Cómputo en Nube

AWS: EC2, Lambda y otras herramientas

Orquestadores de Contenedores

Tanto Fermín como yo trabajamos en el día a día con estas herramientas y hoy os presentaremos para que sirven, porque son necesarios y cuales son los más comunes.

Recordatorio: Contenedores

- Son una manera simple y segura de ejecutar programas y ponerlos en producción
- Utilizan una especie de emulación, haciendo el proceso más ligero que si emulamos todo un sistema operativo
- Generalmente utilizado para microservicios, pero puede ser usado para ejecutar cualquier tipo de aplicación

Contenedores: Revisión de contexto

- Dockerfile -> lo que quieres que haga la máquina
- Imágenes -> las instrucciones para construir la máquina
- Contenedores -> la máquina construida

- Pero lo realmente importante hoy son los contenedores
- Vamos a entender que un contenedor es una tarea/servicio/software funcionando.

¿Qué es un orquestador?

- Un orquestador es un sistema que organizar diferentes piezas para que estas funcionen en conjunto
- Podéis pensar en una orquesta, que tiene un director que dirige a la banda.
- Cada miembro de la banda es un contenedor
- Cada tipo de músico es una máquina (los vientos, las cuerdas, etc)
- El director es el orquestador, y él decide
    - Quien toca (que microservicio va a ejecutarse ahora)
    - Como toca (que va a realizar el microservicio)
    - Cuando toca (vigila si un microservicio tiene que esperar a otro)
    - Y vigila que nadie se quede dormido! (Si un contenedor se cae, levanta otro)

¿Qué es un orquestador?

- En el caso de los microservicios el orquestador dirige el flujo de las llamadas
  - Un microservicio es un servicio web que hace una tarea muy concreta.
  - En Netflix tendríamos un microservicio para ver tu perfil, otro para el algoritmo de recomendación, otro para stremear películas, etc.
  - Los microservicios hacen tareas muy concretas, y para conseguir un resultado final puede que tengas que llamar a varios de estos servicios
  - Por eso utilizas la app de Netflix. Al entrar carga tu perfil, te enseña recomendaciones y si clickas en una película la puedes ver
  - Netflix sería una especie de orquestador

¿Qué es un orquestador?

- En el caso de los contenedores, se encarga de que todo vaya bien
  - Verifica que siempre haya X contenedores funcionando
  - Si hay mucha carga eleva el número de contenedores
  - Si hay que hacer una actualización, reemplaza los contenedores en el orden correcto
  - Si hay que reiniciarlos, detecta que no funcionan correctamente y lo hace

Pero… ¿Para qué queremos un orquestador?

- Yo solo ejecuto un contenedor, ¡no me hace falta!
  - ¿Qué pasa si el contenedor falla y se cae?
  - Imaginemos que ejecutas el contenedor en un ordenador y se reinicia automáticamente, el contenedor se levantará solo?
  - ¿Y si la instancia se cae? ¿Cómo nos aseguramos que va todo bien?
  - Podría darse el caso que no necesitases un orquestador y haya una solución más rápida y simple, pero generalmente es una buena idea utilizar un sistema así
- Generalmente utilizar un orquestador es una **buena práctica**

Pero… ¿Para qué queremos un orquestador?

- Imaginemos un caso donde es crítico que el sistema siempre esté funcionando!
- **NO** podemos asegurar que un contenedor o la instancia nunca caiga. ¿Cómo hacemos?
- Utilizamos un orquestador que nos permitirá tener 2 o más contenedores ejecutándose a la vez.
  - Así, si uno falla, tendremos como mínimo otro funcionando
  - A esto se le llama HA -> High Availability -> Alta Disponibilidad
  - Esto también sirve para realizar actualizaciones manteniendo el sistema en funcionamiento
- Estos dos contenedores estarán en dos instancias/máquinas diferentes. Así si una falla, la otra seguirá funcionando

Pero… ¿Para qué queremos un orquestador?

- El orquestador va a ser capaz de aumentar el número de microservicios según la carga que este tenga
  - Si hay muchas peticiones, el sistema será capaz de escalar automáticamente
  - Si está bien configurado, escalará de una manera optimizada escalando solo las partes necesarias
- Es útil si queremos incorporar un sistema de CI/CD
  - CI = Continuous Integration
  - CD = Continuous Deployment
  - Se automatizan las actualizaciones
  - Digamos que el orquestador ayuda mucho a que todo funcione correctamente

¿Y qué orquestadores podemos elegir?

- Tenemos orquestadores muy muy básicos y simples, como por ejemplo Docker Swarm
  - Es el orquestador desarrollado por Docker
  - Muy muy simple y muy fácil de usar
  - Pero poco versátil
  - No se suele usar en producción y es sirve más para una introducción al mundo de los orquestadores

¿Y qué orquestadores podemos elegir?

- El siguiente nivel serían orquestadores como ECS o Nomad
  - ECS: Elastic Container Service. La solución de AWS
  - Nomad: Una solución de HashiCorp
- Ambos son soluciones más completas que Docker Swarm, pero que siguen siendo “sencillas” de gestionar
  - No acaban de ser una solución 100% completa, y suelen apoyarse en otras herramientas
  - Por ejemplo ECS es mayoritariamente dependiente de otros servicios de AWS
  - Nomad “requiere” instalarse junto a Consul, Vault o introducir un balanceador de carga

¿Y qué orquestadores podemos elegir?

- El sistema más completo y a la vez el más complejo es Kubernetes
- Es el que se usa en la industria
- Usar kubernetes “puro” es bastante complicado y por eso se suele complementar con plugins o servicios gestionados
    - Los plugins son herramientas que simplifican tareas
    - Los servicios gestionados son versiones “mejoradas” de kubernetes. Como una compilación de plugins

¿Y qué orquestadores podemos elegir?

- Los servicios gestionados más conocidos son:
  - EKS: Elastic Kubernetes Service. La solución de AWS
  - AKS: Azure Kubernetes Service. La solución de Azure
  - GKE: Google Kubernetes Engine. La solución de Google
- Tienen las mismas funciones, pero la instalación es más simple y además añaden una interfaz gráfica
- Kubernetes es tan complicado que las empresas han tenido que encontrar maneras de simplificarlo

ECS

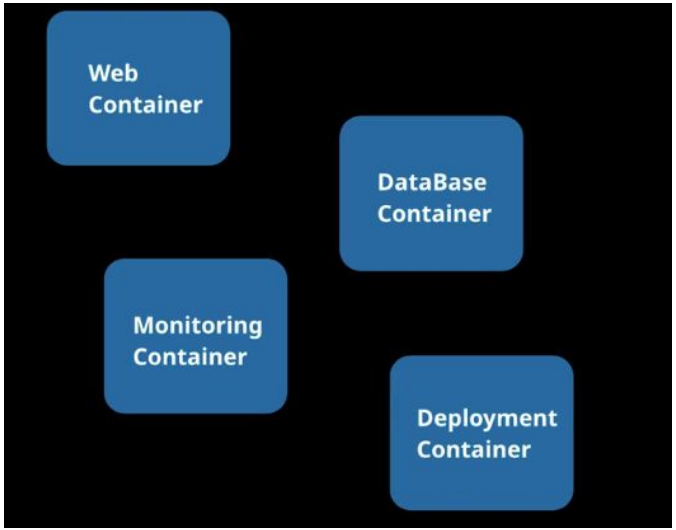
- Como hemos visto kubernetes no es fácil
- Pero nosotros no vamos a estudiar kubernetes :)
- Aunque es el standard de industria y sería muy interesante estudiarlo, no tenemos tiempo suficiente
- Es por eso que vamos a estudiar una solución intermedia: ECS.
- Encaja muy bien con lo que estáis haciendo en la práctica y creemos que va a ser lo más fácil

Introducción a ECS

- Como hemos visto kubernetes no es fácil
- Pero nosotros no vamos a estudiar kubernetes :)
- Aunque es el standard de industria y sería muy interesante estudiarlo, no tenemos tiempo suficiente
- Es por eso que vamos a estudiar una solución intermedia: ECS.
- Encaja muy bien con lo que estáis haciendo en la práctica y creemos que va a ser lo más fácil

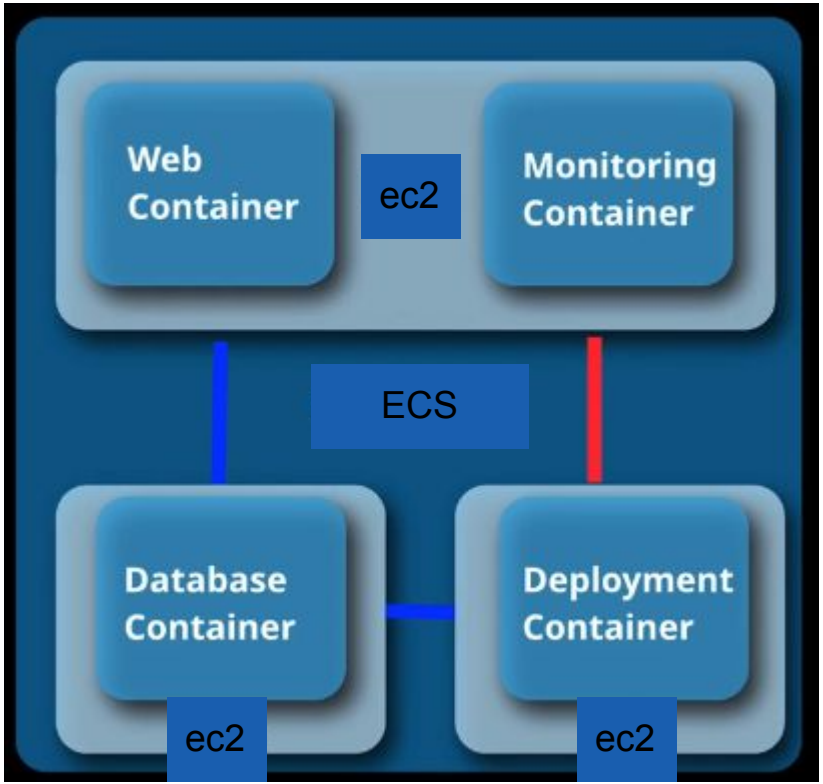
Introducción a ECS

- En un entorno podemos tener varios microservicios/contenedores
- Estos tienen o no tienen que comunicarse entre ellos (dependiendo el caso)
- Y estos pueden o no pueden escalar de la misma forma
- Así que necesitamos organizarlos



Introducción a ECS

- ECS organiza su servicio en tareas
- Cada tarea es un contenedor
- Puede haber una o varias tareas por instancia
- Los contenedores se pueden comunicar entre ellos si lo permitimos
- Los contenedores deberíamos organizarlos lógicamente
  - Si un contenedor funciona independientemente, sólo en una máquina
  - Si un contenedor requiere otro funcionando a su lado, sería mejor opción tener los dos juntos
  - Salvo que… no escalen de la misma forma



Instalación de ECS

- Una de las ventajas de ECS es que no hay que instalar prácticamente nada
- Se integra muy bien con AWS y hace que su uso sea muy fácil
- Sólo necesitamos:
  - Crear un cluster para el servicio
    - Tener más de una instancia EC2 (autoscaling)
    - VPC + Subnets (networking)
  - Crear la(s) tarea(s) para ejecutar en el servicio
  - Poner un balanceador de servicio (para que sepa a qué máquina apuntar)
  - Crear el servicio
  - Abrir ciertos grupos de seguridad
- Ahora vamos a ver punto por punto que nos hace falta para esto

Instalación de ECS: Cluster

- Si queremos tener un servicio necesitamos más de una instancia funcionando
- A esto es lo que llamamos un “cluster”. Un conjunto de máquinas funcionando a la vez con un mismo objetivo
- Luego hay 2 tareas más:
  - Como no queremos levantarlas a mano, haremos un autoscaling group
    - Como la palabra dice, son grupos de autoescalado
    - Lo que le indicaremos a AWS es que queremos que mantenga siempre un número de instancias levantadas automáticamente
  - También nos tenemos que encargar de todas las redes
  - Todo esto se lo pediremos que lo haga a AWS al crear el cluster. Nos preguntará ciertas cosas pero nos ayudará a hacerlo

Instalación de ECS: Creación de las tareas

- Para poder crear un servicio antes tenemos que elegir qué tarea ejecutaremos en el servicio
- Recordemos que una tarea es un contenedor
  - En realidad una tarea podría ser algo más, pero por simplificar vamos a decir que tarea=contenedor
- Para ello definiremos una nueva tarea
  - Nos pedirá que contenedor queremos desplegar
  - Nos preguntará por su configuración y todo lo que le hace falta para desplegar un contenedor automáticamente

Instalación de ECS: Creación del balanceador

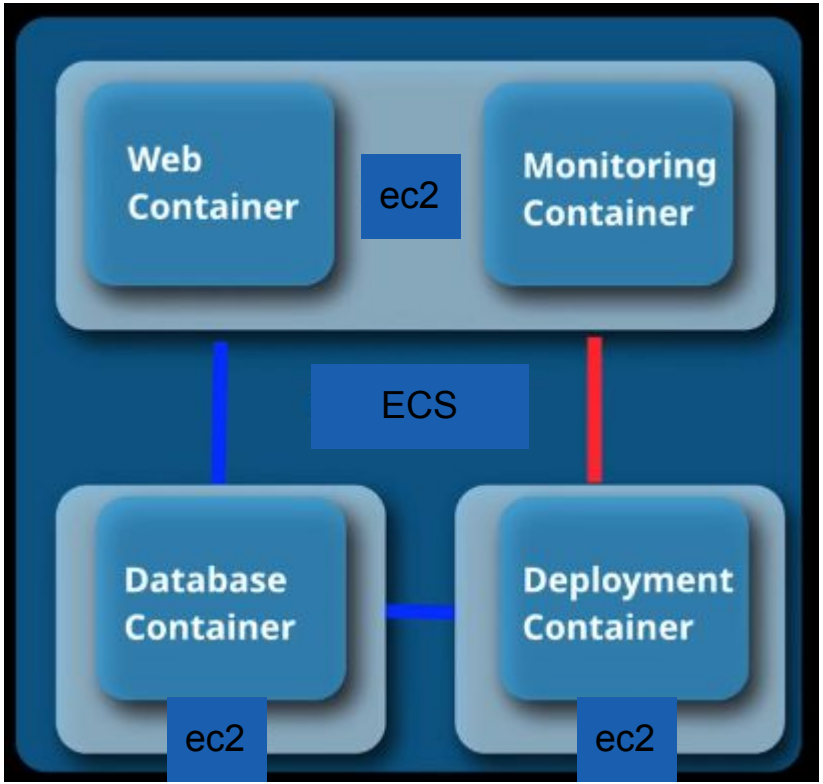
- Ya casi estamos preparados para crear un servicio
- Pero recordemos que uno de los puntos principales de tener un servicio es poder tener varias máquinas a la vez
- Pero… ¿Cómo podemos saber a qué máquina llamamos?
  - Es parecido a cuando vas a un piso viejo que tiene una puerta de servicio, ¿a cual llamas?
- Necesitamos un balanceador de carga para ello
  - Vendría a ser como poner un mayordomo que te indica por qué puerta pasar
- Así cuando llamemos al servicio sabremos a qué instancias apuntar.

Instalación de ECS: Crear Servicio

- Ahora que ya tenemos la tarea, definiremos el servicio
- El servicio como tal es quien se encarga de cómo se está ejecutando la tarea
  - Tarea = Contenedor
  - Servicio = Cuantas tareas
  - Entonces… Servicio = Unos cuantos contenedores
- Completamos el formulario y así tendremos un servicio funcionando

Instalación de ECS: Grupos de seguridad

- Como vemos en la imagen los contenedores y las instancias se tienen que comunicar entre ellas
- Es por eso que tendremos que “abrir los grupos de seguridad”
- Los grupos de seguridad son formas de dar permisos a los servidores de comunicarse entre ellos
- En nuestro caso no nos vamos a meter con este tema, pero es interesante ser consciente de ello



ALBAÑILESDIGITALES

Developing your new career

servicio navarro de empleo

das·nano

veri·das