recordad borrar todo y si tenéis alguna duda preguntadnos sin miedo!