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 como orquestar un servicio de prueba. Habrán varios temas que por reducir la duración de esta práctica se ignoren (por ejemplo subir imágenes a un registry o grupos de seguridad de AWS). Para la entrega final estos temas tendrán que ser mejorados o al menos planteados como arreglarlos.

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. En este caso **NO queremos ocuparnos de las instancias EC2**. Veremos cómo funcionan las instancias Fargate, para quitarnos un dolor de cabeza de encima.

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 azul que dice “Create Cluster”. Clicamos en el
3. Como vamos a usar instancias AWS Fargate, clicamos en “Networking only”
4. Lo configuraremos de tal manera:
    a. Le ponemos un nombre. Por ejemplo “test-cluster”
    b. Creamos una VPC para este cluster. Dejamos los valores por defecto
    c. En tags ponemos. Key: “Name” Value: “test-cluster”
    d. NO clicamos en Enable Container Insights.
    e. Le damos a create
5. Ahora el cluster se estará creando. Puede que tarde un rato, así que esperamos un poco
6. Clicamos en “view cluster”

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 definition”
2. Le dais a “Create new Task Definition”. Así definiremos la tarea
3. Clicais en “Fargate”. Hacemos esto porque queremos usar este tipo de instancias en lugar de las de EC2
4. Lo configuramos de la siguiente manera
    a. Task definition name: “test-task” -> Nombre de la tarea
    b. Task role:
    c. Network mode: awsvpc (la de por defecto) -> La red donde se conecta
    d. Operating system family: linux -> Si queremos linux o windows
    e. Task execution IAM Role: Create new role (creará automáticamente ecsTaskExecutionRole) -> Temas de seguridad
    f. Task memory: 1GB -> Cuanta memoria máxima va a tener esta tarea
    g. Task CPU: 0.5 cpu -> Cuanto cómputo le vamos a dar a esta tarea

Ahora llegaremos a una parte donde hay un botón azul que dice “Add container”. Si clicamos en él veremos que se abre una nueva ventana y que nos pide el nombre del contenedor, su url, etc. Aquí Amazon nos está pidiendo que le indiquemos qué contenedor va a ejecutar.

Para ello tenemos que subir nuestro contenedor a su repositorio de contenedores. Este servicio se llama ECR (Elastic Container Registry). Os vamos a dar directamente la URL del contenedor y toda esta información, pero si queréis subir una imagen vuestra tendréis que investigar como subirla al ecr.

Por lo tanto clicamos en “Add Container” y añadimos lo siguiente:

- Container name: test-api-rest -> Nombre del contenedor
- Image: public.ecr.aws/n0h7v2z5/alba-digi-api-rest:latest -> URL del repositorio donde está el contenedor
- Memory Limits(Mib): (Soft Limit) 1024 -> Cuanta memoria le vamos a dar al contenedor. Le vamos a dar toda la memoria que más arriba (antes hemos marcado 1GB, que son 1024Mib)
- Port Mappings: 5050 - tcp -> Significa que abrimos el puerto 5050, que es el que pide este contenedor según su documentación
- Healthcheck: vamos a ignorarlo, lo controlaremos más adelante.
- Environment:
    - CPU Units: 512 (la que le hemos asignado a la tarea más arriba. Esto es así porque 0.5cpu = 512 cpu Units. 1cpu = 1024 cpu units según amazon.)
    - Essential lo dejamos checked, ya que queremos que si el contenedor se cae se reinicie todo.

- El resto lo dejamos tal cual y le damos al botón azul que está abajo del todo que pone “Add”
- Y bajamos al final y le damos al botón azul que pone “Create”. Vamos a “View Task Definition”

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. 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
    e. Security groups dejamos el de por defecto -> Tema de seguridad sería otro capítulo.
5. Y ahora veremos la sección de Listeners and Routing.

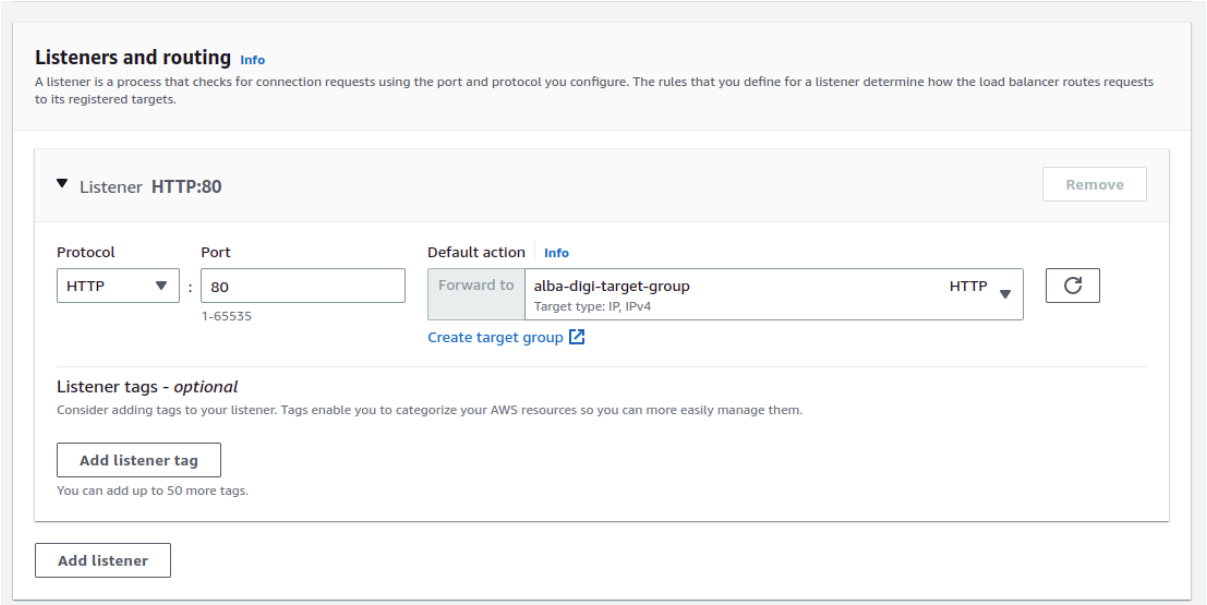
Ahora le vamos a explicar al balanceador a qué grupo de máquinas puede redirigir. Y para ello tenemos que crear ese grupo de máquinas. Será un grupo inteligente, así todo será automático. Será un “Target group”, así que vamos a clicar donde pone “Create target group”

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

1. Choose a target type: IP addresses (Las instancias de tipo Fargate son targets de tipo IP por alguna razón, así que dejamos la configuración así)
2. Target Group Name: test-target-group
3. Protocol: HTTP

4. Port: 5050 -> Más arriba hemos abierto el puerto 5050, que es donde funciona este contenedor. Cuando el balanceador apunte a este grupo de instancias, queremos que apunte en el puerto 5050.
5. VPC: Seleccionamos la que hemos creado anteriormente junto al cluster (que debería ser la única)
6. Protocol Version: HTTP1
7. Ahora sí que vamos a configurar el healthcheck
    a. Health Check Protocol: HTTP
    b. Health Check Path: /alive
8. Clicamos en Next. lo dejamos todo igual y luego 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.



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:

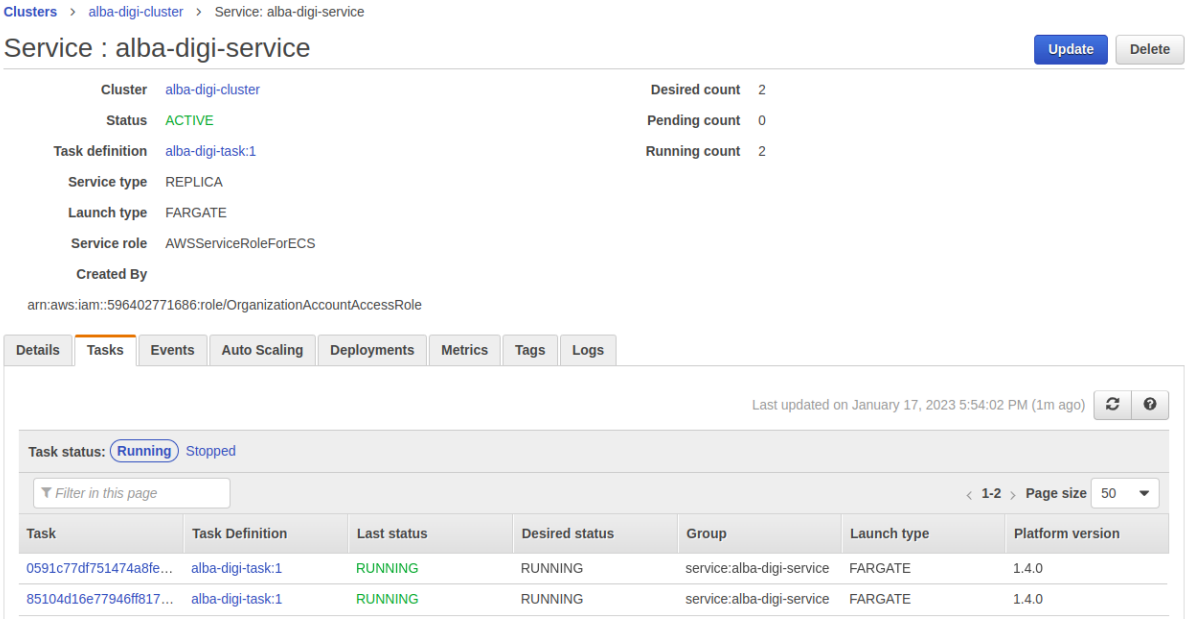
1. Volveremos al cluster que hemos creado anteriormente. Para ello vamos al servicio “Elastic Container Service” y clicamos en él.
2. Abajo veremos que está seleccionada la pestaña “Services”. Clicaremos en el botón de abajo, donde pone “Create”. Así empezaremos a crear el servicio
3. Lo configuraremos tal que así:
    a. Launch Type: Fargate (Habíamos decidido no usar ec2, y usar fargate)
    b. Operating System Family: Linux
    c. Task definition - Family: test-digi-task (la creada anteriormente)
    d. Task Definition - Revision: La que ponga (latest)
    e. Platform version: LATEST
    f. Cluster: test-digi-cluster
    g. Service Name: test-digi-service
    h. Number of tasks: 2 (vamos a poner más de una, que ya que creamos el servicio…)
    i. Minimum healthy percent: 100
    j. Maximum healthy percent: 200
    k. Deployment circuit breaker: Disabled (estos 3 últimos parámetros están relacionados en cómo actualizar servicios, una vez más se nos va un poco del tema)
4. Dejamos el resto igual y le damos a Next Step. Seguimos configurándolo
    a. Cluster VPC: La creada anteriormente en el primer paso
    b. Subnets: Seleccionamos todas
    c. Security groups, le damos a edit. saldrá una ventana y le damos a “Add rule”. Ponemos lo siguiente:
        i. Type: Custom TCP
        ii. Port range: 5050
        iii. Source: Anywhere
        iv. Clicamos en Save
    d. Auto-assign public ip: ENABLED
    e. Load Balancer Type: Application Load Balancer
    f. Volvemos un poco más arriba y donde pone Health Check grace period ponemos 30 (le vamos a dar 30 segundos antes de empezar a verificar si el servicio está sano)
    g. Load balancer name: test-load-balancer
5. En Container to load balance:
    a. Seleccionamos test-api-rest:5050:5050
    b. Clicamos en add to load balancer
    c. En production listener port seleccionamos 80:HTTP
    d. Target group name: test-target-group (el creado anteriormente, no dejamos la opción “create new”
    e. Clicamos en Next Step
6. Ahora le damos a “Do not adjust the service’s desired count” y le damos a next step. Esto está relacionado con un automatismo que nosotros no vamos a realizar, de ahí ignorarlo.
7. Le damos a “Create Service”.
8. Le damos a “View Service”

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

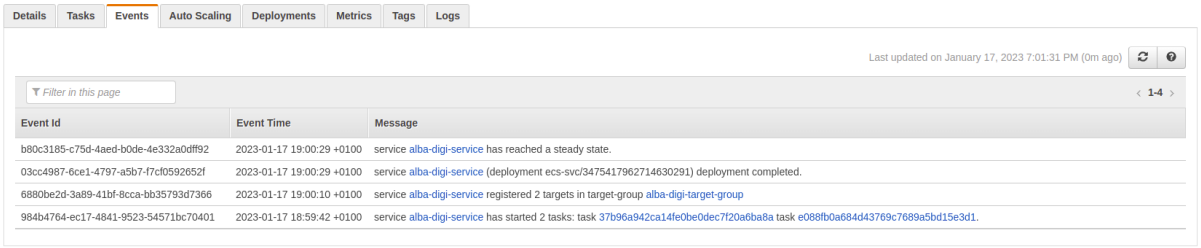
5. Probar el servicio

Para poder dar por buena la práctica tenéis que revisar esto

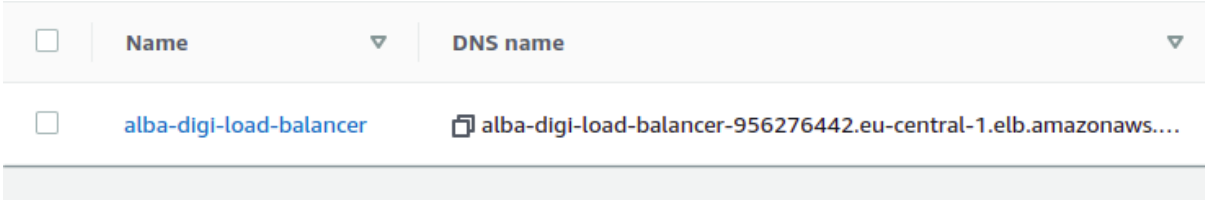
1. Servicio con sus 2 tasks en running, tal que así.



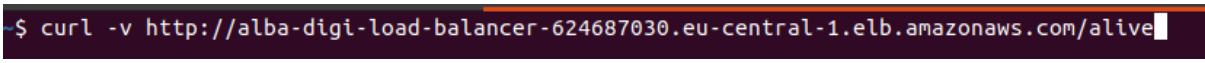
2. Revisar los eventos del servicio, donde se vea que el servicio está steady (puede que tarde un poco en aparecer)



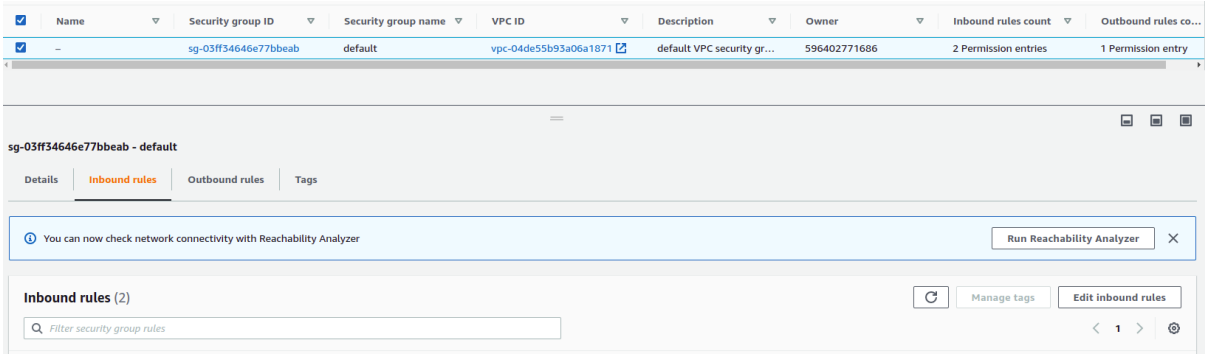
3. Probar que el servicio está ejecutándose. Para ello vamos a hacer lo siguiente:
   a. Ir al servicio EC2
   b. Ir al apartado “Load Balancers”
   c. Copiar el dns name, lo podeis encontrar aquí



   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!