

Lección 2

Creación de aplicaciones de microservicios con los servicios de contenedores de AWS



¿Qué es un contenedor?

Su código de la aplicación



Motor de tiempo de ejecución



Dependencias



Configuraciones



Cuando se construye una arquitectura de microservicios, se pueden utilizar contenedores para la potencia de procesamiento.

Los contenedores son un método de virtualización del sistema operativo que le permite ejecutar una aplicación y sus dependencias en procesos aislados de recursos. Un contenedor es un paquete de software ligero e independiente. Contiene todo lo que una aplicación de software necesita para ejecutarse, como el código de la aplicación, el motor de ejecución, las herramientas del sistema, las bibliotecas del sistema y las configuraciones.

Los contenedores pueden resolver el problema

Lograr que el software funcione correctamente en distintos entornos de trabajo

Estación de trabajo del desarrollador



Entorno de producción



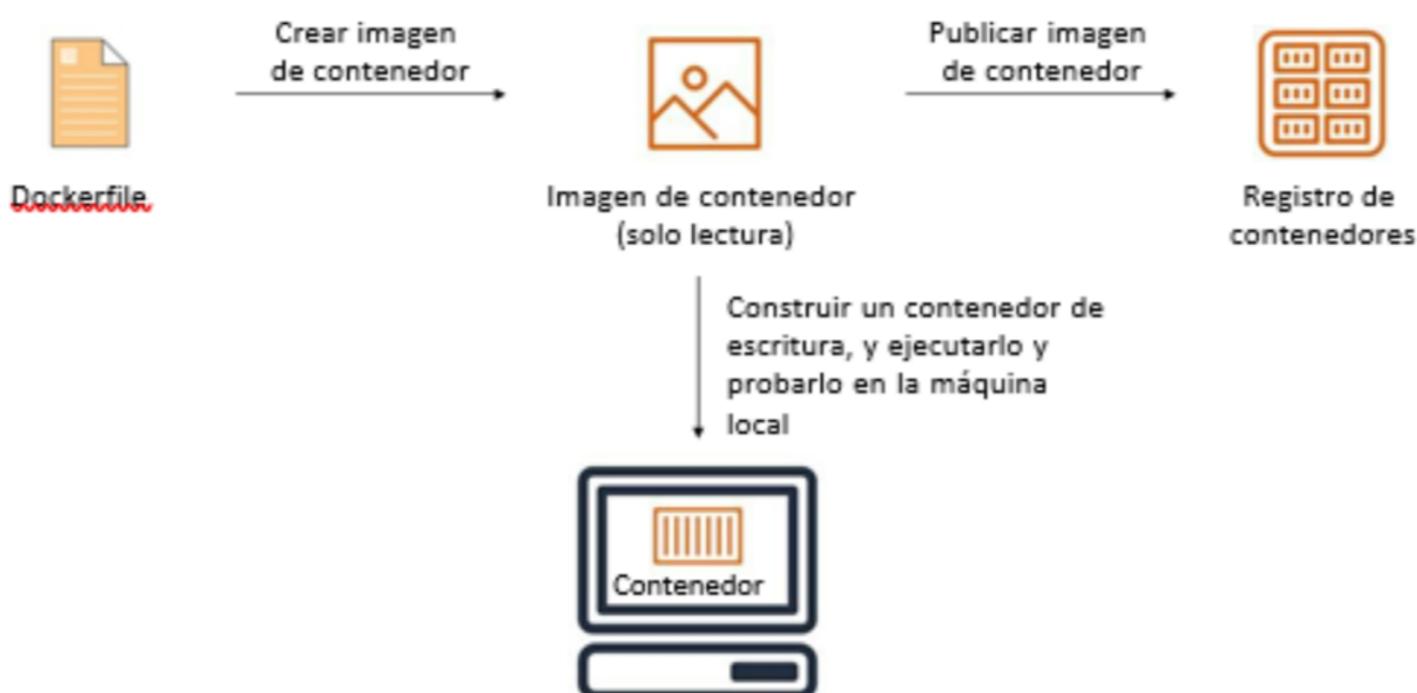
Entorno de pruebas



Los contenedores pueden ayudar a garantizar que las aplicaciones se implementen de forma rápida, fiable y coherente, cualquiera que sea el entorno de implementación. Los contenedores también le ofrecen más control granular sobre los recursos, lo que mejora la eficiencia de su infraestructura.



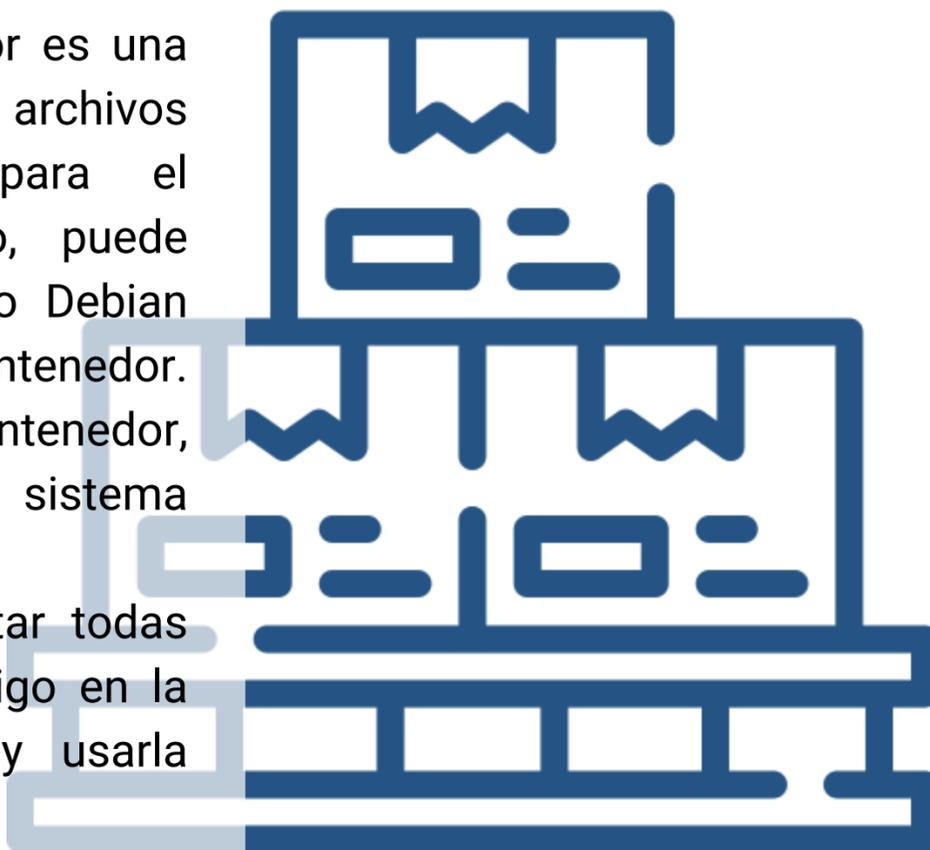
Tecnología de contenedores



Un contenedor se crea a partir de una plantilla de solo lectura que se denomina imagen. Las imágenes se suelen crear a partir de un Dockerfile, que es un archivo de texto plano que especifica todos los componentes incluidos en el contenedor. Puede crear imágenes desde cero o utilizar imágenes creadas por otros y publicadas en un registro de contenedores público o privado.

Una imagen de contenedor es una instantánea del sistema de archivos que está disponible para el contenedor. Por ejemplo, puede tener el sistema operativo Debian como una imagen de contenedor. Cuando ejecute este contenedor, tendrá disponible un sistema operativo Debian.

También puede empaquetar todas sus dependencias de código en la imagen del contenedor y usarla como artefacto de código.



Las imágenes de los contenedores se almacenan en un registro. Puede descargar las imágenes del registro y ejecutarlas en su clúster. Los registros pueden existir dentro o fuera de su infraestructura de AWS.

Amazon ECS



Amazon Elastic Container Service (Amazon ECS)

- Coordina la ejecución de los contenedores
- Mantiene y escala la flota de instancias que ejecutan sus contenedores
- Elimina la complejidad de crear la infraestructura

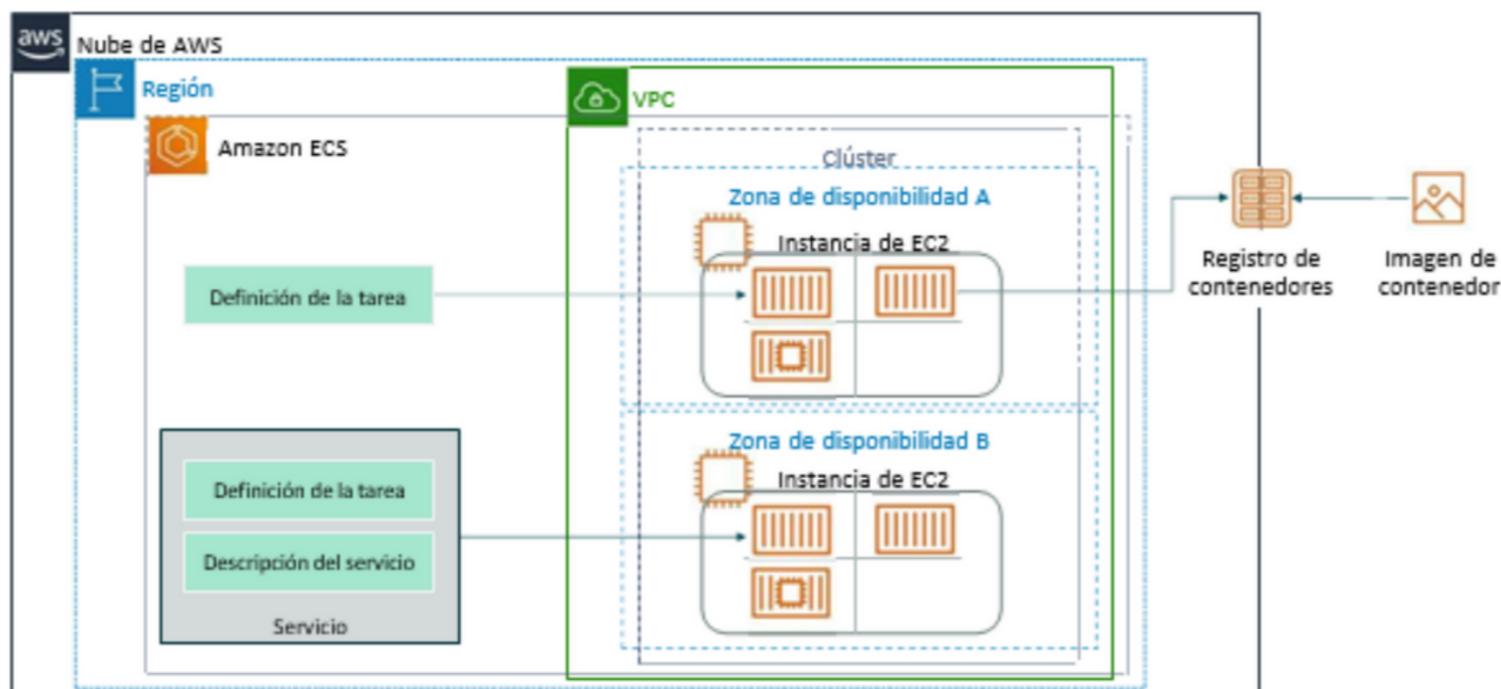
Puede ejecutar sus contenedores de Amazon Elastic Container Service (Amazon ECS). Amazon ECS es un servicio de administración de contenedores altamente escalable y de alto rendimiento. Admite contenedores de Docker y permite ejecutar aplicaciones fácilmente en un clúster administrado de instancias de Amazon Elastic Compute Cloud (Amazon EC2).

Amazon ECS es un servicio de clúster escalable para alojar contenedores que:

- Puede escalar hasta miles de contenedores de Docker en segundos
- Supervisa la implementación de contenedores
- Administra el estado del clúster que ejecuta los contenedores
- Programa contenedores con un programador integrado o de terceros (Apache Mesos o Blox)
- Se puede ampliar mediante API
- Se puede iniciar con los tipos de lanzamiento de AWS Fargate o Amazon EC2

Puede ejecutar clústeres de ECS a escala combinando instancias de spot con instancias bajo demanda y reservadas.

Amazon ECS coordina contenedores



Amazon ECS es un servicio regional que simplifica la ejecución de contenedores de aplicaciones con alta disponibilidad en varias zonas de disponibilidad dentro de una región. Puede crear clústeres de ECS en una nube virtual privada (VPC) nueva o existente. Un clúster es un agrupamiento lógico de recursos.

Una vez que un clúster está en funcionamiento, puede definir tareas y servicios que especifiquen qué imágenes de contenedores de Docker se ejecutarán en los clústeres.

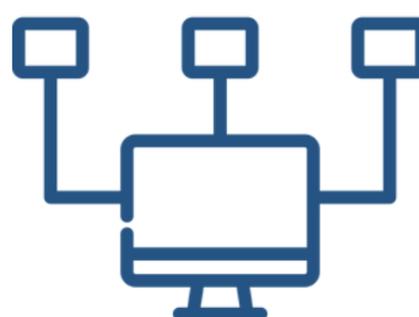


Una definición de tarea es un archivo de texto en formato JavaScript Object Notation (JSON). Describe uno o más contenedores, hasta un máximo de 10, que forman la aplicación. Puede considerarlo como un esquema para su aplicación. Las definiciones de tareas especifican parámetros para su aplicación, por ejemplo, los contenedores y el tipo de inicio que debe utilizar. Otros parámetros incluyen qué puertos se deben abrir para su aplicación y qué volúmenes de datos se deben utilizar con los contenedores en la tarea.

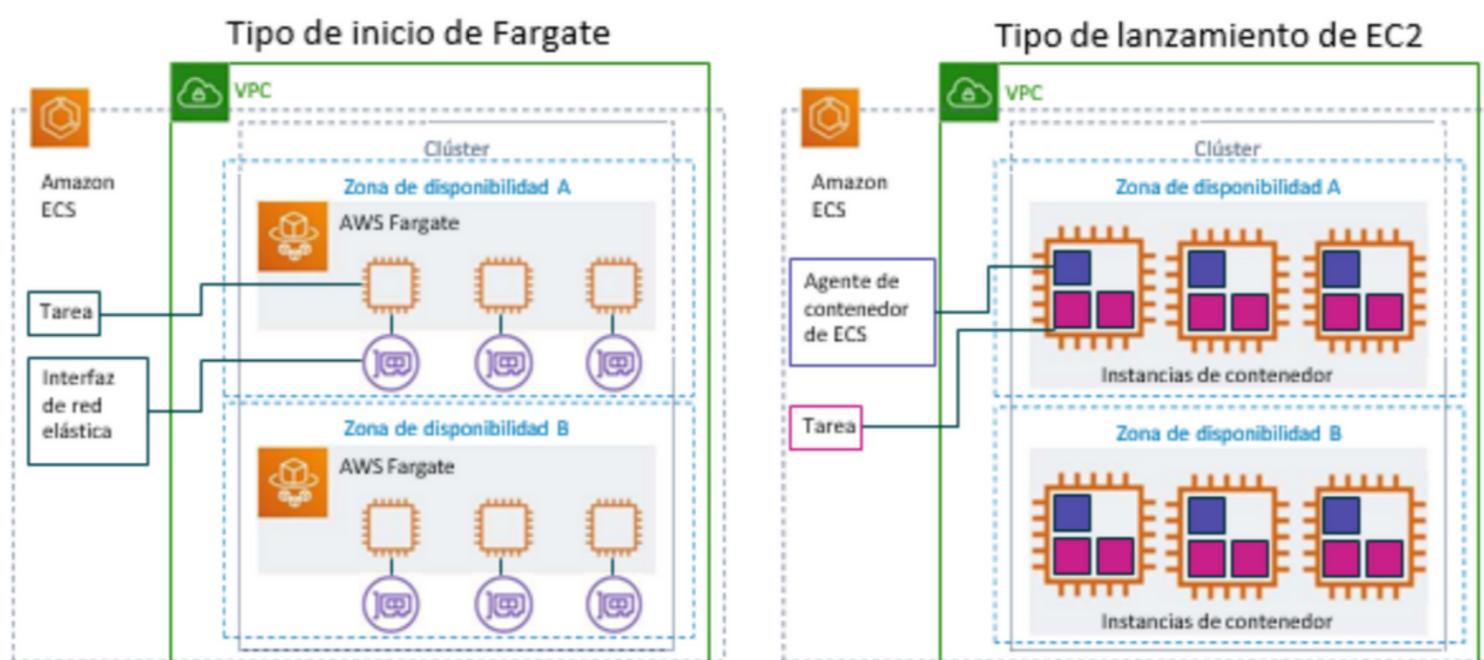
Un servicio le permite determinar cuántas copias de la definición de tareas se ejecutarán y mantendrán en un clúster. Si lo desea, puede elegir utilizar un equilibrador de carga de Elastic Load Balancing para distribuir el tráfico entrante a los contenedores de su servicio. Amazon ECS mantiene ese número de tareas y coordina la programación de las tareas con el equilibrador de carga.

Después de crear una definición de tarea para su aplicación, puede especificar la cantidad de tareas que se ejecutarán en el clúster. Una tarea es la instancia de una definición de tareas dentro de un clúster. Cuando utiliza Amazon ECS para ejecutar tareas, estas se colocan en un clúster.

Amazon ECS descarga las imágenes de contenedor desde un registro específico y las ejecuta en el clúster.



Tipos de lanzamiento de Amazon ECS

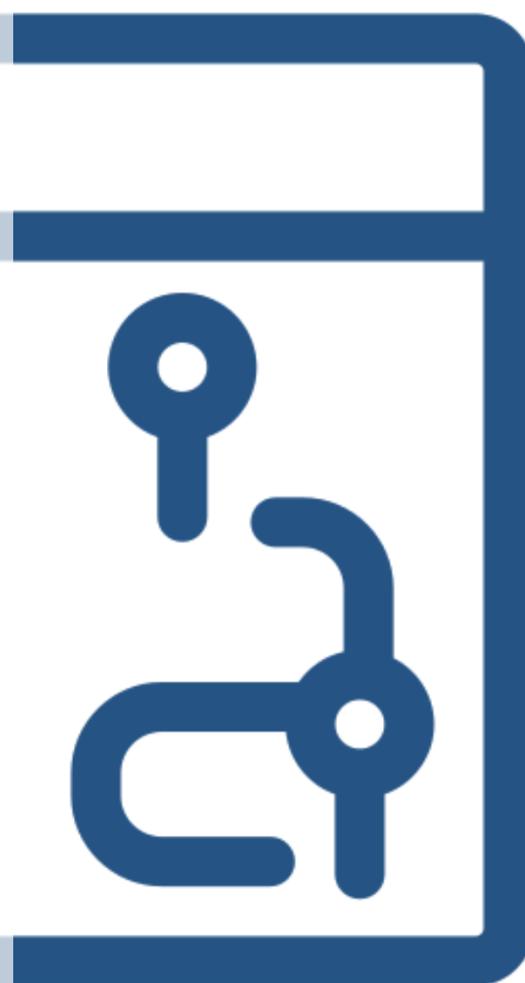


Amazon ECS ofrece dos tipos de lanzamiento para alojar sus aplicaciones en contenedores.

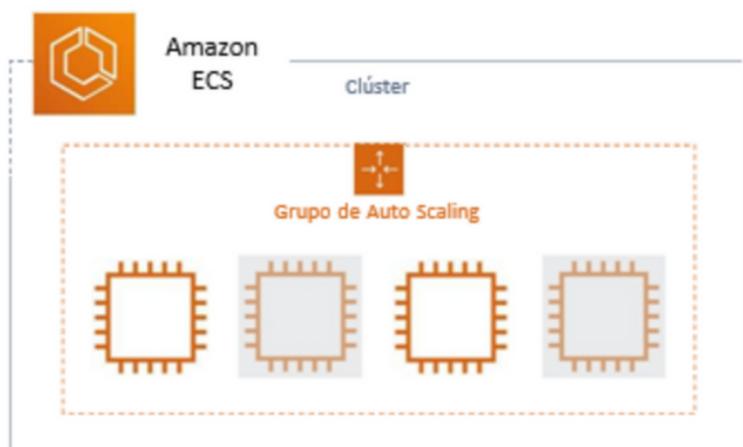
Puede utilizar el tipo de lanzamiento de Fargate para alojar su clúster en una infraestructura sin servidor administrada por Amazon ECS. Solo necesita empaquetar la aplicación en contenedores, especificar los requisitos de CPU y memoria, definir las políticas de redes y de AWS Identity and Access Management (AWS IAM) e iniciar la aplicación.

Por otro lado, si desea un mayor control, puede utilizar el tipo de lanzamiento de EC2 para alojar sus tareas en un clúster de instancias de contenedor de EC2 que usted administre. Una instancia de contenedor es una instancia de EC2 que ejecuta el agente de contenedores de Amazon ECS. Puede usar Amazon ECS para programar la ubicación de los contenedores en el clúster en función de sus necesidades de recursos, las políticas de aislamiento y los requisitos de disponibilidad. Para obtener información sobre las diferentes opciones de programación, consulte [Programar las tareas de Amazon ECS](#). Amazon ECS realiza un seguimiento de toda la CPU, la memoria y otros recursos del clúster. También encuentra el mejor servidor para que ejecute el contenedor en función de los requisitos de los recursos necesarios.

Para obtener más información sobre los tipos de lanzamiento de Fargate y EC2, consulte [Tipos de lanzamiento de Amazon ECS](#).



Escalado automático de clústeres de Amazon ECS

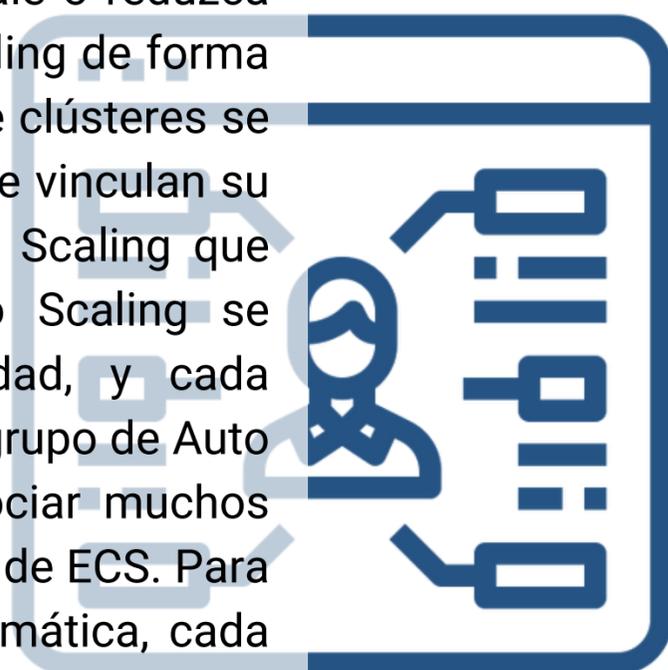


Puede crear un grupo de Auto Scaling para un clúster de Amazon ECS. El grupo de Auto Scaling incluye instancias de contenedor que puede escalar horizontalmente (y reducir) mediante las alarmas de Amazon CloudWatch.

Si configura su grupo de Auto Scaling para eliminar instancias de contenedor, se detendrán todas las tareas que se estén ejecutando en las instancias de contenedor eliminadas. Si sus tareas se ejecutan como parte de un servicio, Amazon ECS las reinicia en otra instancia si los recursos necesarios están disponibles. Algunos ejemplos de los recursos necesarios son la CPU, la memoria y los puertos. Sin embargo, las tareas iniciadas de forma manual no se reinician automáticamente.

También puede aprovechar el escalado automático de clústeres de Amazon ECS, que le ofrece más control sobre cómo escalar las tareas en un clúster. Aumenta la velocidad y la fiabilidad del escalado de clústeres. Le ofrece control sobre la cantidad de capacidad adicional que se mantiene en su clúster y administra automáticamente la terminación de instancias al reducir horizontalmente.

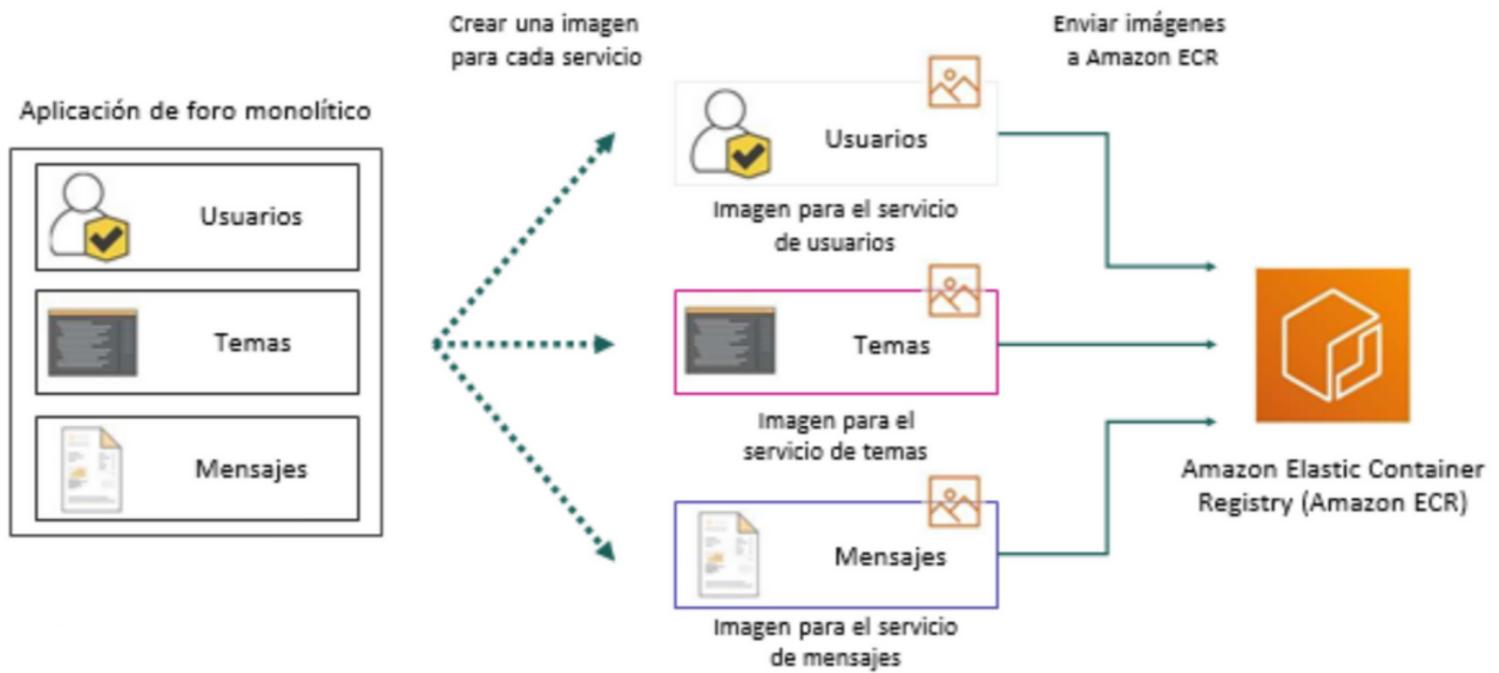
Con el escalado automático del clúster, puede configurar Amazon ECS para que escale o reduzca horizontalmente su grupo de Auto Scaling de forma automática. El escalado automático de clústeres se basa en proveedores de capacidad, que vinculan su clúster de ECS a los grupos de Auto Scaling que desea utilizar. Cada grupo de Auto Scaling se asocia a un proveedor de capacidad, y cada proveedor de capacidad tiene un solo grupo de Auto Scaling. Sin embargo, se pueden asociar muchos proveedores de capacidad a un clúster de ECS. Para escalar todo el clúster de forma automática, cada proveedor de capacidad administra el escalado de su grupo de Auto Scaling asociado.



Para obtener más información sobre el escalado automático de clústeres, consulte la publicación del blog de noticias de AWS [Escalado automático de clústeres de Amazon ECS](#).

Descomposición de monolitos

Paso 1: Crear imágenes de contenedor

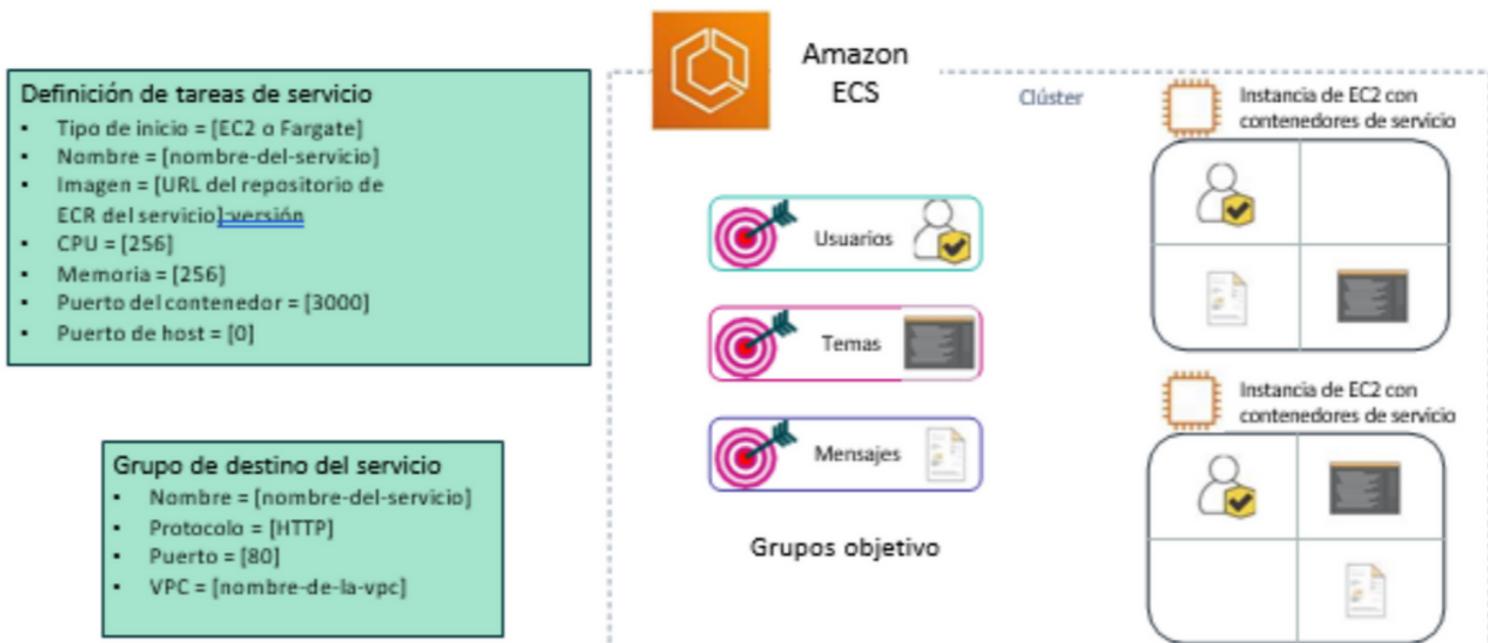


De nuevo, considere la aplicación de foro monolítico que vimos anteriormente, en la que toda la aplicación se ejecuta como un único servicio. Para rediseñar esta aplicación utilizando una arquitectura de microservicios, puede ejecutar cada proceso de la aplicación como un servicio independiente en su propio contenedor. Con una arquitectura de microservicios, los servicios pueden escalar y actualizarse sin depender de los demás.

Para implementar la aplicación monolítica como una aplicación de microservicios, primero cree y etiquete una imagen para cada servicio. Luego, registre las imágenes con Amazon Elastic Container Registry (Amazon ECR).

Descomposición de monolitos

Paso 2: Crear la definición de tareas de servicio y grupos objetivo

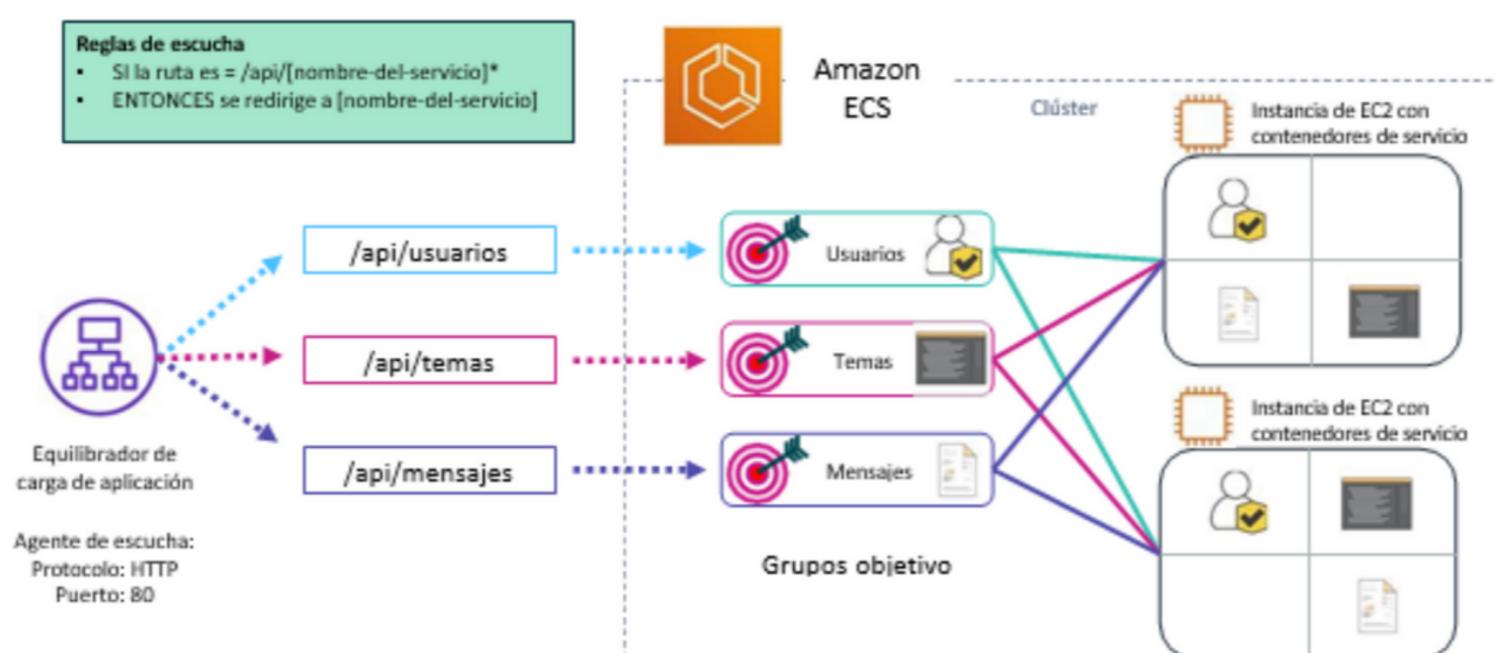


A continuación, elija el tipo de lanzamiento y cree un nuevo servicio para cada pieza de la aplicación monolítica original. Amazon ECS implementa cada servicio en su propio contenedor en un clúster de ECS.

A continuación, cree un grupo objetivo para cada servicio. El grupo objetivo rastrea las instancias y puertos de cada contenedor que se está ejecutando para ese servicio.

Descomposición de monolitos

Paso 3: Conectar el equilibrador de carga a los servicios



Por último, cree un equilibrador de carga de aplicación y configure reglas de escucha para conectar los servicios. El agente de escucha comprueba las solicitudes de conexión entrantes al equilibrador de carga y utiliza las reglas para enrutar el tráfico de forma adecuada. En el ejemplo, el agente de escucha del equilibrador de carga de aplicación escucha las peticiones de servicio HTTP en el puerto 80 y las enruta al servicio correspondiente.

Herramientas para crear arquitecturas de microservicios de alta disponibilidad



- Es un servicio de descubrimiento completamente administrado para recursos en la nube
- Se puede utilizar para definir nombres personalizados de los recursos de la aplicación
- Mantiene actualizada la ubicación de los recursos que se modifican dinámicamente, lo que aumenta la disponibilidad de las aplicaciones.

- Captura métricas, registros y rastreos de todos sus microservicios
- Permite exportar estos datos a Amazon CloudWatch, AWS X-Ray y herramientas compatibles con la red de socios de AWS (APN) y la comunidad
- Permite controlar los flujos de tráfico entre microservicios para garantizar que los servicios tengan alta disponibilidad

AWS Cloud Map y AWS App Mesh son dos herramientas que pueden ayudarle a crear arquitecturas de microservicios de alta disponibilidad.

AWS Cloud Map es un servicio de descubrimiento completamente administrado para recursos en la nube. Puede utilizarlo para definir nombres personalizados para los recursos de su aplicación (como bases de datos, colas, microservicios y otros recursos en la nube). AWS Cloud Map mantiene actualizada la ubicación de estos recursos que se modifican dinámicamente. Este mantenimiento de la ubicación aumenta la disponibilidad de su aplicación porque su servicio web siempre descubre las ubicaciones más actualizadas de sus recursos. Puede agregar y registrar cualquier recurso con una intervención manual mínima del mapeo. AWS Cloud Map le ayuda con el descubrimiento de servicios, la integración continua y la supervisión del estado de sus microservicios y aplicaciones.

Para obtener más información sobre AWS Cloud Map, lea esta publicación del blog [Código abierto de AWS](#). Para obtener más información sobre cómo puede utilizar AWS Cloud Map para permitir que sus servicios en contenedores se descubran y conecten entre sí, lea [AWS Fargate, Amazon EKS y Amazon ECS ahora se integran con AWS Cloud Map](#).

Cuando cree sus definiciones de tareas, puede habilitar la integración de App Mesh.

AWS App Mesh captura métricas, registros y rastreos de todos sus microservicios. Puede exportar estos datos a Amazon CloudWatch, AWS X-Ray y herramientas compatibles con la red de socios de AWS (APN) y la comunidad para la supervisión y rastreo. AWS App Mesh también le permite controlar cómo fluye el tráfico entre los microservicios para garantizar que todos los servicios dispongan de alta disponibilidad durante las implementaciones, después de los errores y a medida que se escala la aplicación.

App Mesh le permite configurar microservicios para que se conecten directamente entre sí a través de un proxy en lugar de necesitar un código dentro de la aplicación o utilizar un equilibrador de carga. App Mesh utiliza Envoy, un proxy de malla de servicios de código abierto, que se implementa junto a sus contenedores de microservicios.



Para obtener más información sobre AWS Cloud Map y AWS App Mesh, consulte este video de [AWS en YouTube](#).

AWS Fargate



AWS
Fargate

- Es un servicio de contenedores completamente administrado
- Funciona con Amazon Elastic Container Service (Amazon ECS) y Amazon Elastic Kubernetes Service (Amazon EKS)
- Aprovisiona, administra y escala sus clústeres de contenedores
- Administra el entorno en tiempo de ejecución
- Brinda escalado automático

En esta Lección, aprendió que Amazon ECS ofrece dos tipos de lanzamiento: EC2 y Fargate.

AWS Fargate es un servicio de contenedores completamente administrado que funciona tanto con Amazon ECS como con Amazon Elastic Kubernetes Service (Amazon EKS). Le permite ejecutar contenedores sin necesidad de administrar servidores o clústeres. Con AWS Fargate, ya no es necesario aprovisionar, configurar ni escalar clústeres de máquinas virtuales para ejecutar contenedores. Por lo tanto, no es necesario elegir tipos de servidores, decidir cuándo escalar los clústeres u optimizar conjuntos de clústeres. AWS Fargate reduce la necesidad de interactuar con servidores ni clústeres, ni siquiera pensar en ellos. Fargate le permite concentrarse en el diseño y la creación de aplicaciones, en lugar de administrar la infraestructura que las ejecuta.

Entre los puntos clave de esta Lección, se incluyen los siguientes:

- Amazon ECS es un servicio de administración de contenedores altamente escalable y de alto rendimiento. Admite contenedores de Docker y permite ejecutar aplicaciones fácilmente en un clúster administrado de instancias de Amazon EC2.
- El escalado automático de clústeres le ofrece más control sobre cómo escalar tareas con un clúster.
- AWS Cloud Map permite definir nombres personalizados para los recursos de la aplicación. Mantiene actualizada la ubicación de estos recursos que se modifican dinámicamente.
- AWS App Mesh es una malla de servicios que proporciona redes a nivel de aplicación para facilitar la comunicación entre los servicios a través de varios tipos de infraestructura de cómputo.
- AWS App Mesh es una malla de servicios que proporciona redes a nivel de aplicación. Permite que sus servicios se comuniquen fácilmente entre sí a través de diversos tipos de infraestructuras de cómputo.
- AWS Fargate es un servicio de contenedores completamente administrado que permite ejecutar contenedores sin necesidad de administrar servidores o clústeres.