aws

Guía de prácticas recomendadas de

Amazon Elastic Container Service

Amazon Elastic Container Service: Guía de prácticas recomendadas de

Table of Contents

Introduction

Amazon Elastic Container Service (Amazon ECS) es un servicio de administración de contenedores muy escalable y rápido que le facilita la tarea de ejecutar, detener y administrar contenedores en un clúster. En esta guía se describen muchas de las mejores prácticas operativas más importantes, al tiempo que se explican los temas principales en los que se basa el funcionamiento de las aplicaciones basadas en Amazon ECS. El objetivo es proporcionar un enfoque concreto y práctico para operar y solucionar problemas de aplicaciones basadas en Amazon ECS.

Esta guía se revisará periódicamente para incorporar las nuevas prácticas recomendadas de Amazon ECS. Si tiene alguna pregunta o comentarios acerca de cualquiera de los contenidos de esta guía, plantee un problema en el repositorio de GitHub. Para obtener más información, consulteGuía de prácticas recomendadas de Amazon ECSen GitHub.

- Prácticas recomendadas de redes
- Prácticas recomendadas: escalado automático y administración de capacidad
- Prácticas recomendadas - Almacenamiento persistente
- Prácticas recomendadas: seguridad

Prácticas recomendadas de redes

Las aplicaciones modernas generalmente se construyen a partir de múltiples componentes distribuidos que se comunican entre sí. Por ejemplo, una aplicación móvil o web podría comunicarse con un endpoint API, y la API podría estar alimentada por varios microservicios que se comunican a través de Internet.

Esta guía presenta las prácticas recomendadas para crear una red en la que los componentes de la aplicación puedan comunicarse entre sí de forma segura y escalable.

Temas

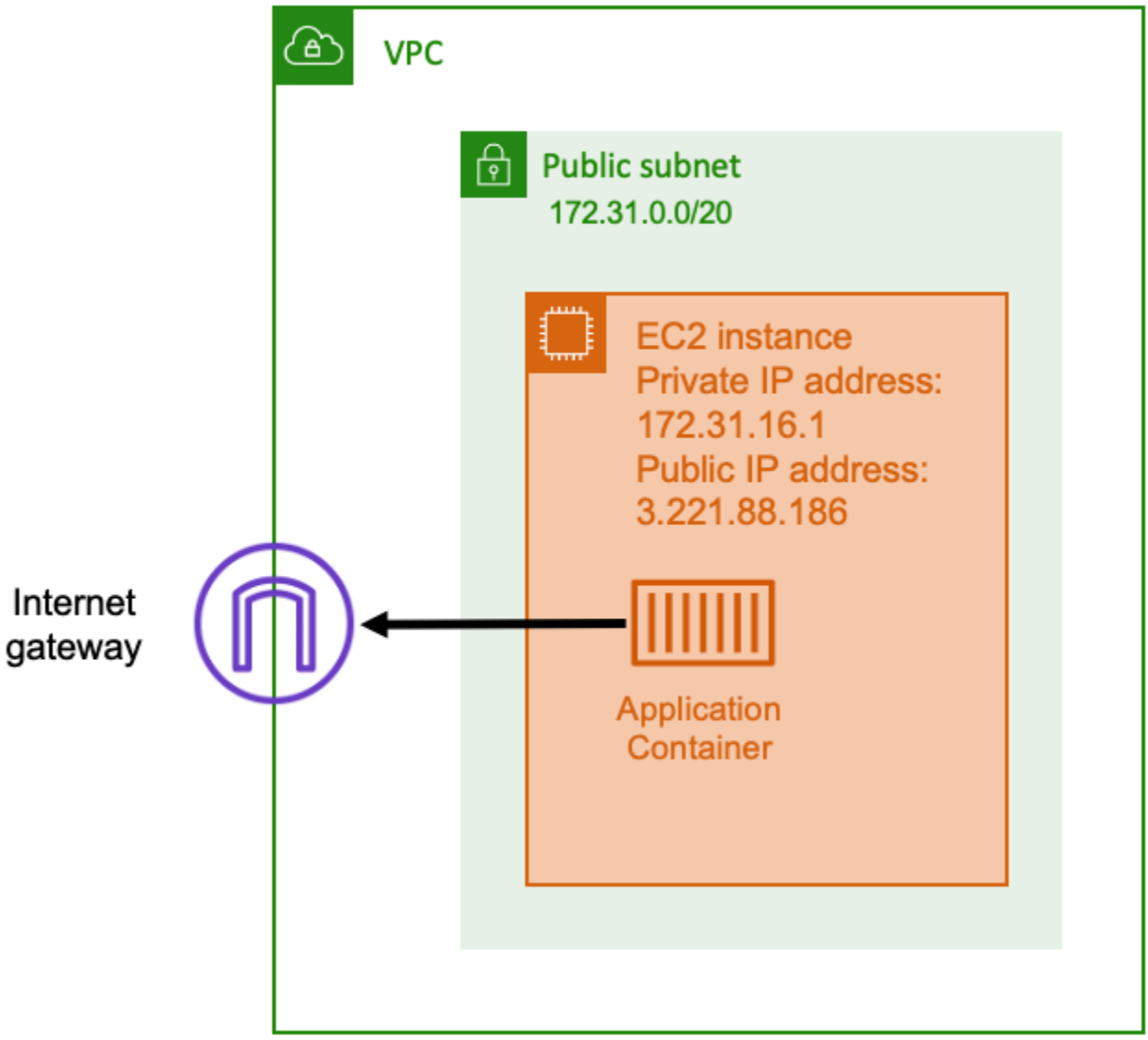
Conexión a Internet

La mayoría de las aplicaciones contenedorizadas tienen al menos algunos componentes que necesitan acceso saliente a Internet. Por ejemplo, el back-end de una aplicación móvil requiere acceso saliente para notificaciones push.

Amazon Virtual Private Cloud cuenta con dos métodos principales para facilitar la comunicación entre su VPC e Internet.

Temas

Uso de una subred pública y una puerta de enlace a Internet



Si utiliza una subred pública que tiene una ruta a una gateway de Internet, la aplicación en contenedores puede ejecutarse en un host dentro de una VPC en una subred pública. Se le asigna una dirección IP pública al host que ejecuta el contenedor. Esta dirección IP pública se puede enrutar desde Internet. Para obtener más información, consulte Puertos de enlace a internet en la Guía del usuario de Amazon VPC.

Esta arquitectura de red facilita la comunicación directa entre el host que ejecuta la aplicación y otros hosts en Internet. La comunicación es bidireccional. Esto significa que no solo puede establecer una conexión saliente con cualquier otro host en Internet, sino que otros hosts de Internet también

podrían intentar conectarse a su host. Por lo tanto, debe prestar mucha atención a sus reglas de grupo de seguridad y firewall. Esto es para garantizar que otros hosts en Internet no puedan abrir ninguna conexión que no desee que se abra.

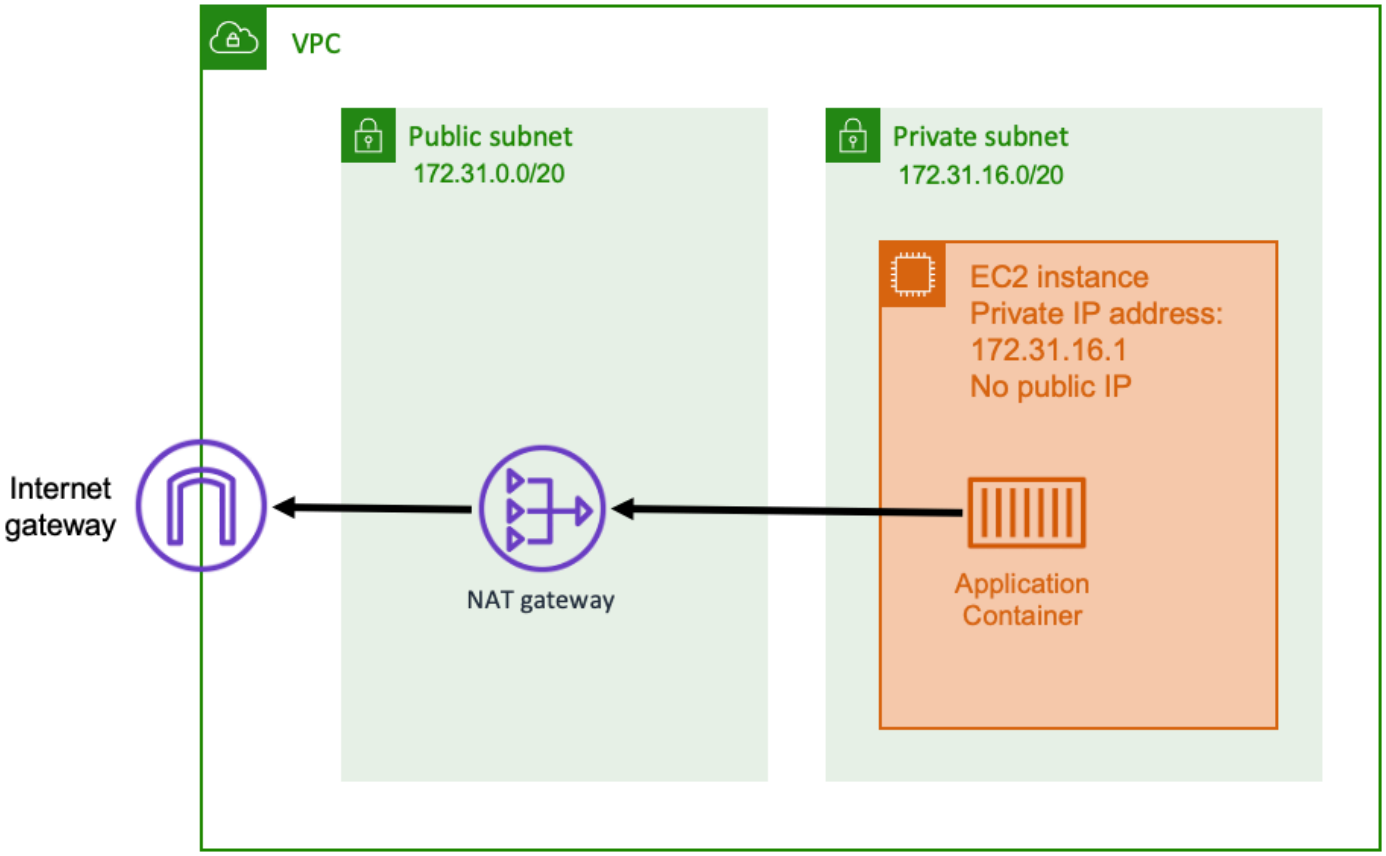
Por ejemplo, si la aplicación se está ejecutando en Amazon EC2, asegúrese de que el puerto 22 para acceso SSH no esté abierto. De lo contrario, su instancia podría recibir constantes intentos de conexión SSH de bots macizos en Internet. Estos bots arrastran a través de direcciones IP públicas. Después de encontrar un puerto SSH abierto, intentan forzar contraseñas brutas para intentar acceder a su instancia. Debido a esto, muchas organizaciones limitan el uso de subredes públicas y prefieren tener la mayoría, si no todos, de sus recursos dentro de subredes privadas.

El uso de subredes públicas para redes es adecuado para aplicaciones públicas que requieren grandes cantidades de ancho de banda o latencia mínima. Los casos de uso aplicables incluyen transmisión de vídeo y servicios de juegos.

Este enfoque de red es compatible tanto cuando utiliza Amazon ECS en Amazon EC2 como cuando lo utiliza enAWS Fargate.

- Uso de Amazon EC2: puede lanzar instancias EC2 en una subred pública. Amazon ECS utiliza estas instancias de EC2 como capacidad de clúster y cualquier contenedor que se ejecute en las instancias puede utilizar la dirección IP pública subyacente del host para redes salientes. Esto se aplica tanto a la`host`y`bridge`modos de red. Sin embargo, el`awsvpc`El modo de red no proporciona ENI de tareas con direcciones IP públicas. Por lo tanto, no pueden hacer uso directo de una puerta de enlace a Internet.
- Uso de Fargate: cuando cree su servicio de Amazon ECS, especifique las subredes públicas para la configuración de red de su servicio y asegúrese de queAsignar una dirección IP públicaestá habilitada. Cada tarea de Fargate está conectada en red en la subred pública y tiene su propia dirección IP pública para la comunicación directa con Internet.

Uso de una subred privada y una puerta de enlace NAT



Mediante el uso de una subred privada y una puerta de enlace NAT, puede ejecutar la aplicación en contenedor en un host que se encuentre en una subred privada. Como tal, este host tiene una dirección IP privada que es enrutable dentro de su VPC, pero no es enrutable desde Internet. Esto significa que otros hosts dentro de la VPC pueden hacer conexiones con el host utilizando su dirección IP privada, pero otros hosts en Internet no pueden realizar ninguna comunicación entrante con el host.

Con una subred privada, puede utilizar una gateway de conversión de las direcciones de red (NAT) para permitir a un host de la subred privada conectarse a Internet. Los hosts de Internet reciben una conexión entrante que parece provenir de la dirección IP pública de la puerta de enlace NAT que está dentro de una subred pública. La puerta de enlace NAT es responsable de servir de puente entre Internet y la VPC privada. Esta configuración suele ser preferida por razones de seguridad, ya que significa que su VPC está protegida contra el acceso directo de los atacantes en Internet. Para obtener más información, consulteGateways NATen laGuía del usuario de Amazon VPC.

Este enfoque de red privada es adecuado para escenarios en los que desea proteger sus contenedores del acceso externo directo. Los escenarios aplicables incluyen sistemas de procesamiento de pagos o contenedores que almacenan datos de usuario y contraseñas. Se le cobrará por la creación y el uso de una gateway NAT en su cuenta. También se aplican las tarifas de procesamiento de datos y uso por horas de la gateway NAT. Para fines de redundancia, debe tener una gateway NAT en cada zona de disponibilidad. De esta forma, la pérdida de disponibilidad de una única zona de disponibilidad no compromete su conectividad saliente. Debido a esto, si tiene una carga de trabajo pequeña, podría ser más rentable usar subredes privadas y puertas de enlace NAT.

Este enfoque de red es compatible tanto cuando se utiliza Amazon ECS en Amazon EC2 como cuando se utiliza enAWS Fargate.

- Uso de Amazon EC2: puede lanzar instancias EC2 en una subred privada. Los contenedores que se ejecutan en estos hosts EC2 utilizan la red de hosts subyacentes y las solicitudes salientes pasan a través de la puerta de enlace NAT.
- Uso de Fargate: cuando cree su servicio de Amazon ECS, especifique subredes privadas para la configuración de red de su servicio y no habilite la opciónAsignar una dirección IP públicaOpción de. Cada tarea de Fargate está alojada en una subred privada. Su tráfico saliente se enruta a través de cualquier puerta de enlace NAT que haya asociado a esa subred privada.

Recibir conexiones entrantes de Internet

Si ejecuta un servicio público, debe aceptar el tráfico entrante de Internet. Por ejemplo, su sitio web público debe aceptar solicitudes HTTP entrantes de los navegadores. En tal caso, otros hosts en Internet también deben iniciar una conexión entrante con el host de su aplicación.

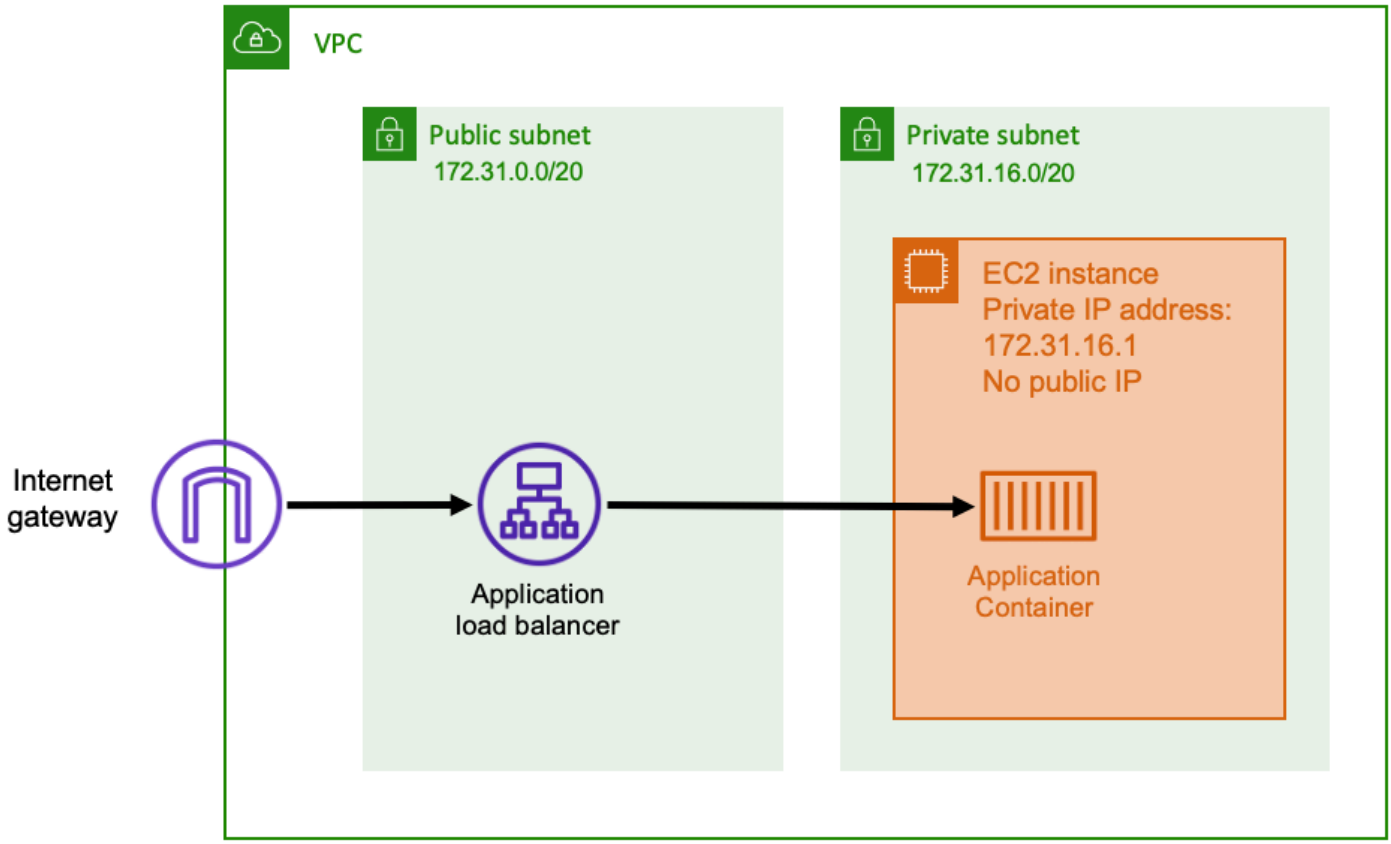
Un enfoque para este problema es lanzar los contenedores en hosts que se encuentran en una subred pública con una dirección IP pública. Sin embargo, no es recomendable para aplicaciones a gran escala. Para estos, un mejor enfoque es tener una capa de entrada escalable que se encuentre entre Internet y su aplicación. Para este método, puede utilizar cualquiera de lasAWSenumerados en esta sección como una entrada.

Temas

- Balanceador de carga de aplicaciones
- Balanceador de carga de red
- API HTTP de Amazon API Gateway

Balanceador de carga de aplicaciones

Un Application Load Balancer funciona en la capa de aplicación. Es la séptima capa del modelo de interconexión de sistemas abiertos (OSI). Esto hace que un Application Load Balancer sea adecuado para servicios HTTP públicos. Si tiene un sitio web o una API HTTP REST, entonces un equilibrador de carga de aplicaciones es un equilibrador de carga adecuado para esta carga de trabajo. Para obtener más información, consulte¿Qué es un Application Load Balancer?en laGuía del usuario para Application Load Balancers.



Con esta arquitectura, se crea un Application Load Balancer en una subred pública para que tenga una dirección IP pública y pueda recibir conexiones entrantes desde Internet. Cuando el Application Load Balancer recibe una conexión entrante, o más específicamente una solicitud HTTP, abre una conexión a la aplicación utilizando su dirección IP privada. Luego, reenvía la solicitud a través de la conexión interna.

Un Application Load Balancer tiene las siguientes ventajas.

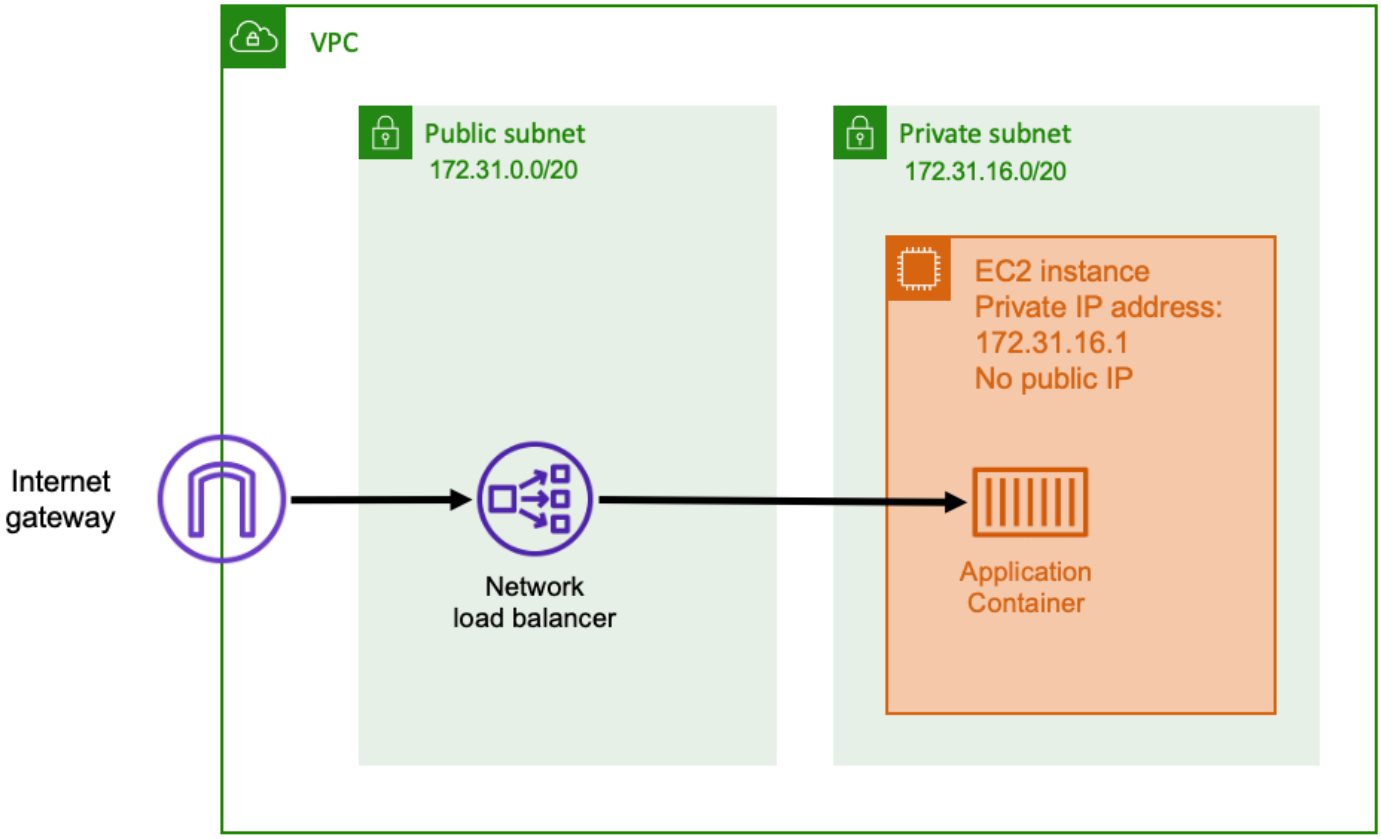
- Terminación SSL/TLS: un Application Load Balancer puede mantener comunicaciones HTTPS seguras y certificados para las comunicaciones con clientes. Opcionalmente, puede terminar la

conexión SSL en el nivel del equilibrador de carga para que no tenga que manejar certificados en su propia aplicación.

- Enrutamiento avanzado: un Application Load Balancer puede tener varios nombres de host DNS. También cuenta con capacidades avanzadas de enrutamiento para enviar solicitudes HTTP entrantes a diferentes destinos en función de métricas como el nombre de host o la ruta de la solicitud. Esto significa que puede usar un único Application Load Balancer como entrada para muchos servicios internos diferentes, o incluso microservicios en diferentes rutas de una API REST.
- Soporte de gRPC y websockets: un Application Load Balancer puede manejar algo más que HTTP. También puede equilibrar la carga gRPC y servicios basados en websocket, con soporte HTTP/2.
- Seguridad: un Application Load Balancer ayuda a proteger la aplicación del tráfico malintencionado. Incluye características como mitigaciones de sincronización HTTP y está integrado con AWS Web Application Firewall (AWS WAF).AWS WAFpuede filtrar aún más el tráfico malicioso que podría contener patrones de ataque, como inyección SQL o secuencias de comandos entre sitios.

Balanceador de carga de red

Un Network Load Balancer actúa como la cuarta capa del modelo de interconexión de sistemas abiertos (OSI). Es adecuado para protocolos o escenarios que no son HTTP donde es necesario el cifrado de extremo a extremo, pero no tiene las mismas características específicas de HTTP que un equilibrador de carga de aplicaciones. Por lo tanto, un Network Load Balancer es el más adecuado para aplicaciones que no utilizan HTTP. Para obtener más información, consulte¿Qué es un Network Load Balancer?en laGuía del usuario para Network Load Balancers.



Cuando se utiliza un Network Load Balancer como entrada, funciona de forma similar a un Application Load Balancer. Esto se debe a que se crea en una subred pública y tiene una dirección IP pública a la que se puede acceder en Internet. A continuación, el Network Load Balancer abre una conexión a la dirección IP privada del host que ejecuta el contenedor y envía los paquetes desde el lado público al lado privado.

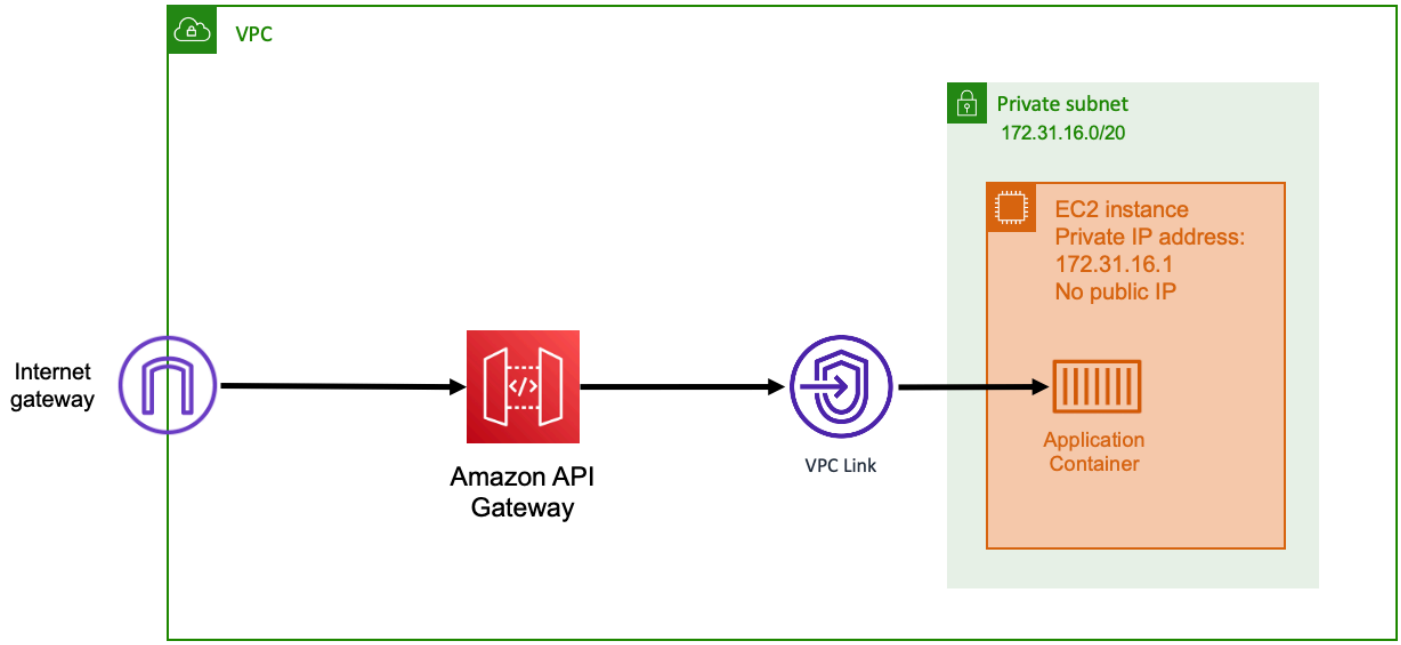
Dado que el Network Load Balancer funciona en un nivel inferior de la pila de redes, no tiene el mismo conjunto de características que el Application Load Balancer. Sin embargo, tiene las siguientes características importantes.

- Cifrado de extremo a extremo: dado que un equilibrador de carga de red funciona en la cuarta capa del modelo OSI, no lee el contenido de los paquetes. De ese modo resulta adecuado para comunicaciones de balanceamiento de carga que necesitan un cifrado integral.
- Cifrado TLS: además del cifrado de extremo a extremo, Network Load Balancer también puede terminar las conexiones TLS. De esta manera, sus aplicaciones back-end no tienen que implementar su propio TLS.

- Compatibilidad con UDP: dado que un Network Load Balancer funciona en la cuarta capa del modelo OSI, es adecuado para cargas de trabajo y protocolos no HTTP distintos de TCP.

API HTTP de Amazon API Gateway

La API HTTP de Amazon API Gateway es un servidor menos entrada que es adecuado para aplicaciones HTTP con ráfagas repentinas en volúmenes de solicitudes o volúmenes de solicitudes bajos. Para obtener más información, consulte¿Qué es Amazon API Gateway?en laGuía para desarrolladores de API Gateway.



El modelo de precios para el Application Load Balancer y el Network Load Balancer incluye un precio por hora para mantener los equilibradores de carga disponibles para aceptar conexiones entrantes en todo momento. Por el contrario, API Gateway cobra por cada solicitud por separado. Esto tiene el efecto de que, si no hay solicitudes, no hay cargos. Bajo cargas de tráfico elevadas, un Application Load Balancer o un Network Load Balancer pueden manejar un mayor volumen de solicitudes a un precio por solicitud más barato que API Gateway. Sin embargo, si tiene un número reducido de solicitudes en general o tiene períodos de tráfico bajo, el precio acumulado por usar API Gateway debería ser más rentable que pagar un cargo por hora para mantener un equilibrador de carga que está siendo infrautilizado.

API Gateway funciona mediante un enlace VPC que permite elAWSpara conectarse a hosts dentro de la subred privada de su VPC, utilizando su dirección IP privada. Puede detectar estas direcciones

IP privadas mirandoAWS Cloud Mapregistros de detección de servicios gestionados por la detección de servicios de Amazon ECS.

API Gateway admite las siguientes características.

- Terminación SSL/TLS
- Enrutamiento de diferentes rutas HTTP a diferentes microservicios back-end

Además de las características anteriores, API Gateway también admite el uso de autorizadores Lambda personalizados que puede utilizar para proteger su API contra el uso no autorizado. Para obtener más información, consulteNotas de campo: API basadas en contenedor sin servidor con Amazon ECS y Amazon API Gateway.

Elección de un modo de red

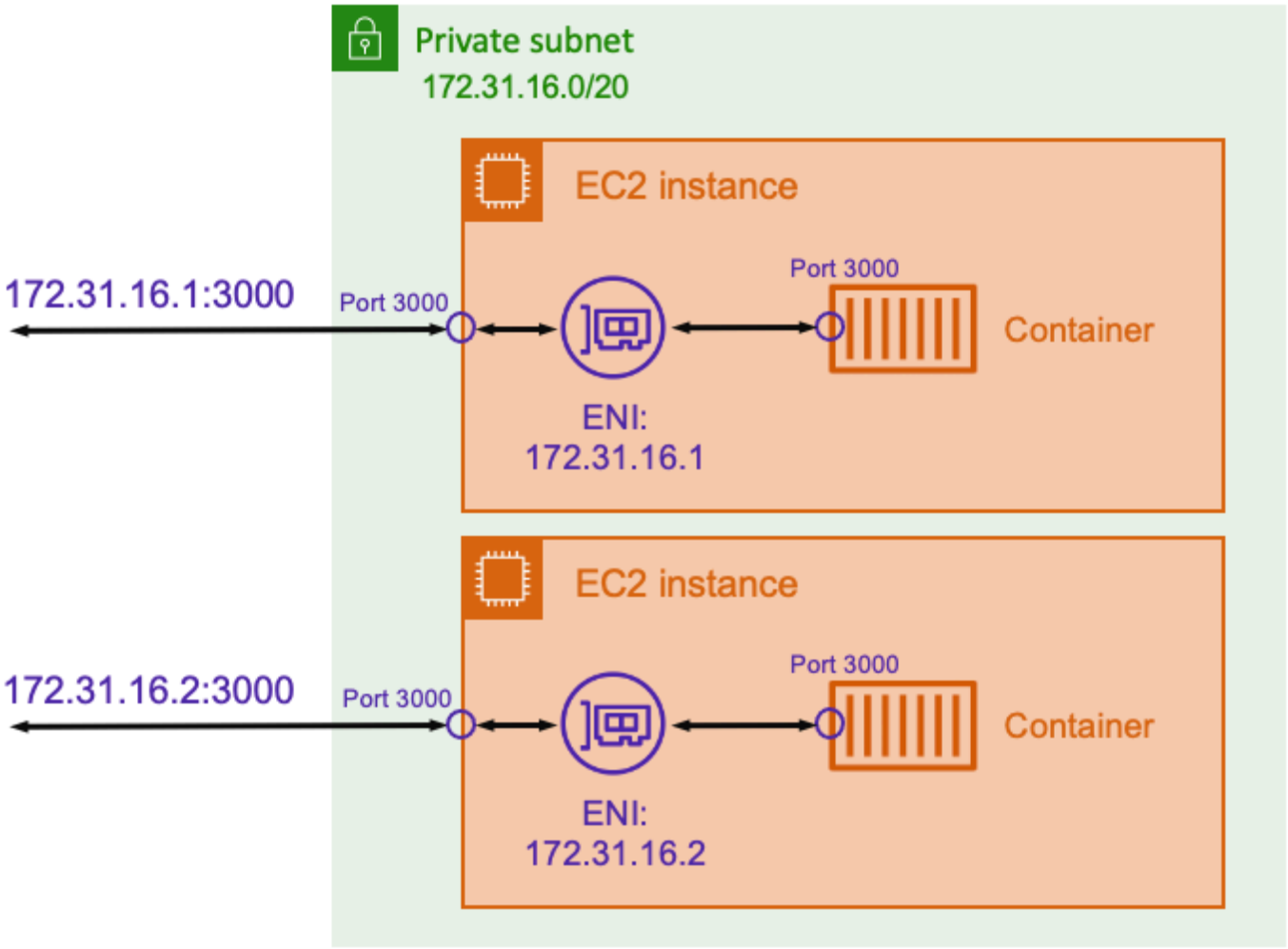
Los enfoques mencionados anteriormente para diseñar conexiones de red entrantes y salientes se pueden aplicar a cualquiera de sus cargas de trabajo enAWS, incluso si no están dentro de un contenedor. Cuando se ejecutan contenedores enAWS, debe considerar otro nivel de red. Una de las principales ventajas de usar contenedores es que puede empaquetar varios contenedores en un único host. Al hacer esto, debe elegir cómo desea conectar en red los contenedores que se ejecutan en el mismo host. Las siguientes son las opciones para elegir.

Temas

- Modo de host
- Modo de puente
- Modo AWSVSVSVPC

Modo de host

La`host`es el modo de red más básico compatible con Amazon ECS. Usando el modo host, la red del contenedor está vinculada directamente al host subyacente que ejecuta el contenedor.



Supongamos que está ejecutando un contenedor Node.js con una aplicación Express que escucha en el puerto`3000`De forma similar a la ilustrada en el diagrama anterior. Cuando se utiliza la`host`, el contenedor recibe tráfico en el puerto 3000 utilizando la dirección IP de la instancia de Amazon EC2 del host subyacente. No recomendamos el uso de este modo.

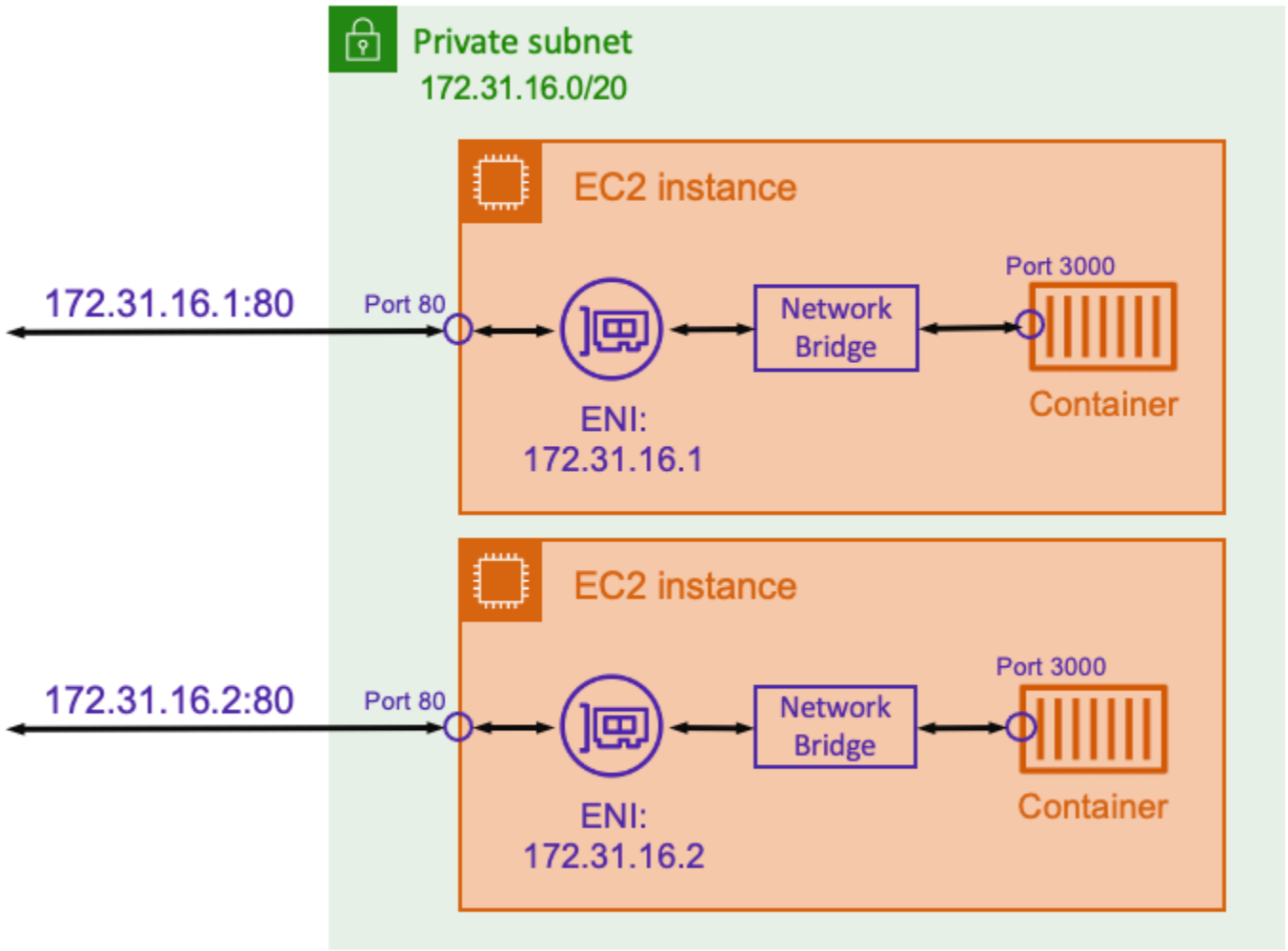
Hay inconvenientes importantes en el uso de este modo de red. No puede ejecutar más de una sola instancia de una tarea en cada host. Esto se debe a que solo la primera tarea puede enlazar a su puerto requerido en la instancia de Amazon EC2. Tampoco hay forma de reasignar un puerto contenedor cuando está usando`host`en el modo de red. Por ejemplo, si una aplicación necesita escuchar un número de puerto determinado, no puede volver a asignar el número de puerto directamente. En su lugar, debe administrar cualquier conflicto de puertos cambiando la configuración de la aplicación.

También hay implicaciones para la seguridad cuando se utiliza el`host`en el modo de red. Este modo permite a los contenedores suplantar el host y permite que los contenedores se conecten a los servicios de red de bucle invertido privados en el host.

La`host`solo se admite en tareas de Amazon ECS alojadas en instancias de Amazon EC2. No se admite cuando se utiliza Amazon ECS en Fargate.

Modo de puente

con`bridge`, está utilizando un puente de red virtual para crear una capa entre el host y la red del contenedor. De esta forma, puede crear asignaciones de puertos que reasignen un puerto host a un puerto contenedor. Las asignaciones pueden ser estáticas o dinámicas.

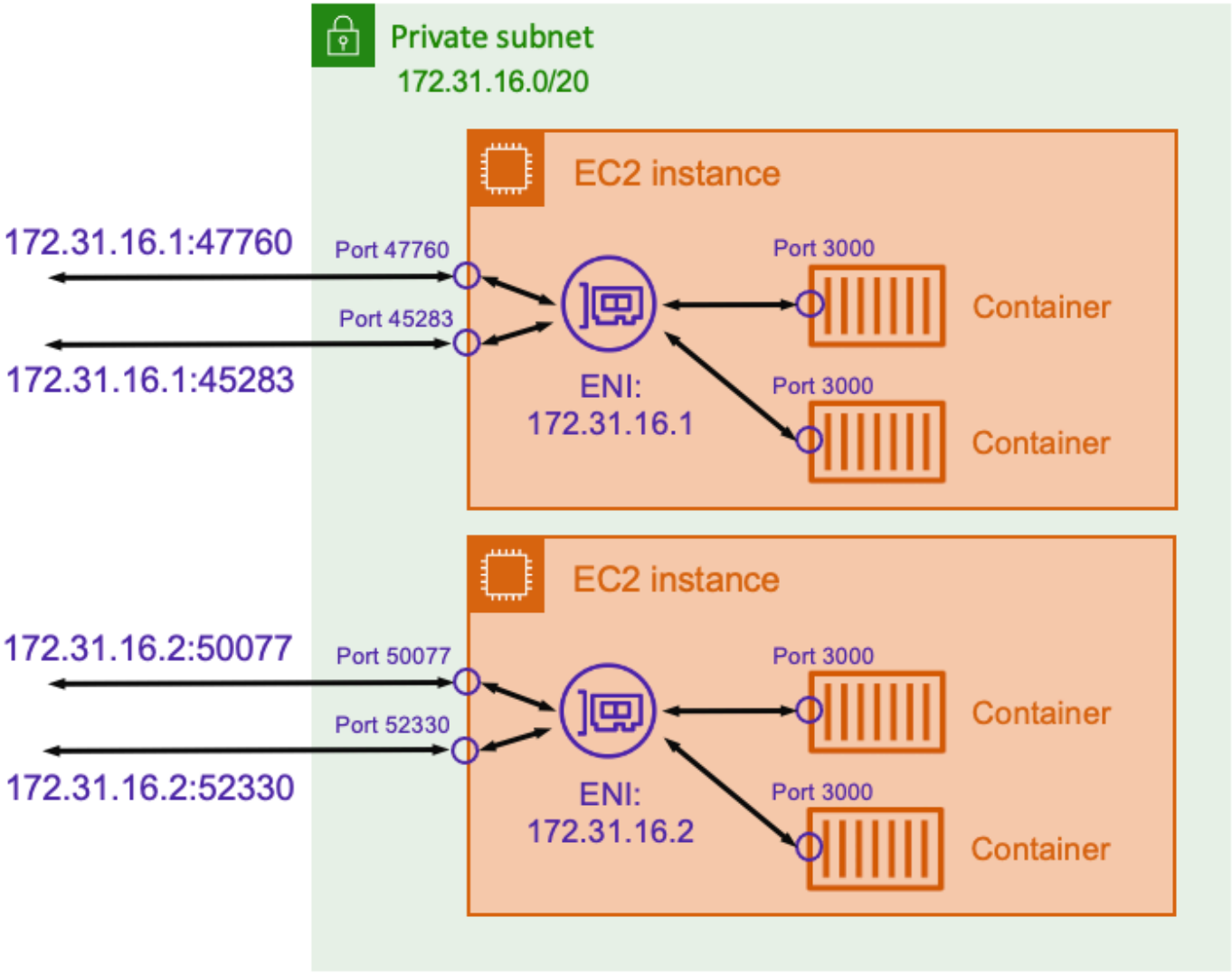


Con una asignación de puertos estáticos, puede definir explícitamente qué puerto host desea asignar a un puerto contenedor. Usando el ejemplo anterior, puerto`80`en el host se está asignando

al puerto`3000`en el contenedor. Para comunicarse con la aplicación contenedorizada, envía tráfico al puerto`80`En la dirección IP de la instancia de Amazon EC2. Desde la perspectiva de la aplicación en contenedores, ve que el tráfico entrante en el puerto`3000`.

Si solo desea cambiar el puerto de tráfico, entonces las asignaciones de puertos estáticos son adecuadas. Sin embargo, esto todavía tiene la misma desventaja que usar el`host`en el modo de red. No puede ejecutar más de una sola instancia de una tarea en cada host. Esto se debe a que una asignación de puertos estáticos sólo permite asignar un único contenedor al puerto 80.

Para solucionar este problema, considere la posibilidad de utilizar el`bridge`Con una asignación de puertos dinámica, tal y como se muestra en el siguiente diagrama.



Al no especificar un puerto host en la asignación de puertos, puede hacer que Docker elija un puerto aleatorio no utilizado del rango de puertos efímero y lo asigne como puerto de host público para el contenedor. Por ejemplo, la aplicación Node.js que escucha en el puerto`3000`en el contenedor se le puede asignar un puerto de número alto aleatorio, como`47760`en el host de Amazon EC2. Hacer esto significa que puede ejecutar varias copias de ese contenedor en el host. Además, a cada contenedor se le puede asignar su propio puerto en el host. Cada copia del contenedor recibe tráfico en el puerto`3000`. Sin embargo, los clientes que envían tráfico a estos contenedores utilizan los puertos de host asignados aleatoriamente.

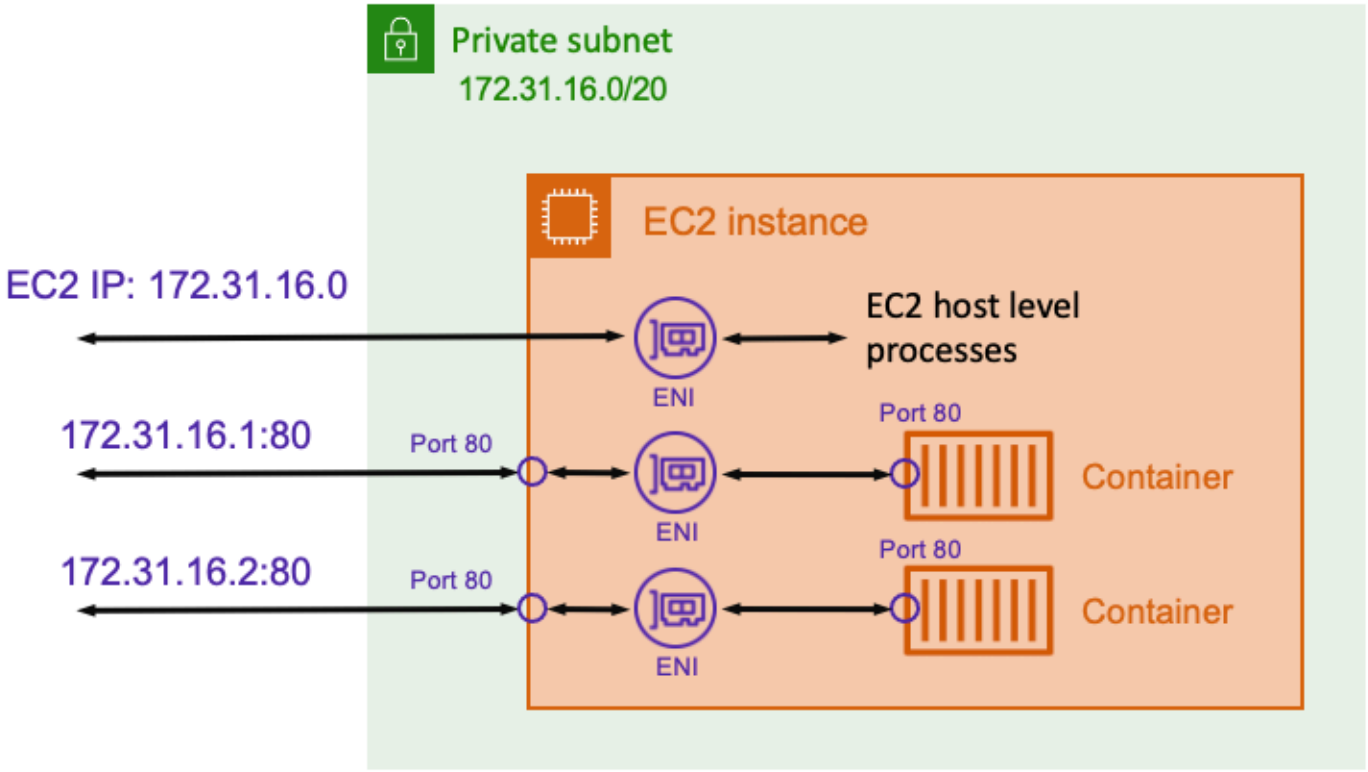
Amazon ECS le ayuda a realizar un seguimiento de los puertos asignados aleatoriamente para cada tarea. Lo hace actualizando automáticamente los grupos de destino del equilibrador de carga yAWS Cloud Mappara tener la lista de direcciones IP y puertos de tareas. De ese modo resulta más sencillo utilizar los servicios que operan con`bridge`con puertos dinámicos.

Sin embargo, una desventaja de usar el`bridge`es que es difícil bloquear las comunicaciones de servicio a servicio. Debido a que los servicios pueden asignarse a cualquier puerto aleatorio no utilizado, es necesario abrir amplios rangos de puertos entre hosts. Sin embargo, no es fácil crear reglas específicas para que un servicio determinado solo pueda comunicarse con otro servicio específico. Los servicios no tienen puertos específicos para usar para reglas de red de grupos de seguridad.

La`bridge`solo se admite en tareas de Amazon ECS alojadas en instancias de Amazon EC2. No se admite cuando se utiliza Amazon ECS en Fargate.

Modo AWSVSVSVPC

Con el`awsvpc`, Amazon ECS crea y administra una interfaz de red elástica (ENI) para cada tarea y cada tarea recibe su propia dirección IP privada dentro de la VPC. Este ENI es independiente de los hosts subyacentes ENI. Si una instancia de Amazon EC2 ejecuta varias tareas, el ENI de cada tarea también es independiente.



En el ejemplo anterior, la instancia de Amazon EC2 se asigna a un ENI. El ENI representa la dirección IP de la instancia EC2 utilizada para las comunicaciones de red en el nivel de host. Cada tarea también tiene una ENI correspondiente y una dirección IP privada. Debido a que cada ENI es independiente, cada contenedor puede vincularse al puerto`80`en la tarea ENI. Por lo tanto, no es necesario realizar un seguimiento de los números de puerto. En su lugar, puede enviar tráfico al puerto`80`En la dirección IP de la ENI.

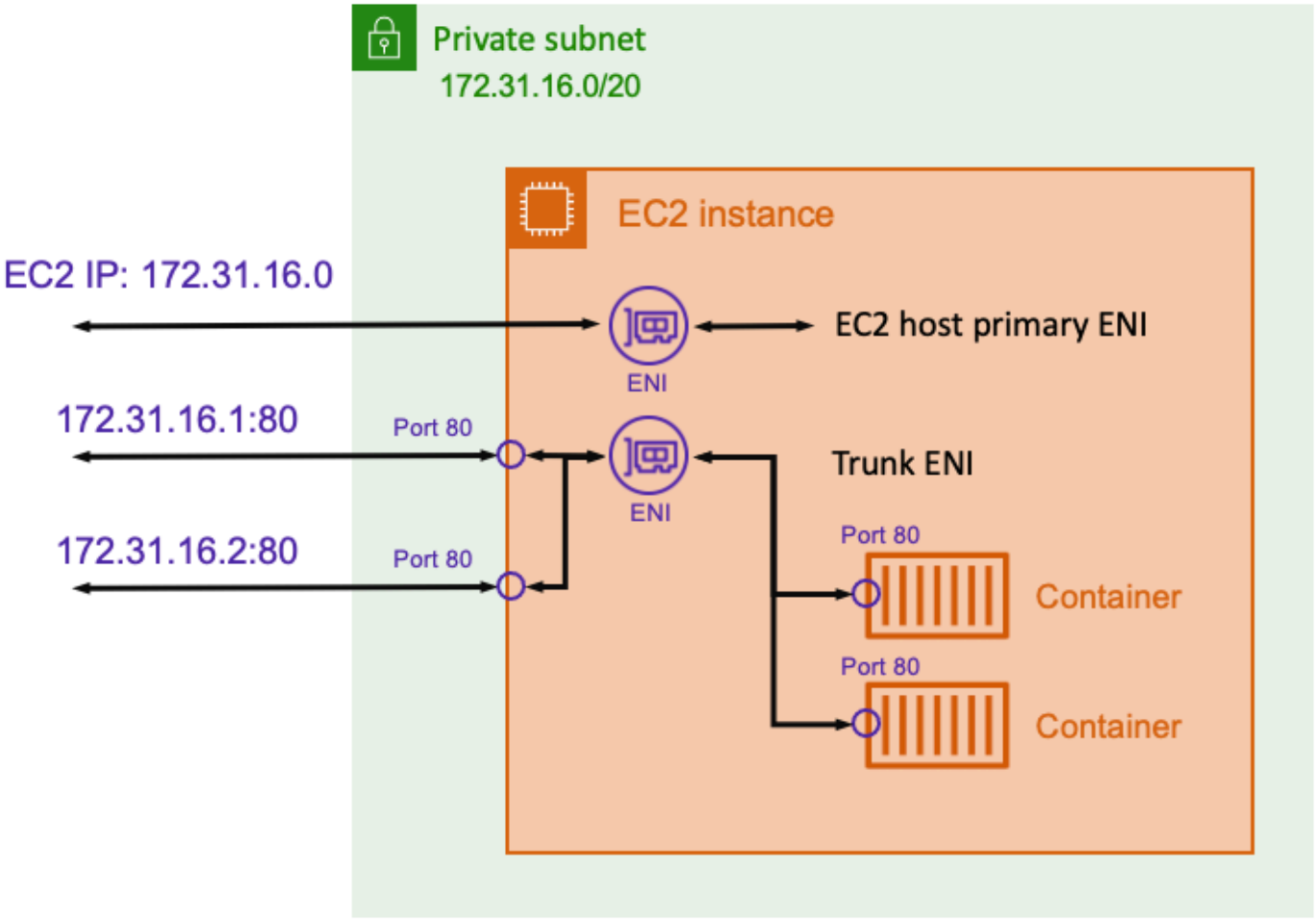
La ventaja de usar el`awsvpc`es que cada tarea tiene un grupo de seguridad separado para permitir o denegar el tráfico. Esto significa que tiene mayor flexibilidad para controlar las comunicaciones entre tareas y servicios a un nivel más detallado. También puede configurar una tarea para denegar el tráfico entrante de otra tarea ubicada en el mismo host.

La`awsvpc`es compatible con las tareas de Amazon ECS alojadas tanto en Amazon EC2 como en Fargate. Tenga en cuenta que, al usar Fargate, el`awsvpc`es necesario el modo de red.

Cuando se utiliza la`awsvpc`hay algunos desafíos que debe tener en cuenta.

Aumento de la densidad de tareas con ENI Trunking

La mayor desventaja de usar el`awsvpc`Con las tareas alojadas en las instancias de Amazon EC2, las instancias EC2 presentan un límite en cuanto al número de ENI que pueden tener asociadas. Esto limita el número de tareas que puede colocar en cada instancia. Amazon ECS proporciona la función de enlace troncal ENI que aumenta el número de ENI disponibles para lograr una mayor densidad de tareas.



Cuando se utiliza el enlace troncal ENI, se utilizan dos adjuntos ENI de forma predeterminada. El primero es el ENI principal de la instancia, que se utiliza para cualquier proceso de nivel de host. El segundo es el ENI troncal, que Amazon ECS crea. Esta característica solo se admite en tipos de instancias de Amazon EC2.

Considere este ejemplo. Sin el enlace troncal ENI, un`c5.large`que tiene dos vCPUs sólo puede alojar dos tareas. Sin embargo, con el enlace troncal ENI, un`c5.large`que tiene dos vCPU puede alojar hasta diez tareas. Cada tarea tiene una dirección IP y un grupo de seguridad diferentes. Para

obtener más información acerca de los tipos de instancias disponibles y su densidad, consulteTipos de instancias de Amazon EC2 admitidosen laGuía de Amazon Elastic Container Service.

El enlace troncal ENI no tiene ningún impacto en el rendimiento del tiempo de ejecución en términos de latencia o ancho de banda. Sin embargo, aumenta el tiempo de inicio de la tarea. Debe asegurarse de que, si se utiliza el enlace troncal ENI, las reglas de escalado automático y otras cargas de trabajo que dependen del tiempo de inicio de la tarea siguen funcionando como espera.

Para obtener más información, consulteTrunking de interfaz de red elásticaen laGuía de Amazon Elastic Container Service.

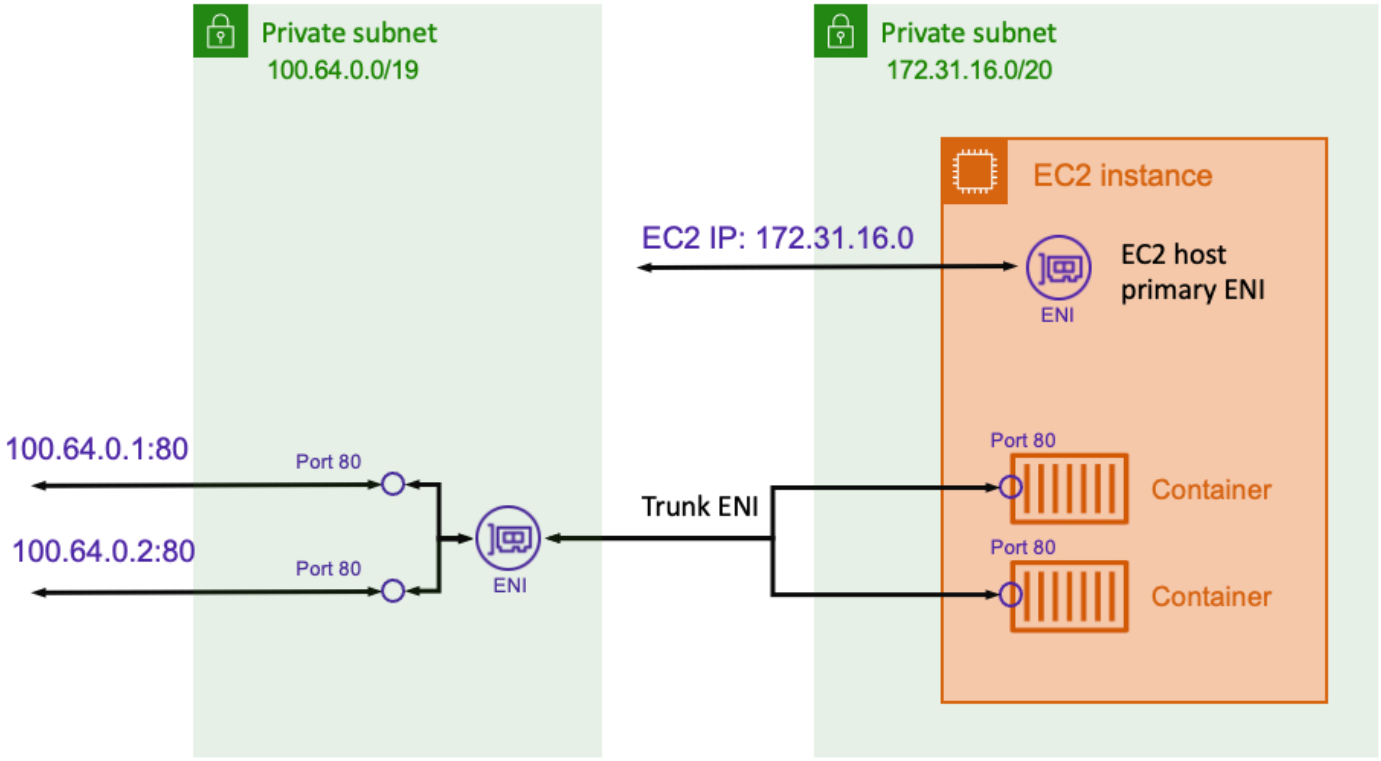
Prevención del agotamiento de direcciones IP

Al asignar una dirección IP independiente a cada tarea, puede simplificar su infraestructura general y mantener grupos de seguridad que proporcionan un gran nivel de seguridad. Sin embargo, esta configuración puede provocar el agotamiento de IP.

La VPC predeterminada en suAWScuenta con subredes preaprovisionadas que tienen un`/20`Gama de CIDR. Esto significa que cada subred tiene 4.091 direcciones IP disponibles. Tenga en cuenta que varias direcciones IP dentro del`/20`están reservados para uso específico de AWS. Considere este ejemplo. Las aplicaciones se distribuyen en tres subredes en tres zonas de disponibilidad para lograr una alta disponibilidad. En este caso, puede utilizar aproximadamente 12.000 direcciones IP en las tres subredes.

Mediante la conexión troncal ENI, cada instancia de Amazon EC2 que inicie requiere dos direcciones IP. Una dirección IP se utiliza para la ENI principal y la otra dirección IP para la ENI troncal. Cada tarea de Amazon ECS de la instancia requiere una dirección IP. Si está iniciando una carga de trabajo extremadamente grande, podría quedarse sin direcciones IP disponibles. Esto puede dar lugar a errores de inicio de Amazon EC2 o errores en el inicio de tareas. Estos errores se producen porque los ENI no pueden agregar direcciones IP dentro de la VPC si no hay direcciones IP disponibles.

Cuando se utiliza la`awsvpc`, debe evaluar sus requisitos de dirección IP y asegurarse de que los rangos CIDR de subred satisfagan sus necesidades. Si ya ha comenzado a utilizar una VPC que tiene subredes pequeñas y comienza a quedarse sin espacio de direcciones, puede agregar una subred secundaria.



Mediante el enlace troncal ENI, el CNI de Amazon VPC se puede configurar para utilizar ENI en un espacio de direcciones IP diferente al del host. De este modo, puede proporcionar a su host de Amazon EC2 y a sus tareas distintos rangos de direcciones IP que no se superponen. En el diagrama de ejemplo, la dirección IP del host EC2 se encuentra en una subred que tiene la propiedad`172.31.16.0/20`Rango de IP. Sin embargo, las tareas que se ejecutan en el host se asignan direcciones IP en el`100.64.0.0/19`Rango de. Al usar dos rangos de IP independientes, no necesita preocuparse por las tareas que consumen demasiadas direcciones IP y no dejan suficientes direcciones IP para las instancias.

Uso del modo de pila dual IPv6

La`awsvpc`es compatible con VPC configuradas para el modo de pila dual IPv6. Una VPC que utiliza el modo de pila doble puede comunicarse mediante IPv4, IPv6 o ambos. Cada subred de la VPC puede tener un intervalo CIDR IPv4 y un intervalo CIDR IPv6. Para obtener más información, consulteDirecciones IP en su VPCen laGuía del usuario de Amazon VPC.

No puede deshabilitar la compatibilidad con IPv4 para su VPC y sus subredes para solucionar problemas de agotamiento de IPv4. Sin embargo, con la compatibilidad con IPv6, puede utilizar algunas capacidades nuevas, específicamente la gateway de Internet de solo salida. Una gateway

de Internet de solo salida permite a las tareas utilizar su dirección IPv6 de enrutamiento público para iniciar las conexiones salientes a Internet. Pero la gateway de Internet de solo salida no permite las conexiones de Internet. Para obtener más información, consulteGateways de Internet de solo salidaen laGuía del usuario de Amazon VPC.

Conexión aAWSdesde el interior de su VPC

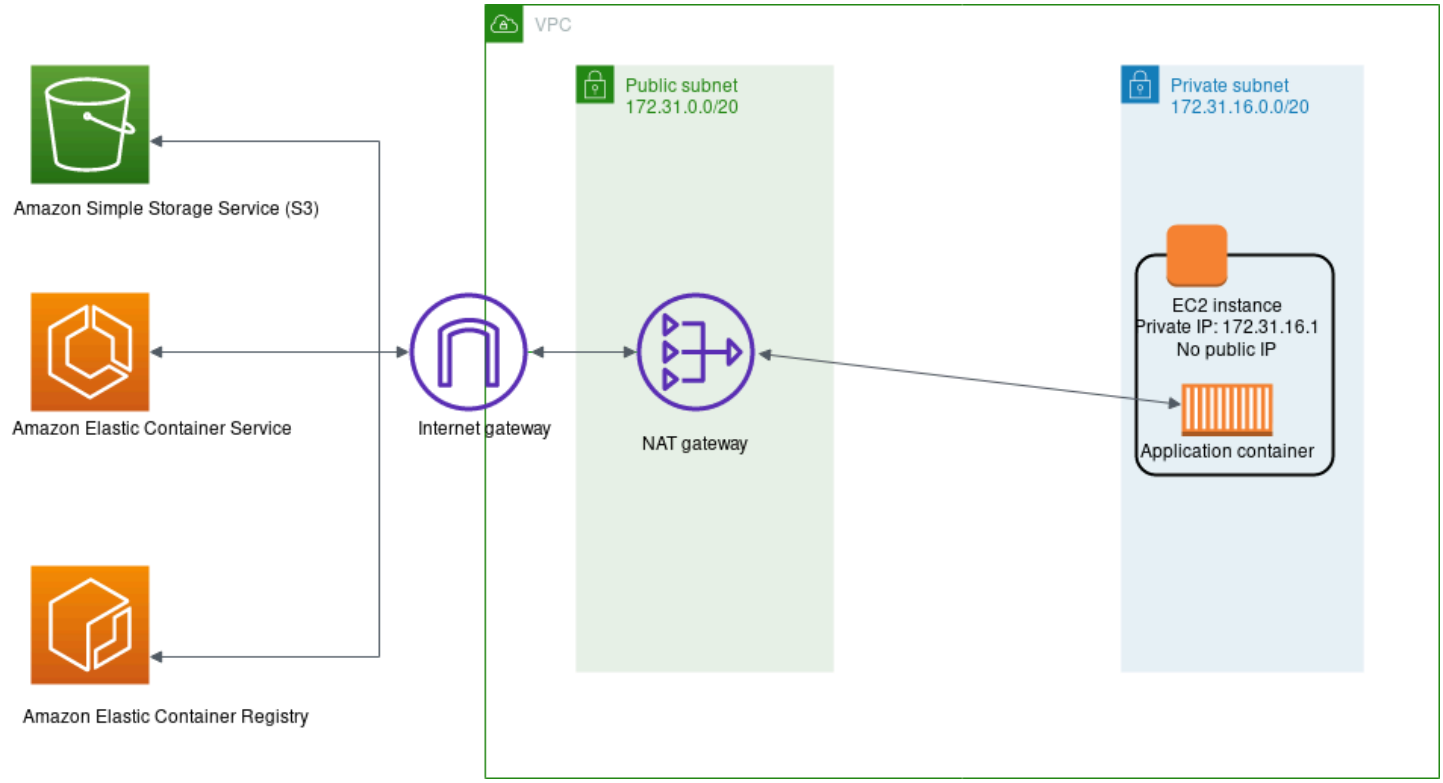
Para que Amazon ECS funcione correctamente, el agente contenedor de ECS que se ejecuta en cada host debe comunicarse con el plano de control de Amazon ECS. Si almacena las imágenes de contenedor en Amazon ECR, los hosts de Amazon EC2 deben comunicarse con el extremo del servicio Amazon ECR y con Amazon S3, donde se almacenan las capas de imágenes. Si usa otrosAWSpara su aplicación contenedorizada, como los datos persistentes almacenados en DynamoDB, compruebe que estos servicios también cuentan con el soporte de red necesario.

Temas

- Gateway NAT
- AWS PrivateLink

Gateway NAT

El uso de una puerta de enlace NAT es la forma más sencilla de garantizar que sus tareas de Amazon ECS puedan acceder a otrosAWSServicios de . Para obtener más información acerca de este enfoque, consulteUso de una subred privada y una puerta de enlace NAT.



Las siguientes son las desventajas de utilizar este enfoque:

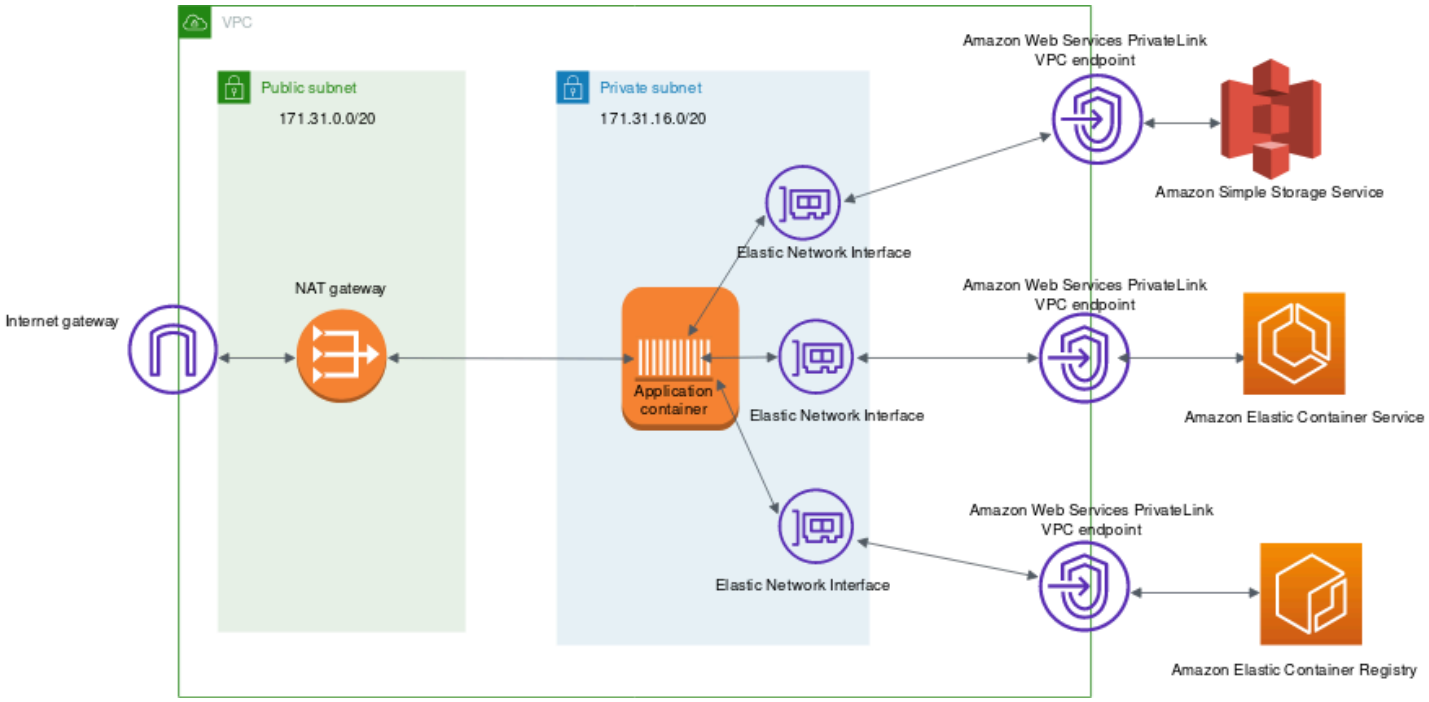
- No puede limitar con qué destinos puede comunicarse la puerta de enlace NAT. Tampoco puede limitar a qué destinos se puede comunicar su neumático back-end sin interrumpir todas las comunicaciones salientes desde su VPC.
- Las puertas de enlace NAT se cargan por cada GB de datos que pasan. Si utiliza la puerta de enlace NAT para descargar archivos de gran tamaño desde Amazon S3 o para realizar un gran volumen de consultas de base de datos a DynamoDB, se le cobrará por cada GB de ancho de banda. Además, las gateways NAT admiten 5 Gbps de ancho de banda y se amplían automáticamente hasta 45 Gbps. Si se dirige a través de una única puerta de enlace NAT, las aplicaciones que requieren conexiones de ancho de banda muy alto pueden encontrar restricciones de red. Como solución alternativa, puede dividir la carga de trabajo entre varias subredes y asignar a cada subred su propia puerta de enlace NAT.

AWS PrivateLink

AWS PrivateLinkproporciona conectividad privada entre VPC,AWSy las redes locales sin exponer el tráfico a Internet público.

Una de las tecnologías utilizadas para lograr esto es el punto final de VPC. Un punto de enlace de la VPC permite conexiones privadas entre la VPC y laAWSy servicios de punto de enlace de la VPC. El tráfico entre su VPC y el servicio no sale de la red de Amazon. Un punto de enlace de la VPC no requiere una gateway de Internet, una gateway privada virtual, un dispositivo NAT, una conexión de VPN niAWS Direct Connectconexión de. Las instancias de Amazon EC2 de su VPC no necesitan direcciones IP públicas para comunicarse con los recursos del servicio.

En el siguiente diagrama, se muestra cómoAWSfuncionan cuando se utilizan puntos finales de VPC en lugar de una puerta de enlace a Internet.AWS PrivateLinkprovisiones interfaces elásticas de red (ENI) dentro de la subred, y las reglas de enrutamiento de VPC se utilizan para enviar cualquier comunicación al nombre de host del servicio a través de ENI, directamente al destinoAWSServicio de. Este tráfico ya no necesita usar la puerta de enlace NAT o la puerta de enlace de Internet.



A continuación se indican algunos de los extremos comunes de VPC que se utilizan con el servicio Amazon ECS.

- Punto de enlace de la VPC de la gateway S3
- Punto de enlace de DynamoDB VPC
- Punto de conexión de Amazon ECS VPC
- Punto de conexión de Amazon ECR VPC

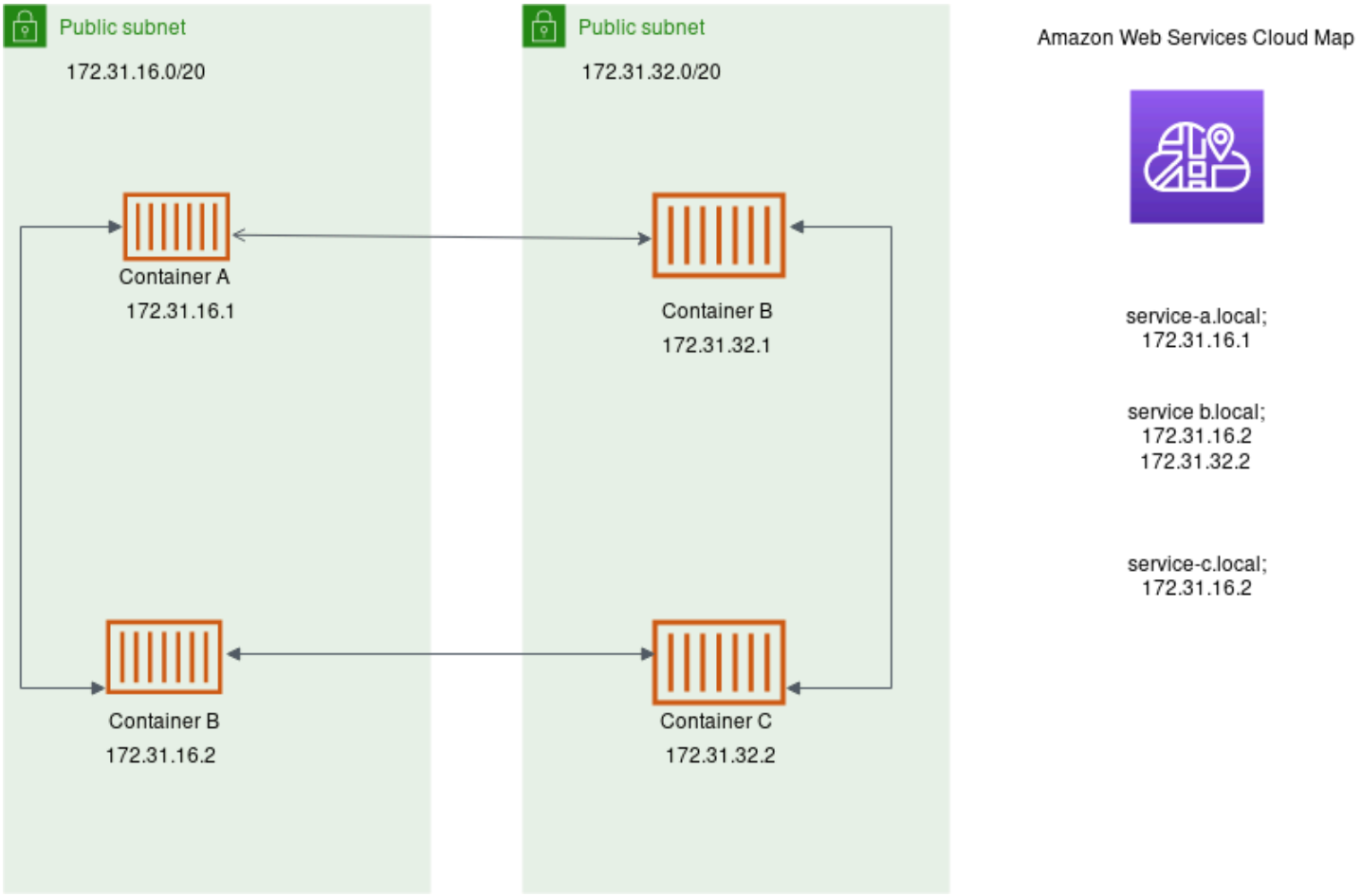
Muchos otrosAWSadmiten los puntos de enlace de la VPC. Si hace un uso intensivo de cualquierAWS, debe buscar la documentación específica para ese servicio y cómo crear un extremo de VPC para ese tráfico.

Redes entre servicios de Amazon ECS en una VPC

Con los contenedores de Amazon ECS en una VPC, puede derramar aplicaciones monolíticas en partes separadas que se pueden implementar y escalar de forma independiente en un entorno seguro. Sin embargo, puede ser difícil asegurarse de que todas estas partes, tanto dentro como fuera de una VPC, puedan comunicarse entre sí. Existen varios enfoques para facilitar la comunicación, todos con diferentes ventajas y desventajas.

Uso de detección de servicios

Un enfoque para la comunicación de servicio a servicio es la comunicación directa mediante el descubrimiento de servicios. En este método, puede utilizar elAWS Cloud Mapintegración de detección de servicios con Amazon ECS. Mediante la detección de servicios, Amazon ECS sincroniza la lista de tareas iniciadas conAWS Cloud Map, que mantiene un nombre de host DNS que resuelve las direcciones IP internas de una o más tareas de ese servicio en particular. Otros servicios de Amazon VPC pueden utilizar este nombre de host DNS para enviar tráfico directamente a otro contenedor mediante su dirección IP interna. Para obtener más información, consulteDetección de serviciosen laGuía de Amazon Elastic Container Service.



En el diagrama anterior, hay tres servicios.`service`Atiene un contenedor y se comunica con`service`B, que tiene dos contenedores.`service`Btambién debe comunicarse con`service`C, que tiene un contenedor. Cada contenedor de los tres servicios puede usar los nombres DNS internos deAWS Cloud Mappara encontrar las direcciones IP internas de un contenedor desde el servicio descendente al que necesita comunicarse.

Este enfoque de la comunicación de servicio a servicio proporciona baja latencia. A primera vista, también es simple ya que no hay componentes adicionales entre los contenedores. El tráfico viaja directamente de un contenedor a otro contenedor.

Este enfoque es adecuado cuando se utiliza el`awsvpc`, donde cada tarea tiene su propia dirección IP única. La mayoría del software solo admite el uso de DNSA, que se resuelven directamente en direcciones IP. Cuando se utiliza la`awsvpc`, la dirección IP de cada tarea es unARegistro de. Sin embargo, si utiliza`bridge`, varios contenedores podrían estar compartiendo la misma dirección IP. Además, las asignaciones de puertos dinámicos hacen que los contenedores se asignen aleatoriamente números de puerto en esa única dirección IP. En este punto, unAya no es suficiente para la detección de servicios. También debe usar unSRVRegistro de. Este tipo de registro puede

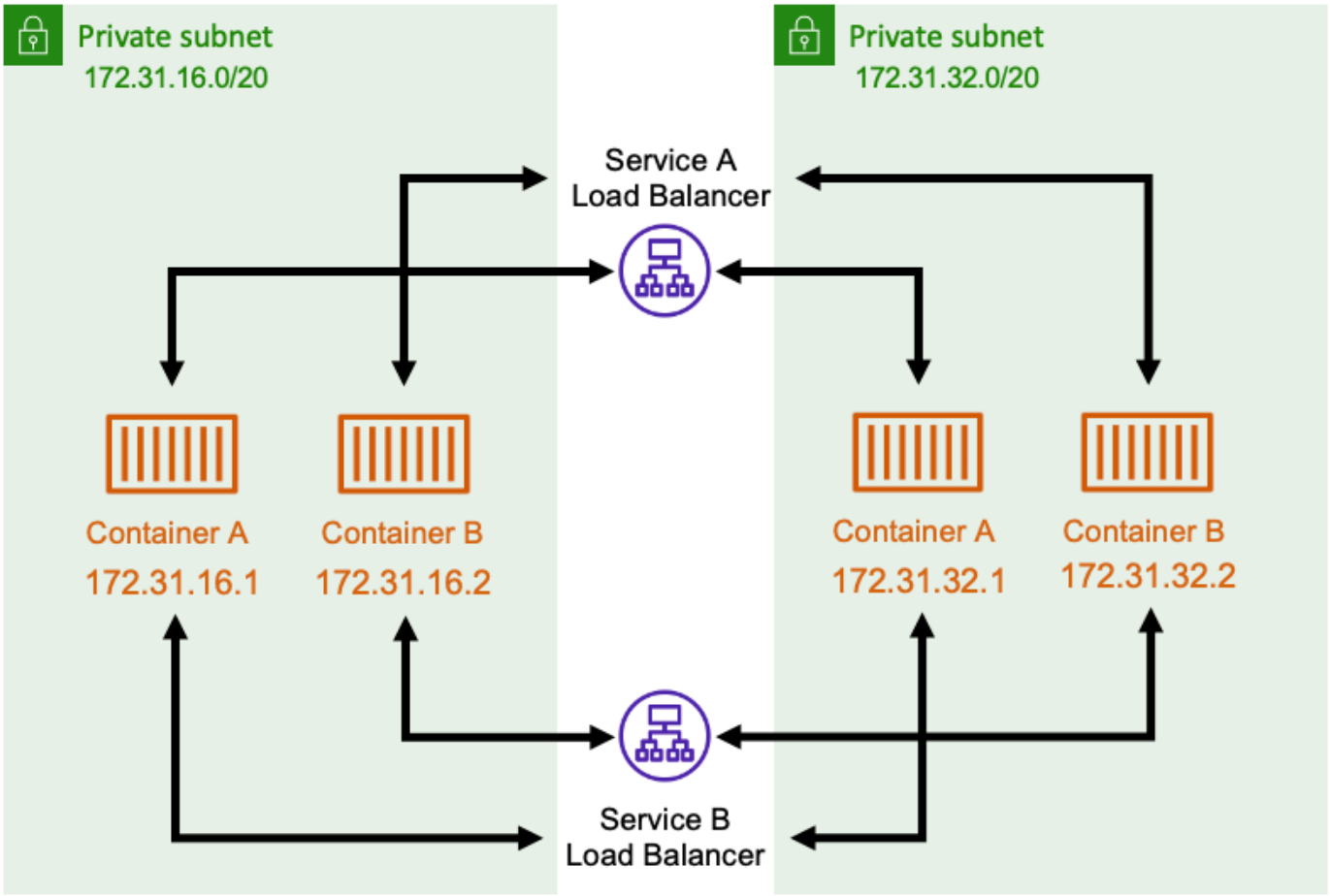
realizar un seguimiento de las direcciones IP y los números de puerto, pero requiere que configure las aplicaciones de forma adecuada. Es posible que algunas aplicaciones preconstruidas que utilice no admitanSRVRegistros de.

Otra ventaja de laawsvpces que tiene un grupo de seguridad único para cada servicio. Puede configurar este grupo de seguridad para permitir conexiones entrantes sólo de los servicios ascendentes específicos que necesitan comunicarse con ese servicio.

La principal desventaja de la comunicación directa de servicio a servicio mediante la detección de servicios es que debe implementar una lógica adicional para tener reintentos y lidiar con fallas de conexión. Los registros DNS tienen un periodo de tiempo de vida (TTL) que controla el tiempo durante el que se almacenan en caché. El registro DNS tarda algún tiempo en actualizarse y la caché caducar para que las aplicaciones puedan recoger la última versión del registro DNS. Por lo tanto, su aplicación podría terminar resolviendo el registro DNS para apuntar a otro contenedor que ya no está allí. Su aplicación necesita manejar reintentos y tener lógica para ignorar los backends defectuosos.

Uso de un balanceador de carga interno

Otro enfoque para la comunicación de servicio a servicio es usar un equilibrador de carga interno. Un equilibrador de carga interno existe completamente dentro de su VPC y solo es accesible a los servicios dentro de su VPC.



El equilibrador de carga mantiene una alta disponibilidad mediante la implementación de recursos redundantes en cada subred. Cuando un contenedor de`service`Anecesita comunicarse con un contenedor de`service`B, abre una conexión con el balanceador de carga. A continuación, el equilibrador de carga abre una conexión a un contenedor des`service B`. El balanceador de carga actúa como un lugar centralizado para administrar todas las conexiones entre cada servicio.

Si un contenedor de`service`B, entonces el equilibrador de carga puede eliminar ese contenedor del grupo. El equilibrador de carga también realiza comprobaciones de estado con cada destino descendente de su grupo y puede eliminar automáticamente los objetivos defectuosos del grupo hasta que vuelvan a estar en buen estado. Las aplicaciones ya no necesitan ser conscientes de cuántos contenedores descendentes hay allí. Solo abren sus conexiones con el balanceador de carga.

Este enfoque es ventajoso para todos los modos de red. El equilibrador de carga puede realizar un seguimiento de las direcciones IP de las tareas cuando se utiliza el`awsvpc`, así como combinaciones más avanzadas de dirección IP y puerto cuando se utiliza el`bridge`en el modo de red. Distribuye el

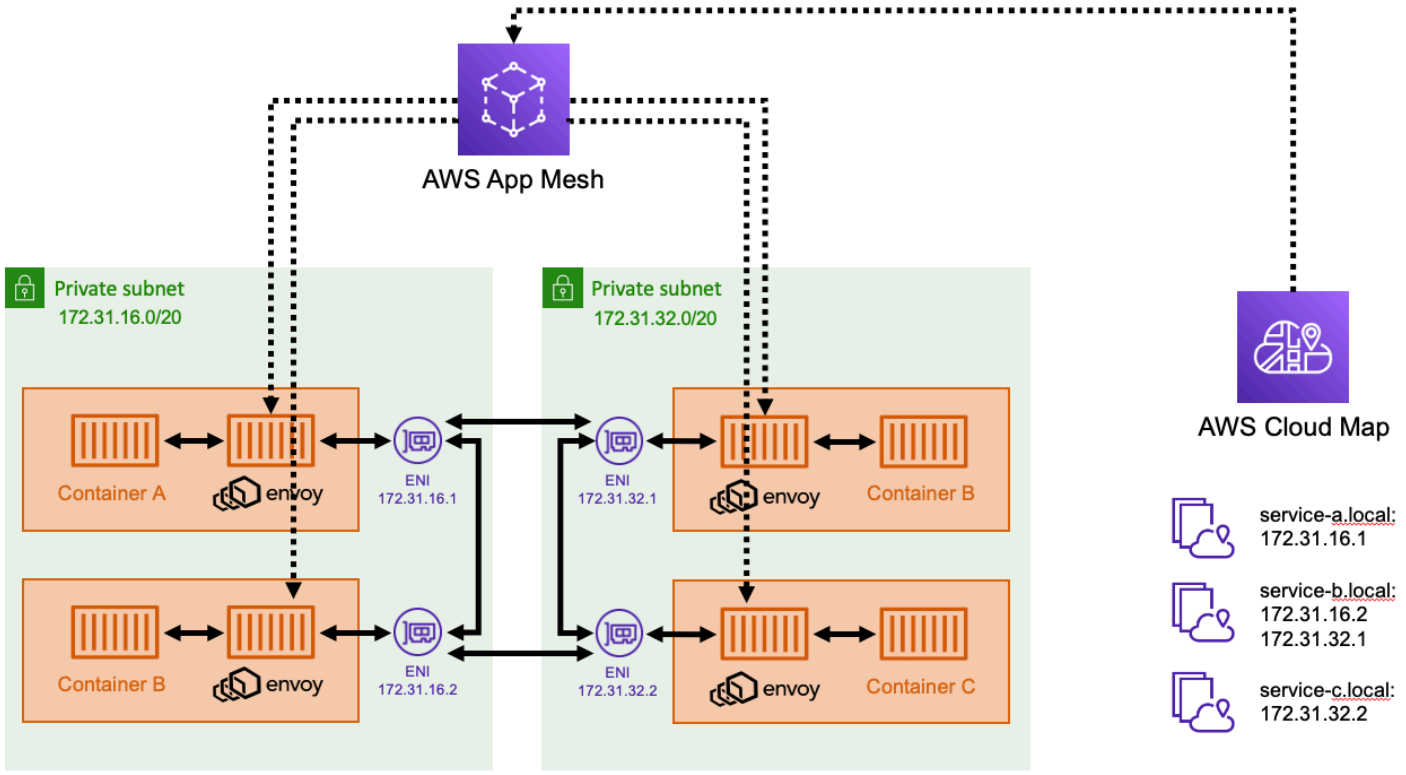
tráfico de manera uniforme en todas las combinaciones de direcciones IP y puertos, incluso si varios contenedores están alojados en la misma instancia de Amazon EC2, solo en puertos diferentes.

La única desventaja de este enfoque es el costo. Para estar altamente disponible, el equilibrador de carga necesita tener recursos en cada zona de disponibilidad. Esto agrega un costo adicional debido a la sobrecarga de pagar por el equilibrador de carga y por la cantidad de tráfico que atraviesa el equilibrador de carga.

Sin embargo, puede reducir los costes generales haciendo que varios servicios compartan un equilibrador de carga. Esto es especialmente adecuado para los servicios REST que utilizan un Application Load Balancer. Puede crear reglas de enrutamiento basadas en rutas que redirijan el tráfico a diferentes servicios. Por ejemplo,`/api/user/*`podría enrutarse a un contenedor que forma parte del`user`servicio, mientras que`/api/order/*`podría enrutar a la`order`Servicio de. Con este enfoque, solo paga por un Application Load Balancer y tiene una URL coherente para su API. Sin embargo, puede dividir el tráfico en varios microservicios en el back-end.

Uso de una malla de servicios

AWS App Meshes una malla de servicios que puede ayudarle a administrar una gran cantidad de servicios y tener un mejor control de cómo se enruta el tráfico entre los servicios. App Mesh funciona como intermediario entre la detección de servicios básicos y el equilibrio de carga. Con App Mesh, las aplicaciones no interactúan directamente entre sí, pero tampoco usan un equilibrador de carga centralizado. En su lugar, cada copia de su tarea va acompañada de un sidecar proxy de Envoy. Para obtener más información, consulte ¿Qué es AWS App Mesh? en la Guía del usuario de AWS App Mesh.



En el diagrama anterior, cada tarea tiene un sidecar proxy de Envoy. Este sidecar es responsable de enviar proxy todo el tráfico entrante y saliente para la tarea. El plano de control de App Mesh utilizaAWS Cloud Mappara obtener la lista de servicios disponibles y las direcciones IP de tareas específicas. A continuación, App Mesh entrega la configuración al sidecar proxy Envoy. Esta configuración incluye la lista de contenedores disponibles a los que se pueden conectar. El sidecar proxy de Envoy también realiza comprobaciones de estado contra cada objetivo para asegurarse de que están disponibles.

Este enfoque proporciona las características del descubrimiento de servicios, con la facilidad del equilibrador de carga administrado. Las aplicaciones no implementan tanta lógica de equilibrio de carga dentro de su código porque el sidecar proxy Envoy maneja ese equilibrio de carga. El proxy de Envoy se puede configurar para detectar errores y reintentar solicitudes fallidas. Además, también se puede configurar para utilizar MTL para cifrar el tráfico en tránsito y asegurarse de que sus aplicaciones se comunican a un destino verificado.

Hay algunas diferencias entre un proxy de Envoy y un balanceador de carga. En resumen, con el proxy de Envoy, usted es responsable de implementar y administrar su propio sidecar proxy de Envoy. El sidecar proxy Envoy utiliza parte de la CPU y la memoria asignadas a la tarea de

Amazon ECS. Esto agrega cierta sobrecarga al consumo de recursos de tareas y carga de trabajo operacional adicional para mantener y actualizar el proxy cuando sea necesario.

App Mesh y un proxy de Envoy permiten una latencia extremadamente baja entre tareas. Esto se debe a que el proxy de Envoy se ejecuta colocado en cada tarea. Sólo hay una instancia para el salto de red de instancias, entre un proxy de Envoy y otro proxy de Envoy. Esto significa que también hay menos sobrecarga de red en comparación con cuando se usan equilibradores de carga. Cuando se utilizan equilibradores de carga, hay dos saltos de red. La primera es desde la tarea ascendente hasta el equilibrador de carga, y la segunda es desde el equilibrador de carga hasta la tarea descendente.

Servicios de red a través deAWScuentas y VPC

Si forma parte de una organización con varios equipos y divisiones, probablemente implemente servicios de forma independiente en VPC independientes dentro de unAWSo en VPC asociadas con variasAWSCuentas de. Independientemente de la forma en que implemente sus servicios, le recomendamos que complemente los componentes de red para ayudar a enrutar el tráfico entre VPC. Para esto, variosAWSse pueden utilizar para complementar los componentes de red existentes.

- AWS Transit Gateway: debe tener en cuenta primero este servicio de red. Este servicio sirve como centro central para enrutar las conexiones entre Amazon VPC,AWScuentas y redes locales. Para obtener más información, consulte¿Qué es una gateway de tránsito?en laGuía de pasarelas de tránsito de Amazon VPC.
- Compatibilidad con Amazon VPC y VPN: puede utilizar este servicio para crear conexiones VPN de sitio a sitio para conectar redes locales a su VPC. Para obtener más información, consulte ¿Qué es AWS Site-to-Site VPN? en la Guía del usuario de AWS Site-to-Site VPN.
- Amazon VPC: puede utilizar el emparejamiento de Amazon VPC para ayudarle a conectar varias VPC, ya sea en la misma cuenta o entre cuentas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la Amazon VPC Peering Guide.
- VPC compartidas: puede utilizar una VPC y subredes VPC en varias redesAWSCuentas de. Para obtener más información, consulteUsar VPC compartidasen laGuía del usuario de Amazon VPC.

Optimización y solución de problemas

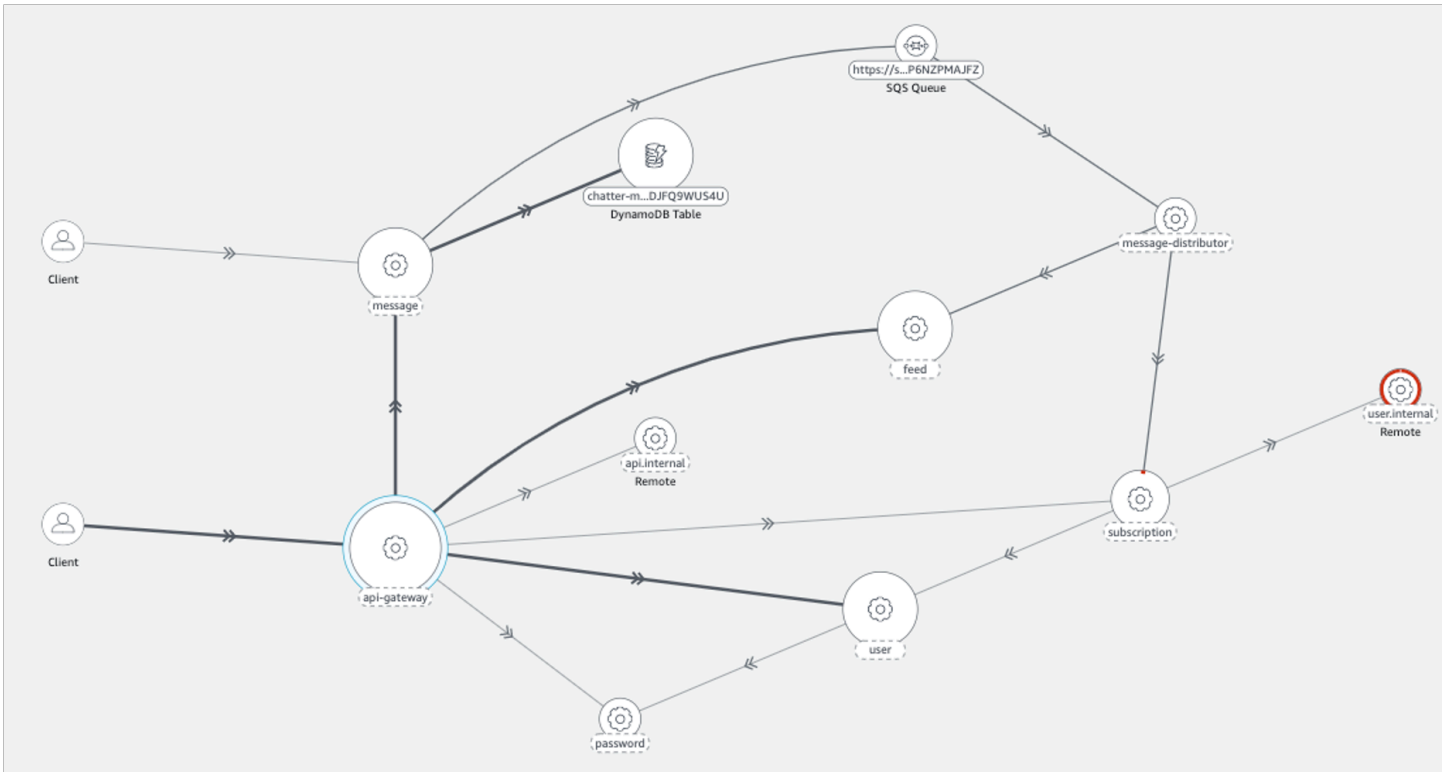
Los siguientes servicios y características pueden ayudarle a obtener información sobre sus configuraciones de red y servicio. Puede utilizar esta información para solucionar problemas de red y optimizar mejor los servicios.

Información de Container CloudWatch

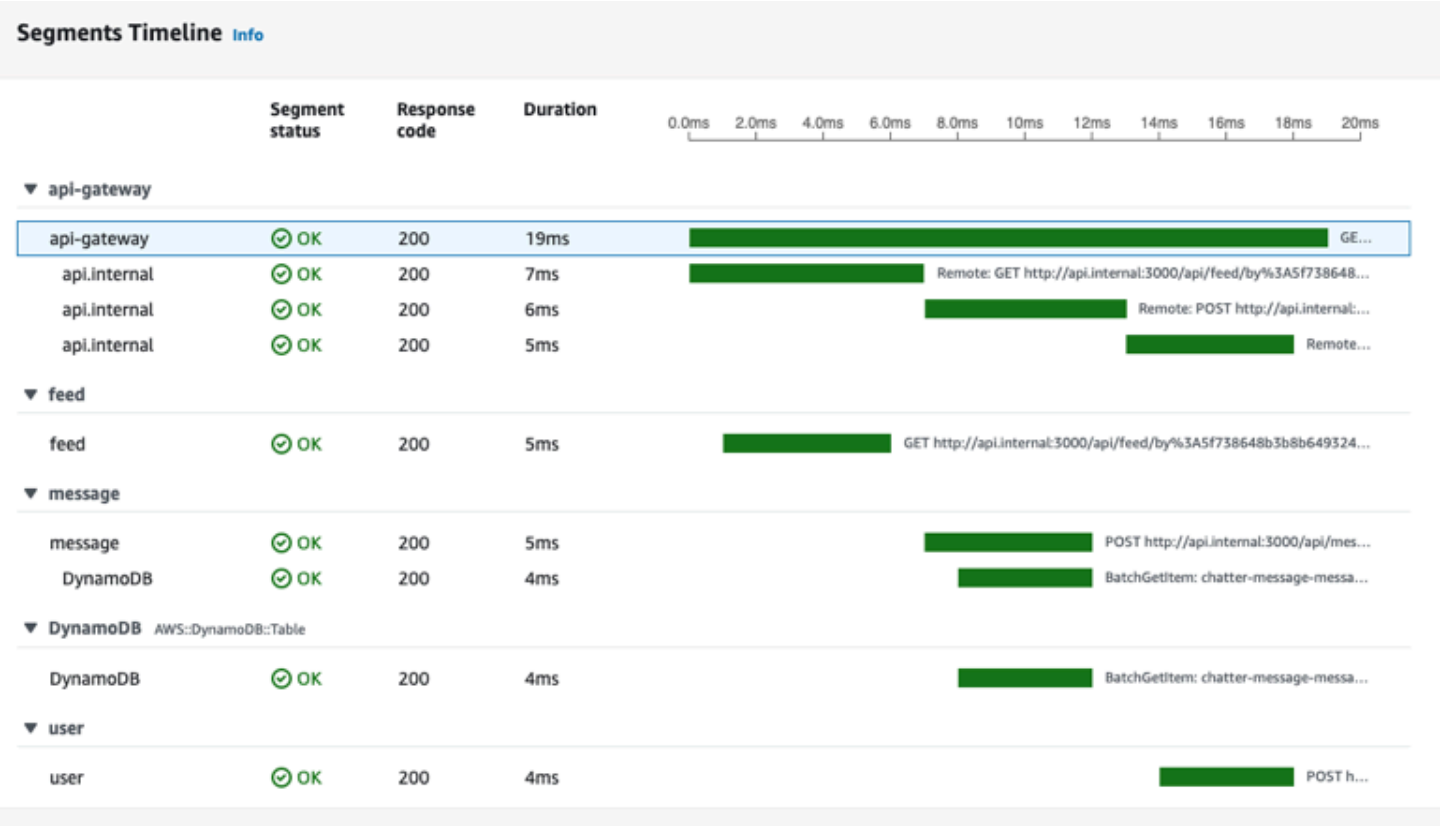
CloudWatch Container Insights recopila, agrega y resume métricas y registros de las aplicaciones y microservicios en contenedores. Las métricas incluyen la utilización de recursos como CPU, memoria, disco y red. Están disponibles en paneles automáticos de CloudWatch. Para obtener más información, consulteConfiguración de Container Insights en Amazon ECSen laGuía del usuario de Amazon CloudWatch.

AWS X-Ray

AWS X-Rayes un servicio de seguimiento que puede utilizar para recopilar información sobre las solicitudes de red que realiza su aplicación. Puede usar el SDK para instrumentar su aplicación y capturar tiempos y códigos de respuesta del tráfico entre sus servicios y entre sus servicios yAWSPuntos de enlace de servicio de. Para obtener más información, consulte¿Qué es ?AWS X-Rayen laAWS X-RayGuía para desarrolladores.



También puede explorarAWS X-Raygráficos de cómo su red de servicios entre sí. O bien, úsalos para explorar estadísticas agregadas sobre el rendimiento de cada enlace de servicio a servicio. Por último, puede profundizar en cualquier transacción específica para ver cómo los segmentos que representan llamadas de red están asociados con esa transacción en particular.



Puede usar estas características para identificar si hay un cuello de botella de red o si un servicio específico dentro de la red no funciona como se esperaba.

Logs de flujo de VPC

Puede utilizar los registros de flujo de Amazon VPC para analizar el rendimiento de la red y depurar problemas de conectividad. Con los registros de flujo de VPC habilitados, puede capturar un registro de todas las conexiones de su VPC. Entre ellas se incluyen las conexiones a interfaces de red asociadas con Elastic Load Balancing, Amazon RDS, puertas de enlace NAT y otras clavesAWSservicios que puede estar utilizando. Para obtener más información, consulte Registros de flujo de VPC en la Guía del usuario de Amazon VPC.

Consejos de ajuste de red

Hay algunas configuraciones que puede ajustar con precisión para mejorar su red.

Nofile ulimit

Si espera que su aplicación tenga mucho tráfico y maneje muchas conexiones simultáneas, debe tener en cuenta la cuota del sistema para el número de archivos permitidos. Cuando hay muchos sockets de red abiertos, cada uno debe estar representado por un descriptor de archivo. Si la cuota del descriptor de archivo es demasiado baja, limitará los sockets de red,. Esto da como resultado conexiones fallidas o errores. Puede actualizar la cuota específica del contenedor para el número de archivos en la definición de tareas de Amazon ECS. Si está ejecutando Amazon EC2 (en lugar deAWS Fargate), es posible que también tenga que ajustar estas cuotas en su instancia subyacente de Amazon EC2.

net de sysctl

Otra categoría de configuración sintonizable es la`sysctl`NET. Debe consultar la configuración específica para su distribución de Linux de elección. Muchas de estas configuraciones ajustan el tamaño de los búferes de lectura y escritura. Esto puede ayudar en algunas situaciones cuando se ejecutan instancias de Amazon EC2 a gran escala que tienen muchos contenedores en ellas.

Prácticas recomendadas: escalado automático y administración de capacidad

Amazon ECS se utiliza para ejecutar cargas de trabajo de aplicaciones en contenedores de todos los tamaños. Esto incluye tanto los extremos de los entornos de prueba mínimos como los grandes entornos de producción que operan a escala global.

Con Amazon ECS, como todos losAWSLos servicios de, paga únicamente por lo que utiliza. Cuando se diseña adecuadamente, puede ahorrar costos haciendo que su aplicación consuma solo los recursos que necesita en el momento en que los necesite. Esta guía de prácticas recomendadas muestra cómo ejecutar sus cargas de trabajo de Amazon ECS de una manera que satisfaga sus objetivos de nivel de servicio, sin dejar de funcionar de manera rentable.

Temas

Determinar el tamaño de tarea

Una de las opciones más importantes a tomar al implementar contenedores en Amazon ECS es el tamaño de los contenedores y las tareas. Los tamaños de los contenedores y las tareas son esenciales para el escalado y la planificación de la capacidad. En Amazon ECS, hay dos métricas de recursos utilizadas para la capacidad: Memoria y CPU. La CPU se mide en unidades de 1/1024 de una vCPU completa (donde 1024 unidades es igual a 1 vCPU completa). La memoria se mide en megabytes. En la definición de tarea, puede declarar reservas y límites de recursos.

Cuando declara una reserva, declara la cantidad mínima de recursos que requiere una tarea. Su tarea recibe al menos la cantidad de recursos solicitados. Es posible que su aplicación pueda usar

más CPU o memoria que la reserva que declara. Sin embargo, esto está sujeto a cualquier límite que usted también haya declarado. El uso de más de la cantidad de la reserva se conoce como explosión. En Amazon ECS, las reservas están garantizadas. Por ejemplo, si utiliza instancias de Amazon EC2 para proporcionar capacidad, Amazon ECS no coloca una tarea en una instancia en la que no se pueda realizar la reserva.

Un límite es la cantidad máxima de unidades de CPU o memoria que puede usar su contenedor o tarea. Cualquier intento de usar más CPU que este imit resulta en la limitación. Cualquier intento de usar más memoria hace que su contenedor se detenga.

Elegir estos valores puede resultar complicado. Esto se debe a que los valores más adecuados para su aplicación dependen en gran medida de los requisitos de recursos de su aplicación. La prueba de carga de su aplicación es la clave para una planificación exitosa de los requisitos de recursos y una mejor comprensión de los requisitos de su aplicación.

Aplicaciones sin estado

Para aplicaciones sin estado que escalan horizontalmente, como una aplicación detrás de un equilibrador de carga, se recomienda determinar primero la cantidad de memoria que consume la aplicación cuando sirve solicitudes. Para ello, puede utilizar herramientas tradicionales como `psortop` o soluciones de supervisión como CloudWatch Container Insights.

Al determinar una reserva de CPU, considere cómo desea escalar su aplicación para satisfacer los requisitos de su empresa. Puede utilizar reservas de CPU más pequeñas, como 256 unidades de CPU (o 1/4 vCPU), para escalar horizontalmente de forma precisa que minimice el costo. Pero, es posible que no se escalen lo suficientemente rápido como para satisfacer picos significativos en la demanda. Puede utilizar reservas de CPU más grandes para escalar más rápidamente y, por lo tanto, igualar los picos de demanda más rápidamente. Sin embargo, las reservas de CPU más grandes son más costosas.

Otras aplicaciones

Para aplicaciones que no escalan horizontalmente, como trabajadores individuales o servidores de bases de datos, la capacidad y el costo disponibles representan sus consideraciones más importantes. Debe elegir la cantidad de memoria y CPU en función de lo que las pruebas de carga indican que necesita servir el tráfico para cumplir con su objetivo de nivel de servicio. Amazon ECS garantiza que la aplicación se coloque en un host con capacidad adecuada.

Configuración de escalado automático de servicio

Un servicio de Amazon ECS es una colección gestionada de tareas. Cada servicio tiene una definición de tarea asociada, un recuento de tareas deseado y una estrategia de ubicación opcional. El escalado automático del servicio Amazon ECS se implementa a través del servicio Application Auto Scaling. Application Auto Scaling utiliza las métricas de CloudWatch como fuente para escalar las métricas. También utiliza alarmas de CloudWatch para establecer umbrales sobre cuándo escalar el servicio dentro o fuera. Proporcione los umbrales para la escala, ya sea estableciendo un destino de métrica, denominadoescalado de seguimiento de destino, o especificando umbrales, denominadosescalado por pasos. Después de configurar Application Auto Scaling, calcula continuamente el recuento de tareas deseado apropiado para el servicio. También notifica a Amazon ECS cuándo debe cambiar el recuento de tareas deseado, ya sea ampliándolo o escalándolo.

Para utilizar el escalado automático del servicio de manera eficaz, debe elegir una métrica de escala adecuada. En las siguientes secciones se describe cómo elegir una métrica.

Caracterización de la aplicación

Escalar correctamente una aplicación requiere conocer las condiciones en las que se debe escalar la aplicación y cuándo se debe escalar hacia fuera. En esencia, una aplicación debe ampliarse si se prevé que la demanda supere la capacidad. Por el contrario, una aplicación se puede escalar para ahorrar costos cuando los recursos superan la demanda.

Identificar una métrica de utilización

Para escalar eficazmente, es fundamental identificar una métrica que indique la utilización o la saturación. Esta métrica debe mostrar las siguientes propiedades para ser útil para escalar.

- La métrica debe estar correlacionada con la demanda. Cuando los recursos se mantienen estables, pero la demanda cambia, el valor de la métrica también debe cambiar. La métrica debe aumentar o disminuir cuando la demanda aumenta o disminuye.
- El valor de la métrica debe escalar en proporción a la capacidad. Cuando la demanda se mantiene constante, la adición de más recursos debe dar lugar a un cambio proporcional en el valor de la métrica. Por lo tanto, duplicar el número de tareas debería hacer que la métrica disminuya un 50%.

La mejor manera de identificar una métrica de utilización es mediante pruebas de carga en un entorno de preproducción, como un entorno de ensayo. Las soluciones de pruebas de carga

comerciales y de código abierto están ampliamente disponibles. Estas soluciones normalmente pueden generar carga sintética o simular tráfico real de usuarios.

Para iniciar el proceso de prueba de carga, debe comenzar creando paneles para las métricas de utilización de su aplicación. Estas métricas incluyen la utilización de la CPU, la utilización de la memoria, las operaciones de E/S, la profundidad de la cola de E/S y el rendimiento de la red. Puede recopilar estas métricas con un servicio como CloudWatch Container Insights. O bien, utilice el servicio gestionado de Amazon para Prometheus junto con el servicio gestionado de Amazon para Grafana. Durante este proceso, asegúrese de recopilar y trazar métricas para los tiempos de respuesta de la aplicación o las tasas de finalización del trabajo.

Al realizar pruebas de carga, comience con una pequeña solicitud o tasa de inserción de trabajos. Mantenga esta tasa estable durante varios minutos para permitir que su aplicación se caliente. Luego, aumente lentamente la velocidad y manténgala firme durante unos minutos. Repita este ciclo, aumentando la tasa cada vez hasta que los tiempos de respuesta o finalización de la aplicación sean demasiado lentos para cumplir con sus objetivos de nivel de servicio (SLO).

Durante la prueba de carga, examine cada una de las métricas de utilización. Las métricas que aumentan junto con la carga son los mejores candidatos para servir como sus mejores métricas de utilización.

A continuación, identifique el recurso que alcanza la saturación. Al mismo tiempo, examine también las métricas de utilización para ver cuál se aplana primero en un nivel alto. O bien, examine cuál alcanza el pico y luego bloquea su aplicación primero. Por ejemplo, si la utilización de la CPU aumenta de 0% a 70 -80% a medida que agrega carga, entonces permanece en esa lebel después de que se agrega aún más carga, entonces es seguro decir que la CPU está saturada. Dependiendo de la arquitectura de la CPU, es posible que nunca llegue al 100%. Por ejemplo, suponga que la utilización de la memoria aumenta a medida que agrega carga y, a continuación, la aplicación se bloquea repentinamente cuando alcanza el límite de memoria de la instancia de Amazon EC2 o la tarea. En esta situación, es probable que la memoria se haya consumido por completo. La aplicación puede consumir varios recursos. Por lo tanto, elija la métrica que representa el recurso que se agota primero.

Por último, intente probar la carga de nuevo después de duplicar el número de tareas o instancias de Amazon EC2. Supongamos que la métrica clave aumenta, o disminuye, a la mitad de la tasa que antes. Si este es el caso, entonces la métrica es proporcional a la capacidad. Esta es una buena métrica de utilización para el escalado automático.

Ahora considere este hipotético escenario. Supongamos que carga prueba una aplicación y encuentra que la utilización de la CPU finalmente alcanza el 80% a 100 solicitudes por segundo. Cuando se agrega más carga, ya no aumenta la utilización de la CPU. Sin embargo, hace que su aplicación responda más lentamente. A continuación, vuelva a ejecutar la prueba de carga, duplicando el número de tareas pero manteniendo la velocidad en su valor máximo anterior. Si encuentra que la utilización promedio de la CPU cae a alrededor del 40%, entonces la utilización promedio de la CPU es un buen candidato para una métrica de escalado. Por otro lado, si la utilización de la CPU se mantiene en el 80% después de aumentar el número de tareas, entonces la utilización promedio de la CPU no es una buena métrica de escalado. En ese caso, se necesita más investigación para encontrar una métrica adecuada.

Modelos de aplicación comunes y propiedades de escalado

Software de todo tipo se ejecutan enAWS. Muchas cargas de trabajo son propias, mientras que otras se basan en el popular software de código abierto. Independientemente de dónde se originen, hemos observado algunos patrones de diseño comunes para los servicios. Cómo escalar eficazmente depende en gran parte del patrón.

El eficiente servidor vinculado a la CPU

El eficiente servidor vinculado a la CPU no utiliza casi ningún otro recurso que no sea la CPU y el rendimiento de la red. Cada solicitud puede ser manejada por la aplicación sola. Las solicitudes no dependen de otros servicios, como bases de datos. La aplicación puede manejar cientos de miles de solicitudes simultáneas y puede utilizar eficientemente varias CPU para hacerlo. Cada solicitud es atendida por un subproceso dedicado con poca sobrecarga de memoria, o hay un bucle de eventos asincrónico que se ejecuta en cada CPU que presta servicios a las solicitudes. Cada réplica de la aplicación es igualmente capaz de manejar una solicitud. El único recurso que podría agotarse antes de la CPU es el ancho de banda de red. En los servicios límite de CPU, la utilización de la memoria, incluso en el máximo rendimiento, es una fracción de los recursos disponibles.

Este tipo de aplicación es adecuada para el escalado automático basado en CPU. La aplicación goza de la máxima flexibilidad en términos de escalado. Se puede escalar verticalmente proporcionando instancias de Amazon EC2 más grandes o vCPUs Fargate. Además, también se puede escalar horizontalmente agregando más réplicas. Al agregar más réplicas, o duplicar el tamaño de la instancia, se reduce a la mitad la utilización media de la CPU en relación con la capacidad.

Si utiliza la capacidad de Amazon EC2 para esta aplicación, considere colocarla en instancias optimizadas para cómputos, como la `c5or c6g` familia.

El servidor eficiente enlazado a la memoria

El eficiente servidor enlazado a memoria asigna una cantidad significativa de memoria por solicitud. En la concurrencia máxima, pero no necesariamente el rendimiento, la memoria se agota antes de que se agoten los recursos de la CPU. La memoria asociada a una solicitud se libera cuando finaliza la solicitud. Se pueden aceptar solicitudes adicionales siempre y cuando haya memoria disponible.

Este tipo de aplicación es adecuada para el escalado automático basado en memoria. La aplicación goza de la máxima flexibilidad en términos de escalado. Se puede escalar verticalmente proporcionando recursos de memoria más grandes de Amazon EC2 o Fargate. Además, también se puede escalar horizontalmente agregando más réplicas. Agregar más réplicas, o duplicar el tamaño de la instancia, puede reducir a la mitad la utilización media de la memoria relativa a la capacidad.

Si utiliza la capacidad de Amazon EC2 para esta aplicación, considere colocarla en instancias optimizadas para la memoria, como la`r5`o`r6g`familia.

Algunas aplicaciones enlazadas a la memoria no liberan la memoria asociada a una solicitud cuando finaliza, por lo que una reducción en la concurrencia no da lugar a una reducción en la memoria utilizada. Para ello, no recomendamos que utilice escalado basado en memoria.

El servidor basado en el trabajador

El servidor basado en el trabajador procesa una solicitud para cada subproceso de trabajo individual una tras otra. Los subprocesos de trabajo pueden ser subprocesos ligeros, como subprocesos POSIX. También pueden ser hilos de mayor peso, como procesos UNIX. Independientemente del hilo que sean, siempre hay una simultaneidad máxima que la aplicación puede admitir. Por lo general, el límite de concurrencia se establece proporcionalmente a los recursos de memoria disponibles. Si se alcanza el límite de simultaneidad, las solicitudes adicionales se colocan en una cola de trabajo atrasado. Si la cola de trabajo atrasado se desborda, las solicitudes entrantes adicionales se rechazan inmediatamente. Las aplicaciones comunes que se ajustan a este patrón incluyen el servidor web Apache y Gunicorn.

La concurrencia de solicitudes suele ser la mejor métrica para escalar esta aplicación. Debido a que hay un límite de simultaneidad para cada réplica, es importante escalar hacia fuera antes de que se alcance el límite medio.

La mejor manera de obtener métricas de concurrencia de solicitudes es hacer que su aplicación los informe a CloudWatch. Cada réplica de la aplicación puede publicar el número de solicitudes simultáneas como una métrica personalizada con una frecuencia alta. Recomendamos que la frecuencia esté configurada para ser al menos una vez cada minuto. Una vez recopilados

varios informes, puede utilizar la simultaneidad media como métrica de escala. Esta métrica se calcula tomando la concurrencia total y dividiéndola por el número de réplicas. Por ejemplo, si la simultaneidad total es 1000 y el número de réplicas es 10, entonces la simultaneidad media es 100.

Si su aplicación está detrás de un Application Load Balancer, también puede usar la herramienta`ActiveConnectionCount`para el equilibrador de carga como factor en la métrica de escalado. La`ActiveConnectionCount`debe dividirse por el número de réplicas para obtener un valor promedio. El valor promedio se debe utilizar para escalar, a diferencia del valor de recuento sin procesar.

Para que este diseño funcione mejor, la desviación estándar de la latencia de respuesta debe ser pequeña a bajas tasas de solicitud. Recomendamos que, durante los períodos de baja demanda, la mayoría de las solicitudes se respondan en poco tiempo, y no hay muchas solicitudes que tardan mucho más que el tiempo promedio en responder. El tiempo medio de respuesta debe ser cercano al tiempo de respuesta del percentil 95. De lo contrario, podrían producirse desbordamientos de cola como resultado. Esto conduce a errores. Le recomendamos que proporcione réplicas adicionales cuando sea necesario para mitigar el riesgo de desbordamiento.

Servidor de espera

El servidor en espera realiza algún procesamiento para cada solicitud, pero depende en gran medida de uno o más servicios descendentes para funcionar. Las aplicaciones de contenedores a menudo hacen un uso intensivo de servicios descendentes, como bases de datos y otros servicios de API. Esto se debe a que estas aplicaciones tienden a utilizar pocos recursos de CPU y su máxima concurrencia en términos de memoria disponible.

El servicio de espera es adecuado tanto en el patrón de servidor vinculado a la memoria como en el patrón de servidor basado en el trabajo, dependiendo de cómo se diseñe la aplicación. Si la concurrencia de la aplicación sólo está limitada por la memoria, entonces la utilización media de la memoria se debe utilizar como métrica de escalado. Si la simultaneidad de la aplicación se basa en un límite de trabajo, se debe utilizar la simultaneidad media como métrica de escalado.

Servidor basado en Java

Si su servidor basado en Java está vinculado a la CPU y se escala proporcionalmente a los recursos de la CPU, entonces podría ser adecuado para el patrón de servidor dependiente de la CPU eficiente. Si ese es el caso, la utilización media de la CPU podría ser apropiada como métrica de escalado. Sin embargo, muchas aplicaciones Java no están vinculadas a la CPU, lo que las hace difíciles de escalar.

Para conseguir el mejor rendimiento, le recomendamos que asigne la mayor cantidad de memoria posible al montón de máquina virtual de Java (JVM). Las versiones recientes de la JVM, incluida la actualización 191 de Java 8 o posterior, establecen automáticamente el tamaño del montón lo más grande posible para que quepa dentro del contenedor. Esto significa que, en Java, la utilización de la memoria rara vez es proporcional a la utilización de la aplicación. A medida que aumenta la velocidad de solicitud y la simultaneidad, la utilización de la memoria permanece constante. Debido a esto, no recomendamos escalar servidores basados en Java en función de la utilización de la memoria. En su lugar, normalmente recomendamos escalar en la utilización de la CPU.

En algunos casos, los servidores basados en Java encuentran agotamiento del montón antes de agotar la CPU. Si su aplicación es propensa al agotamiento del montón en alta concurrencia, entonces las conexiones medias son la mejor métrica de escalado. Si su aplicación es propensa al agotamiento del montón a un alto rendimiento, entonces la tasa de solicitud promedio es la mejor métrica de escalado.

Servidores que utilizan otros tiempos de ejecución recopilados de basura

Muchas aplicaciones de servidor se basan en tiempos de ejecución que realizan la recolección de basura, como .NET y Ruby. Estas aplicaciones de servidor pueden encajar en uno de los patrones descritos anteriormente. Sin embargo, al igual que con Java, no recomendamos escalar estas aplicaciones en función de la memoria, porque su uso promedio de memoria observado a menudo no está correlacionado con el rendimiento o la concurrencia.

Para estas aplicaciones, le recomendamos que escale la utilización de la CPU si la aplicación está vinculada a la CPU. De lo contrario, se recomienda escalar en promedio el rendimiento o la simultaneidad promedio, en función de los resultados de las pruebas de carga.

Procesadores de Job

Muchas cargas de trabajo implican procesamiento asincrónico de trabajos. Incluyen aplicaciones que no reciben solicitudes en tiempo real, sino que se suscriben a una cola de trabajos para recibir trabajos. Para estos tipos de aplicaciones, la métrica de escala adecuada es casi siempre la profundidad de la cola. El crecimiento de la cola es una indicación de que el trabajo pendiente supera la capacidad de procesamiento, mientras que una cola vacía indica que hay más capacidad que trabajo por hacer.

AWS, como Amazon SQS y Amazon Kinesis Data Streams, proporcionan métricas de CloudWatch que se pueden utilizar para escalar. Para Amazon SQS,`ApproximateNumberOfMessagesVisible`es la mejor métrica. Para Kinesis Data Streams,

considere la posibilidad de utilizar el`MillisBehindLatest`, publicada por Kinesis Client Library (KCL). Esta métrica debe promediarse entre todos los consumidores antes de usarla para escalar.

Disponibilidad y capacidad

La disponibilidad de las aplicaciones es crucial para proporcionar una experiencia sin errores y para minimizar la latencia de las aplicaciones. La disponibilidad depende de contar con recursos accesibles y con capacidad suficiente para satisfacer la demanda.AWSproporciona varios mecanismos para administrar la disponibilidad. Para las aplicaciones alojadas en Amazon ECS, estas incluyen el escalado automático y las zonas de disponibilidad (AZ). La escala automática administra el número de tareas o instancias en función de las métricas definidas, mientras que las zonas de disponibilidad le permiten alojar la aplicación en ubicaciones aisladas pero geográficamente cercanas.

Al igual que con los tamaños de las tareas, la capacidad y la disponibilidad presentan ciertas compensaciones que debe tener en cuenta. Idealmente, la capacidad estaría perfectamente alineada con la demanda. Siempre habría suficiente capacidad para atender solicitudes y procesar trabajos para cumplir con los Objetivos de Nivel de Servicio (SLO), incluyendo una baja latencia y tasa de error. La capacidad nunca sería demasiado alta, lo que daría lugar a costos excesivos; tampoco sería nunca demasiado baja, lo que daría lugar a altas tasas de latencia y error.

El autoescalado es un proceso latente. En primer lugar, las métricas en tiempo real deben entregarse a CloudWatch. Luego, deben agregarse para el análisis, que puede tardar hasta varios minutos dependiendo de la granularidad de la métrica. CloudWatch compara las métricas con los umbrales de alarma para identificar la escasez o el exceso de recursos. Para evitar la inestabilidad, configure las alarmas para exigir que el umbral establecido se cruce durante unos minutos antes de que se apague la alarma. También lleva tiempo aprovisionar nuevas tareas y terminar tareas que ya no son necesarias.

Debido a estos retrasos potenciales en el sistema descrito, es importante que mantenga un margen de ampliación mediante el aprovisionamiento excesivo. Hacer esto puede ayudar a acomodar ráfagas a corto plazo en demanda. Esto también ayuda a su aplicación a atender solicitudes adicionales sin alcanzar la saturación. Como buena práctica, puede establecer su objetivo de escalado entre el 60 y el 80% de la utilización. Esto ayuda a su aplicación a manejar mejor las ráfagas de demanda adicional mientras la capacidad adicional aún está en proceso de aprovisionamiento.

Otra razón por la que recomendamos que aprovisione en exceso es para que pueda responder rápidamente a los errores de la zona de disponibilidad.AWSrecomienda que las cargas de trabajo de producción se sirvan desde varias zonas de disponibilidad. Esto se debe a que, si se produce un error en la zona de disponibilidad, las tareas que se están ejecutando en las zonas de disponibilidad restantes todavía pueden servir a la demanda. Si la aplicación se ejecuta en dos zonas de disponibilidad, debe duplicar el recuento normal de tareas. Esto es para que pueda proporcionar capacidad inmediata durante cualquier falla potencial. Si la aplicación se ejecuta en tres zonas de disponibilidad, le recomendamos que ejecute 1,5 veces el recuento normal de tareas. Es decir, ejecutar tres tareas por cada dos que se necesitan para servir ordinario.

Maximizar la velocidad de escalado

El autoescalado es un proceso reactivo que toma tiempo para surtir efecto. Sin embargo, hay algunas maneras de ayudar a minimizar el tiempo que se necesita para escalar hacia fuera.

Minimizar tamaño de imagen. Las imágenes más grandes tardan más en descargarse de un repositorio de imágenes y desempaquetarlas. Por lo tanto, mantener los tamaños de imagen más pequeños reduce la cantidad de tiempo que se necesita para iniciar un contenedor. Para reducir el tamaño de la imagen, puede seguir estas recomendaciones específicas:

- Si puede construir un binario estático o usar Golang, construya su imagen`FROM`rascar e incluir solo su aplicación binaria en la imagen resultante.
- Utilice imágenes base minimizadas de proveedores de distribución ascendente, como Amazon Linux o Ubuntu.
- No incluyas ningún artefacto de compilación en tu imagen final. El uso de compilaciones de varias etapas puede ayudar con esto.
- Compact`RUN`siempre que sea posible. Cada`RUN`crea una nueva capa de imagen, lo que lleva a un viaje de ida y vuelta adicional para descargar la capa. Una sola`RUN`que tiene varios comandos unidos por`&&`tiene menos capas que una con varias`RUN`etapas.
- Si desea incluir datos, como datos de inferencia ML, en la imagen final, incluya solo los datos necesarios para iniciar y comenzar a servir tráfico. Si obtiene datos a petición de Amazon S3 u otro almacenamiento sin afectar al servicio, almacene sus datos en esos lugares en su lugar.

Mantén tus imágenes cerca. Cuanto mayor sea la latencia de red, más tiempo tardará en descargar la imagen. Aloje sus imágenes en un repositorio en el mismoAWSRegión en la que se encuentra la carga de trabajo. Amazon ECR es un repositorio de imágenes de alto rendimiento que está disponible en todas las regiones en las que Amazon ECS está disponible. Evite atravesar Internet

o un enlace VPN para descargar imágenes de contenedor. El alojamiento de sus imágenes en la misma región mejora la fiabilidad general. Mitiga el riesgo de problemas de conectividad de red y problemas de disponibilidad en una región diferente. Como alternativa, también puede implementar la replicación entre regiones de Amazon ECR para ayudarlo con esto.

Reducir los umbrales de comprobación de estado del balanceador de carga Los equilibradores de carga realizan comprobaciones de estado antes de enviar tráfico a la aplicación. La configuración de comprobación de estado predeterminada para un grupo de destino puede tardar 90 segundos o más. Durante esto, comprueba el estado de salud y las solicitudes de recepción. Reducir el intervalo de comprobación de estado y el recuento de umbrales puede hacer que la aplicación acepte el tráfico más rápido y reducir la carga en otras tareas.

Considere el rendimiento de arranque en frío. Algunas aplicaciones usan tiempos de ejecución como Java realizan compilación Just-In-Time (JIT). El proceso de compilación al menos cuando se inicia puede mostrar el rendimiento de la aplicación. Una solución alternativa es reescribir las partes críticas de la carga de trabajo en idiomas que no imponen una penalización de rendimiento de inicio en frío.

Use las políticas de escalado por pasos, no de seguimiento de destino. Tiene varias opciones de Application Auto Scaling para las tareas de Amazon ECS. El seguimiento de objetivos es el modo más fácil de usar. Con él, todo lo que necesita hacer es establecer un valor objetivo para una métrica, como la utilización media de la CPU. A continuación, el escalador automático administra automáticamente el número de tareas necesarias para alcanzar ese valor. Sin embargo, le recomendamos que utilice el escalado de pasos en su lugar para que pueda reaccionar más rápidamente ante los cambios en la demanda. Con el escalado de pasos, puede definir los umbrales específicos para las métricas de escala y cuántas tareas agregar o quitar cuando se cruzan los umbrales. Y, lo que es más importante, puede reaccionar muy rápidamente a los cambios en la demanda minimizando la cantidad de tiempo que una alarma de umbral está infringiendo. Para obtener más información, consulteAuto Scaling de serviciosen laAmazon Elastic Container Service.

Si utiliza instancias de Amazon EC2 para proporcionar capacidad de clúster, tenga en cuenta las siguientes recomendaciones:

Utilice instancias de Amazon EC2 más grandes y volúmenes de Amazon EBS más rápidos. Puede mejorar las velocidades de descarga y preparación de imágenes utilizando una instancia de Amazon EC2 más grande y un volumen de Amazon EBS más rápido. Dentro de una familia de instancias de Amazon EC2 determinada, el rendimiento máximo de la red y Amazon EBS aumenta a medida que aumenta el tamaño de la instancia (por ejemplo, desde`m5.xlarge`De a`m5.2xlarge`). Además, también puede personalizar los volúmenes de Amazon EBS para aumentar su rendimiento y las

IOPS. Por ejemplo, si utilizagp2, utilice volúmenes más grandes que ofrezcan más rendimiento de línea base. Si utilizagp3Al crear el volumen, especifique el rendimiento y las IOPS.

Utilice el modo de red puente para las tareas que se ejecutan en instancias de Amazon EC2. Tareas que utilizanbridgeen Amazon EC2 se inician más rápido que las tareas que utilizan elawsvpcModo de red. CuandoawsvpcAmazon ECS conecta una elastic network interface (ENI) a la instancia antes de iniciar la tarea. Esto introduce latencia adicional. Sin embargo, hay varias compensaciones para el uso de redes de puente. Estas tareas no tienen su propio grupo de seguridad, y existen algunas implicaciones para el equilibrio de carga. Para obtener más información, consulteGrupos de destino del equilibrador de cargaen laGuía del usuario de Elastic Load Balancing.

Manifiesto de demanda

Algunas aplicaciones experimentan grandes choques repentinos en la demanda. Esto ocurre por una variedad de razones: un evento de noticias, una gran venta, un evento mediático o algún otro evento que se vuelva viral y haga que el tráfico aumente de forma rápida y significativa en un lapso de tiempo muy corto. Si no se planifica, esto puede hacer que la demanda supere rápidamente los recursos disponibles.

La mejor manera de manejar los choques de la demanda es anticiparlos y planificarlos en consecuencia. Dado que el escalado automático puede llevar tiempo, le recomendamos que escale la aplicación antes de que comience el choque de la demanda. Para obtener los mejores resultados, recomendamos tener un plan de negocio que implique una estrecha colaboración entre equipos que utilicen un calendario compartido. El equipo que está planeando el evento debe trabajar estrechamente con el equipo a cargo de la aplicación por adelantado. Esto le da a ese equipo tiempo suficiente para tener un plan de programación claro. Pueden programar la capacidad para escalar hacia fuera antes del evento y para escalar después del evento. Para obtener más información, consulteEscalado programadoen laGuía del usuario de la aplicación Auto Scaling.

Si tiene un plan de Support técnico para empresas, asegúrese también de trabajar con su administrador de cuentas técnico (TAM). Su TAM puede verificar sus cuotas de servicio y asegurarse de que se eleven las cuotas necesarias antes de que comience el evento. De esta manera, no golpea accidentalmente ninguna cuota de servicio. También pueden ayudarle con servicios de precalentamiento, como equilibradores de carga, para asegurarse de que su evento se desarrolle sin problemas.

Manejar los choques de demanda no programados es un problema más difícil. Los choques no programados, si son lo suficientemente grandes en amplitud, pueden hacer que la demanda supere rápidamente la capacidad. También puede superar la capacidad del escalado automático para

reaccionar. La mejor manera de prepararse para las perturbaciones no programadas es aprovisionar recursos excesivamente. Debe disponer de recursos suficientes para manejar la demanda máxima de tráfico prevista en cualquier momento.

Mantener la capacidad máxima en previsión de los choques de la demanda no programados puede ser costoso. Para mitigar el impacto en los costos, busque una métrica o evento indicador líder que prediga que es inminente un gran choque de demanda. Si la métrica o el evento proporciona un aviso anticipado significativo de forma fiable, inicie el proceso de escalado horizontal inmediatamente cuando se produzca el evento o cuando la métrica cruce el umbral específico establecido.

Si su aplicación es propensa a choques repentinos de demanda no programados, considere la posibilidad de agregar un modo de alto rendimiento a su aplicación que sacrifique la funcionalidad no crítica pero que mantenga la funcionalidad crucial para un cliente. Por ejemplo, suponga que su aplicación puede pasar de generar costosas respuestas personalizadas a servir una página de respuesta estática. En este escenario, puede aumentar significativamente el rendimiento sin escalar la aplicación en absoluto.

Por último, puede considerar romper los servicios monolíticos para lidiar mejor con los choques de la demanda. Si su aplicación es un servicio monolítico que es costoso de ejecutar y lento para escalar, es posible que pueda extraer o reescribir piezas críticas para el rendimiento y ejecutarlas como servicios independientes. Estos nuevos servicios pueden escalarse independientemente de los componentes menos críticos. Tener la flexibilidad para escalar la funcionalidad crítica del rendimiento por separado de otras partes de la aplicación puede reducir el tiempo que se tarda en agregar capacidad y ayudar a ahorrar costos.

Capacidad de clúster

Anteriormente en este tema, se discutió cómo escalar la cuenta de réplica para su mediante el uso de métricas de escala. Sus tareas también deben ejecutarse en recursos, incluidos recursos de CPU y memoria. Esto vuelve a relacionarse con el tema de la capacidad. En Amazon ECS, la capacidad se proporciona a través de dos proveedores principales:AWS FargateAmazon EC2.

Puede proporcionar capacidad a un clúster de Amazon ECS de varias maneras. Por ejemplo, puede iniciar instancias de Amazon EC2 y registrarlas en el clúster al inicio mediante el agente contenedor de Amazon ECS. Sin embargo, este método puede ser un reto porque necesita administrar el escalado por su cuenta. Por lo tanto, recomendamos que utilice proveedores de capacidad Amazon ECS. Administran el escalado de recursos para usted. Existen tres tipos de proveedores de capacidad: Amazon EC2, Fargate y Fargate Spot.

Los proveedores de capacidad Fargate y Fargate Spot manejan el ciclo de vida de las tareas de Fargate por usted. Fargate proporciona capacidad bajo demanda, y Fargate Spot proporciona capacidad Spot. Cuando se inicia una tarea, ECS aprovisiona un recurso de Fargate para usted. Este recurso de Fargate viene con las unidades de memoria y CPU que corresponden directamente a los límites de nivel de tarea declarados en la definición de tarea. Cada tarea recibe su propio recurso de Fargate, creando una relación 1:1 entre la tarea y los recursos informáticos.

Las tareas que se ejecutan en Fargate Spot están sujetas a interrupción. Las interrupciones vienen después de una advertencia de dos minutos. Estos ocurren durante períodos de gran demanda. Fargate Spot funciona mejor para cargas de trabajo tolerantes a interrupciones, como trabajos por lotes, entornos de desarrollo o ensayo. También son adecuados para cualquier otro escenario en el que la alta disponibilidad y la baja latencia no sean un requisito.

Puede ejecutar tareas de Fargate Spot junto con tareas bajo demanda de Fargate. Al usarlos juntos, recibirá la capacidad de aprovisionamiento de «ráfaga» a un costo menor.

ECS también puede gestionar la capacidad de instancia de Amazon EC2 para sus tareas. Cada proveedor de capacidad de Amazon EC2 está asociado a un grupo de Auto Scaling de Amazon EC2 que especifique. Cuando utiliza Amazon EC2 Capacity Provider, el Auto Scaling del clúster ECS mantiene el tamaño del grupo de escalado automático de Amazon EC2 para garantizar que se puedan realizar todas las tareas programadas.

Prácticas recomendadas para la capacidad

Agregue espacio de ampliación a su servicio, no al proveedor de capacidad. Los proveedores de capacidad de Amazon EC2 ofrecen un valor de capacidad objetivo. Si establece un valor inferior al 100%, ECS aprovisiona más instancias de Amazon EC2 de las necesarias para adaptarse a sus tareas. Tener varias instancias de Amazon EC2 listas para aceptar tareas puede ser útil. Sin embargo, cuando utiliza Amazon Virtual Private Cloud, el inicio de nuevas tareas requiere tiempo adicional para descargar la imagen y adjuntar una interfaz de red. Esta latencia añadida podría ser perjudicial para su resultado final.

Por lo tanto, le recomendamos que haga lo siguiente. En lugar de reducir la capacidad de destino de su proveedor de capacidad, aumente el número de réplicas en su servicio modificando la métrica de escalamiento de seguimiento de destino o los umbrales de escalado de paso del servicio. Para obtener más información acerca de las políticas de escalado relacionadas, consultePolíticas de escalado de seguimiento de destinoorPolíticas de escalado por pasosen laAmazon Elastic Container Service. El proveedor de capacidad de Amazon EC2 aprovisiona la capacidad necesaria para tareas adicionales mediante la adición de instancias adicionales al grupo de Auto Scaling. Esto ayuda a

garantizar que los recursos informáticos y de aplicaciones estén disponibles cuando los necesite. Por ejemplo, puede ayudar duplicando el número de tareas en un servicio ECS para acomodar una ráfaga inmediata del 100% de la demanda.

Elegir tamaños de tarea de Fargate

Si ejecuta sus tareas enAWS FargateEn la definición de esta, debe declarar los límites de memoria y CPU de la tarea en la definición de esta. ECS utiliza estos límites para determinar el tipo de instancia de Fargate en el que ejecutar la tarea. Los límites que determine deben ser superiores o iguales a las reservas que haya declarado. En la mayoría de los casos, puede establecerlas en la suma de las reservas de cada uno de los contenedores declarados en la definición de tarea. A continuación, redondea el número hasta el tamaño de instancia de Fargate más cercano. Para obtener más información acerca de los tamaños disponibles, consulteMemoria y CPU de tareaen laAmazon Elastic Container Service.

Elección del tipo de instancia Amazon EC2

Si utiliza Amazon EC2 para proporcionar capacidad para su clúster ECS, puede elegir entre una amplia selección de tipos de instancias. Todos los tipos de instancias y familias de Amazon EC2 son compatibles con ECS.

Para determinar qué tipos de instancia puede utilizar, comience eliminando los tipos de instancia o las familias de instancias que no cumplan los requisitos específicos de la aplicación. Por ejemplo, si su aplicación requiere una GPU, puede excluir cualquier tipo de instancia que no tenga una GPU. Sin embargo, también debe considerar otros requisitos. Por ejemplo, considere la arquitectura de CPU, el rendimiento de red y si el almacenamiento de instancias es un requisito. A continuación, examine la cantidad de CPU y memoria proporcionada por cada tipo de instancia. Como regla general, la CPU y la memoria deben ser lo suficientemente grandes como para contener al menos una réplica de la tarea que desea ejecutar.

Puede elegir entre los tipos de instancia que sean compatibles con su aplicación. Con instancias de mayor tamaño, puede lanzar más tareas al mismo tiempo. Además, con instancias más pequeñas, puede escalar horizontalmente de una manera más precisa para ahorrar costos. No es necesario que elija un único tipo de instancia de Amazon EC2 que se adapte a todas las aplicaciones del clúster. En su lugar, puede crear varios grupos de Auto Scaling,. Cada grupo de puede tener un tipo de instancia diferente. A continuación, puede crear un proveedor de capacidad de Amazon EC2 para cada uno de estos grupos. Por último, en la estrategia del proveedor de capacidad de su servicio y tarea, puede seleccionar el proveedor de capacidad que mejor se adapte a sus necesidades.

Uso de Amazon EC2 Spot y FARGATE_SPOT

La capacidad puntual puede proporcionar ahorros significativos en comparación con las instancias bajo demanda. La capacidad puntual es una capacidad excesiva que tiene un precio significativamente más bajo que la capacidad reservada o bajo demanda. La capacidad puntual es adecuada para el procesamiento por lotes y las cargas de trabajo de aprendizaje automático, así como para entornos de desarrollo y ensayo. De manera más general, es adecuado para cualquier carga de trabajo que tolere el tiempo de inactividad temporal.

Comprenda que las siguientes consecuencias porque es posible que la capacidad puntual no esté disponible todo el tiempo.

- En primer lugar, durante períodos de demanda extremadamente alta, la capacidad puntual podría no estar disponible. Esto puede provocar retrasos en el lanzamiento de la tarea de subasta de Fargate y de instancias de subasta de Amazon EC2. En estos eventos, los servicios de ECS reintentan iniciar tareas y los grupos de Auto Scaling de Amazon EC2 también reintentan iniciar instancias, hasta que la capacidad requerida esté disponible. Fargate y Amazon EC2 no reemplazan la capacidad de subasta por la capacidad bajo demanda.
- En segundo lugar, cuando aumenta la demanda general de capacidad, las instancias puntuales y las tareas pueden terminar con solo una advertencia de dos minutos. Después de enviar la advertencia, las tareas deben comenzar un apagado ordenado si es necesario antes de que la instancia finalice por completo. Esto ayuda a minimizar la posibilidad de errores. Para obtener más información acerca de un apagado elegante, consulteApagas elegantes con ECS.

Para ayudar a minimizar la escasez de capacidad puntual, tenga en cuenta las siguientes recomendaciones:

- Usar varias regiones y zonas de disponibilidad. La capacidad puntual varía según la región y la zona de disponibilidad. Puede mejorar la disponibilidad puntual ejecutando sus cargas de trabajo en varias regiones y zonas de disponibilidad. Si es posible, especifique subredes en todas las zonas de disponibilidad de las regiones donde ejecute sus tareas e instancias.
- Utilice varios tipos de instancias Amazon EC2. Cuando utiliza directivas de instancia mixta con Amazon EC2 Auto Scaling, se inician varios tipos de instancias en su grupo de Auto Scaling. Esto garantiza que una solicitud de capacidad puntual pueda cumplirse cuando sea necesario. Para maximizar la fiabilidad y minimizar la complejidad, utilice tipos de instancias con aproximadamente la misma cantidad de CPU y memoria en la directiva de instancias mixtas. Estas instancias pueden ser de una generación diferente o variantes del mismo tipo de instancia base. Tenga en cuenta

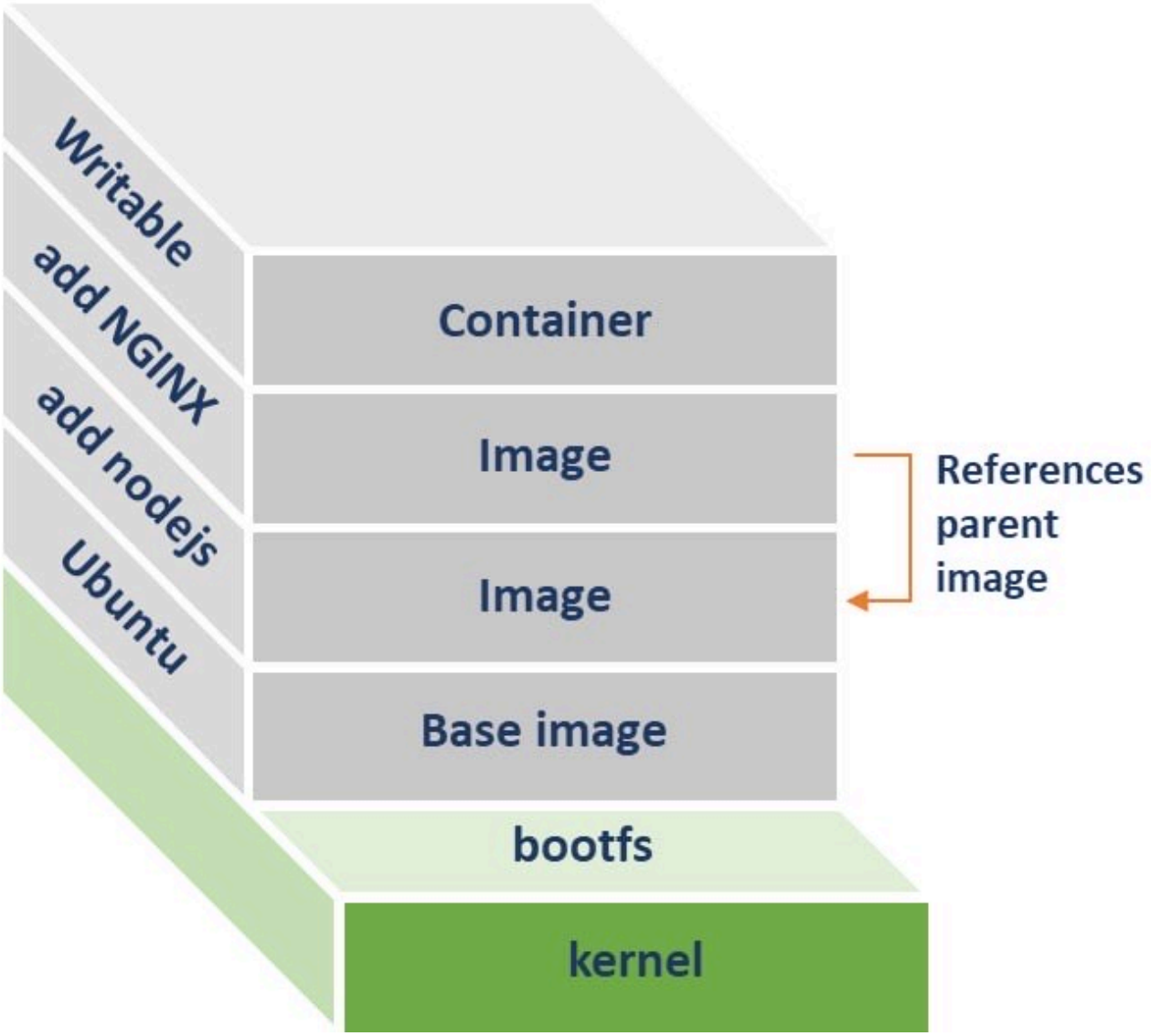
que pueden venir con funciones adicionales que puede que no necesite. Un ejemplo de tal lista podría incluir m4.large, m5.large, m5a.large, m5d.large, m5n.large, m5dn.large y m5ad.large. Para obtener más información, consulte la sección sobre Grupos de Auto Scaling con varios tipos de instancias y opciones de compra en la guía del usuario de Amazon EC2 Auto Scaling.

- Utilice la estrategia de asignación puntual optimizada para la capacidad. Con Amazon EC2 Spot, puede elegir entre las estrategias de asignación optimizadas para la capacidad y los costes. Si elige la estrategia de capacidad optimizada al lanzar una nueva instancia, Amazon EC2 Spot selecciona el tipo de instancia con mayor disponibilidad en la zona de disponibilidad seleccionada. Esto ayuda a reducir la posibilidad de que la instancia finalice poco después de que se inicie.

Prácticas recomendadas - Almacenamiento persistente

Puede utilizar Amazon ECS para ejecutar aplicaciones en contenedores con estado a escala medianteAWS, como Amazon EFS, Amazon EBS o Amazon FSx for Windows File Server, que proporcionan persistencia de datos a contenedores intrínsecamente efímeros. El términoPersistencia de datossignifica que los datos en sí duran más que el proceso que los creó. Persistencia de datos enAWSse logra mediante el acoplamiento de los servicios de computación y almacenamiento. Al igual que Amazon EC2, también puede utilizar Amazon ECS para desacoplar el ciclo de vida de las aplicaciones en contenedores de los datos que consumen y producen. UsoAWS, las tareas de Amazon ECS pueden conservar los datos incluso después de que finalicen las tareas.

De forma predeterminada, los contenedores no conservan los datos que producen. Cuando se termina un contenedor, los datos que escribió en su capa de escritura se destruyen con el contenedor. Esto hace que los contenedores sean adecuados para aplicaciones sin estado que no necesitan almacenar datos localmente. Las aplicaciones en contenedores que requieren persistencia de datos necesitan un back-end de almacenamiento que no se destruya cuando finaliza el contenedor de la aplicación.



Una imagen de contenedor se construye a partir de una serie de capas. Cada capa representa una instrucción en Dockerfile desde la que se creó la imagen. Cada capa es de sólo lectura, excepto para el contenedor. Es decir, cuando crea un contenedor, se agrega una capa de escritura sobre las capas subyacentes. Todos los archivos que el contenedor crea, elimina o modifica se escriben en la capa de escritura. Cuando el contenedor termina, la capa grabable también se elimina simultáneamente. Un nuevo contenedor que utiliza la misma imagen tiene su propia capa de escritura. Esta capa no incluye ningún cambio. Por lo tanto, los datos de un contenedor siempre deben almacenarse fuera de la capa de escritura del contenedor.

Con Amazon ECS, puede ejecutar contenedores con estado utilizando volúmenes. Amazon ECS está integrado con Amazon EFS de forma nativa y utiliza volúmenes integrados con Amazon EBS.

En el caso de los contenedores de Windows, Amazon ECS se integra con Amazon FSx for Windows File Server para proporcionar almacenamiento persistente.

Temas

Elección del tipo de almacenamiento adecuado para sus contenedores

Las aplicaciones que se ejecutan en un clúster de Amazon ECS pueden utilizar una variedad deAWSLos servicios de almacenamiento de información y productos de terceros para proporcionar almacenamiento persistente para cargas de trabajo con estado. Debe elegir su back-end de almacenamiento de información para su aplicación en contenedores en función de la arquitectura y los requisitos de almacenamiento de la aplicación. Para obtener más información acerca deAWSservicios de almacenamiento, consulteAlmacenamiento en la nubeAWS.

Para los clústeres de Amazon ECS que contienen instancias Linux o se utilizan con Fargate, Amazon ECS se integra con Amazon EFS y Amazon EBS para proporcionar almacenamiento de contenedores. La diferencia más distintiva entre Amazon EFS y Amazon EBS es que puede montar simultáneamente un sistema de archivos de Amazon EFS en miles de tareas de Amazon ECS. Por el contrario, los volúmenes de Amazon EBS no admiten el acceso simultáneo. Dado esto, Amazon EFS es la opción de almacenamiento recomendada para aplicaciones en contenedores que escalan horizontalmente. Esto se debe a que admite la concurrencia. Amazon EFS almacena sus datos de forma redundante en varias zonas de disponibilidad y ofrece acceso de baja latencia desde las tareas de Amazon ECS, independientemente de la zona de disponibilidad. Amazon EFS admite tareas que se ejecutan tanto en Amazon EC2 como en Fargate.

Supongamos que tiene una aplicación como una base de datos transaccional que requiere una latencia inferior a milisegundos y no necesita un sistema de archivos compartido cuando se escala horizontalmente. Para una aplicación de este tipo, se recomienda utilizar volúmenes de Amazon EBS para el almacenamiento persistente. Actualmente, Amazon ECS admite volúmenes de Amazon EBS únicamente para tareas alojadas en Amazon EC2. La Support con volúmenes de Amazon EBS no está disponible para tareas en Fargate. Antes de utilizar volúmenes de Amazon EBS con tareas

de Amazon ECS, primero debe adjuntar volúmenes de Amazon EBS a instancias de contenedor y gestionar volúmenes por separado del ciclo de vida de la tarea.

Para los clústeres que contienen instancias de Windows, Amazon FSx for Windows File Server proporciona almacenamiento persistente para los contenedores. Los sistemas de archivos de Amazon FSx for Windows File Server admiten las implementaciones Multi-AZ. A través de estas implementaciones, puede compartir un sistema de archivos con las tareas de Amazon ECS que se ejecutan en varias zonas de disponibilidad.

También puede utilizar el almacenamiento de instancias de Amazon EC2 para la persistencia de datos de las tareas de Amazon ECS alojadas en Amazon EC2 mediante montajes de enlace o volúmenes Docker. Cuando se utilizan montajes de enlace o volúmenes Docker, los contenedores almacenan datos en el sistema de archivos de instancia de contenedor. Una limitación del uso de un sistema de archivos host para el almacenamiento de contenedores es que los datos solo están disponibles en una sola instancia de contenedor a la vez. Esto significa que los contenedores solo pueden ejecutarse en el host donde residen los datos. Por lo tanto, el uso de almacenamiento de host solo se recomienda en escenarios donde la replicación de datos se gestiona a nivel de aplicación.

Volúmenes de Amazon EFS

Amazon Elastic File System (Amazon EFS) ofrece un sistema de archivos NFS elástico simple, escalable y totalmente administrado. Está diseñado para poder escalar bajo demanda a petabytes sin interrumpir las aplicaciones. Puede escalar hacia dentro o hacia fuera a medida que agrega y elimina archivos.

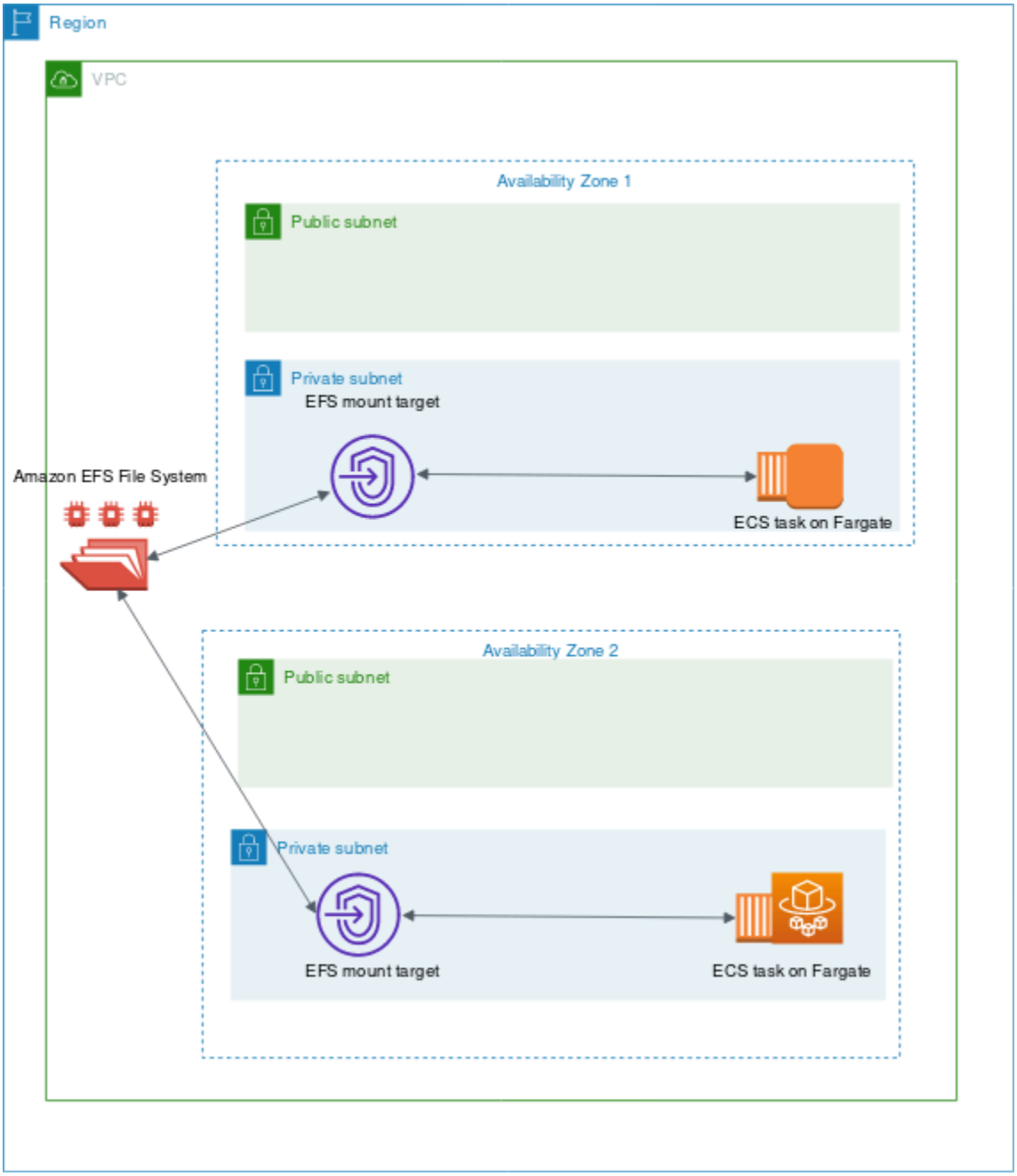
Puede ejecutar sus aplicaciones con estado en Amazon ECS utilizando volúmenes de Amazon EFS para proporcionar almacenamiento persistente. Tareas de Amazon ECS que se ejecutan en instancias de Amazon EC2 o en Fargate mediante la versión de plataforma`1.4.0`y posteriormente pueden montar un sistema de archivos de Amazon EFS existente. Dado que varios contenedores pueden montar y acceder a un sistema de archivos de Amazon EFS simultáneamente, sus tareas tienen acceso al mismo conjunto de datos independientemente de dónde estén alojadas.

Para montar un sistema de archivos de Amazon EFS en el contenedor, puede hacer referencia al sistema de archivos de Amazon EFS y el punto de montaje del contenedor en su definición de tarea de Amazon ECS. A continuación, se incluye un fragmento de una definición de tarea que utiliza Amazon EFS para el almacenamiento de contenedores.

```
...
```

```
"containerDefinitions":[
    {
        "mountPoints": [
            {
                "containerPath": "/opt/my-app",
                 "sourceVolume": "Shared-EFS-Volume"
            }
    }
  ]
...
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "transitEncryption": "DISABLED",
        "rootDirectory": ""
      },
      "name": "Shared-EFS-Volume"
    }
  ]
```

Amazon EFS almacena datos de forma redundante entre varias zonas de disponibilidad de una sola región. Una tarea de Amazon ECS monta el sistema de archivos de Amazon EFS utilizando un destino de montaje de Amazon EFS en su zona de disponibilidad. Una tarea de Amazon ECS sólo puede montar un sistema de archivos de Amazon EFS si el sistema de archivos de Amazon EFS tiene un destino de montaje en la zona de disponibilidad en la que se ejecuta la tarea. Por lo tanto, una práctica recomendada es crear destinos de montaje de Amazon EFS en todas las zonas de disponibilidad en las que planea alojar tareas de Amazon ECS.



Para obtener más información, consulteVolúmenes de Amazon EFSen laGuía del desarrollador de Amazon Elastic Container.

Controles de seguridad y acceso

Amazon EFS ofrece funciones de control de acceso que puede utilizar para garantizar que los datos almacenados en un sistema de archivos de Amazon EFS sean seguros y accesibles sólo desde aplicaciones que lo necesiten. Puede proteger los datos habilitando el cifrado en reposo y en tránsito.

Para obtener más información, consulteCifrado de datos en Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Además del cifrado de datos, también puede utilizar Amazon EFS para restringir el acceso a un sistema de archivos. Existen tres formas de implementar el control de acceso en EFS.

- Grupos de seguridad: con los destinos de montaje de Amazon EFS, puede configurar un grupo de seguridad que se utilice para permitir y denegar el tráfico de red. Puede configurar el grupo de seguridad adjunto a Amazon EFS para permitir el tráfico NFS (puerto 2049) del grupo de seguridad adjunto a sus instancias de Amazon ECS o, cuando utilice el`awsvpc`, la tarea Amazon ECS.
- IAM: puede restringir el acceso a un sistema de archivos de Amazon EFS mediante IAM. Cuando se configuran, las tareas de Amazon ECS requieren un rol de IAM para el acceso al sistema de archivos a fin de montar un sistema de archivos EFS. Para obtener más información, consulteUso de IAM para controlar el acceso a los datos del sistema de archivosen laGuía del usuario de Amazon Elastic File System.

  Las políticas de IAM también pueden imponer condiciones predefinidas como exigir a un cliente que utilice TLS al conectarse a un sistema de archivos de Amazon EFS. Para obtener más información, consulteClaves de condición Amazon EFS para clientesen laGuía del usuario de Amazon Elastic File System.
- Puntos de acceso de Amazon EFSLos puntos de acceso de Amazon EFS son puntos de entrada específicos de la aplicación en un sistema de archivos de Amazon EFS. Puede utilizar puntos de acceso para imponer una identidad de usuario, incluidos los grupos POSIX del usuario, para todas las solicitudes del sistema de archivos que se realizan a través del punto de acceso. Los puntos de acceso también pueden imponer un directorio raíz diferente para el sistema de archivos. Esto es para que los clientes solo puedan acceder a los datos del directorio especificado o de sus subdirectorios.

Considere la posibilidad de implementar los tres controles de acceso en un sistema de archivos de Amazon EFS para la máxima seguridad. Por ejemplo, puede configurar el grupo de seguridad adjunto a un punto de montaje de Amazon EFS para que sólo permita la entrada del tráfico NFS de un grupo de seguridad asociado a la instancia de contenedor o a la tarea de Amazon ECS. Además, puede configurar Amazon EFS para que requiera un rol de IAM para acceder al sistema de archivos, incluso si la conexión procede de un grupo de seguridad permitido. Por último, puede utilizar los puntos de acceso de Amazon EFS para exigir permisos de usuario POSIX y especificar directorios raíz para las aplicaciones.

En el siguiente fragmento de definición de tarea se muestra cómo montar un sistema de archivos de Amazon EFS mediante un punto de acceso.

```
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "authorizationConfig": {
          "acessPointId": "fsap-1234",
          "iam": "ENABLED"
        },
        "transitEncryption": "ENABLED",
        "rootDirectory": ""
      },
      "name": "my-filesystem"
    }
]
```

Performance

Amazon EFS ofrece dos modos de rendimiento: Propósito general y E/S máx. de uso general es adecuado para aplicaciones sensibles a la latencia como sistemas de gestión de contenido y herramientas CI/CD. Por el contrario, los sistemas de archivos de E/S Max son adecuados para cargas de trabajo como análisis de datos, procesamiento de medios y aprendizaje automático. Estas cargas de trabajo necesitan realizar operaciones paralelas desde cientos o incluso miles de contenedores y requieren el mayor rendimiento agregado posible y las IOPS. Para obtener más información, consulteModos de rendimiento de Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Algunas cargas de trabajo sensibles a la latencia requieren los niveles de E/S más altos proporcionados por el modo de desempeño de E/S máximo y la latencia más baja proporcionada por el modo de desempeño de uso general. Para este tipo de carga de trabajo, recomendamos crear varios sistemas de archivos en modo de desempeño de uso general. De este modo, puede distribuir la carga de trabajo de las aplicaciones entre todos estos sistemas de archivos, siempre que la carga de trabajo y las aplicaciones lo admitan.

Throughput

Todos los sistemas de archivos de Amazon EFS tienen un rendimiento medido asociado determinado por la cantidad de rendimiento aprovisionado para los sistemas de archivos que

utilizanRendimiento provisionadoo la cantidad de datos almacenados en la clase de almacenamiento estándar EFS o One Zone para sistemas de archivos que utilizanDesempeño por ráfaga de. Para obtener más información, consulteDescripción del rendimiento medidoen laGuía del usuario de Amazon Elastic File System.

El modo de rendimiento predeterminado para los sistemas de archivos de Amazon EFS es el modo de fragmentación. Con el modo de fragmentación, el rendimiento disponible para un sistema de archivos se amplía hacia dentro o hacia fuera a medida que crece un sistema de archivos. Dado que las cargas de trabajo basadas en archivos suelen generar picos, lo que requiere altos niveles de desempeño durante períodos de tiempo y bajos niveles de desempeño el resto del tiempo, Amazon EFS se ha diseñado para transmitir por ráfagas altos niveles de desempeño durante períodos de tiempo. Además, debido a que muchas cargas de trabajo son de lectura pesada, las operaciones de lectura se miden en una proporción de 1:3 respecto a otras operaciones NFS (como escritura).

Todos los sistemas de archivos de Amazon EFS ofrecen un rendimiento de línea de base coherente de 50 MB/s por cada TB de almacenamiento de Amazon EFS Standard o Amazon EFS One Zone. Todos los sistemas de archivos (independientemente del tamaño) pueden explotar hasta 100 MB/s. Los file systems con más de 1 TB de almacenamiento EFS Standard o EFS One Zone pueden expandir hasta 100 MB/s por cada TB. Debido a que las operaciones de lectura se miden en una proporción de 1:3, puede conducir hasta 300 MIBs/s por cada TiB de rendimiento de lectura. A medida que agrega datos al sistema de archivos, el rendimiento máximo disponible para el sistema de archivos se escala de forma lineal y automática con el almacenamiento en la clase de almacenamiento estándar de Amazon EFS. Si necesita más rendimiento del que puede lograr con la cantidad de datos almacenados, puede configurar el rendimiento aprovisionado en la cantidad específica que su carga de trabajo requiera.

El rendimiento del sistema de archivos se comparte en todas las instancias de Amazon EC2 conectadas a un sistema de archivos. Por ejemplo, un sistema de archivos de 1 TB que puede ráfagas a 100 MB/s de rendimiento puede conducir 100 MB/s desde una única instancia de Amazon EC2 puede cada unidad de 10 MB/s. Para obtener más información, consulteDesempeño de Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Cost optimization (Optimización de costos)

Amazon EFS simplifica el escalado del almacenamiento para usted. Los sistemas de archivos de Amazon EFS crecen automáticamente a medida que agrega más datos. Especialmente con Amazon EFSDesempeño por ráfaga deEn el modo de desempeño, el desempeño de Amazon EFS se escala a medida que aumenta el tamaño del sistema de archivos en la clase de almacenamiento estándar.

Para mejorar el rendimiento sin pagar un coste adicional por el rendimiento aprovisionado en un sistema de archivos EFS, puede compartir un sistema de archivos de Amazon EFS con varias aplicaciones. Mediante los puntos de acceso de Amazon EFS, puede implementar el aislamiento de almacenamiento en sistemas de archivos compartidos de Amazon EFS. Al hacerlo, aunque las aplicaciones sigan compartiendo el mismo sistema de archivos, no pueden acceder a los datos a menos que usted lo autorice.

A medida que aumentan los datos, Amazon EFS le ayuda a mover automáticamente los archivos a los que se accede con poca frecuencia a una clase de almacenamiento inferior. La clase de almacenamiento de acceso infrecuente estándar de Amazon EFS (IA) reduce los costos de almacenamiento de los archivos a los que no se tiene acceso todos los días. Esto lo hace sin que la alta disponibilidad, alta durabilidad, elasticidad y acceso al sistema de archivos POSIX que proporciona Amazon EFS se vean afectados. Para obtener más información, consulteClase de almacenamiento de Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Considere la posibilidad de utilizar políticas de ciclo de vida de Amazon EFS para ahorrar dinero automáticamente moviendo archivos a los que se accede con poca frecuencia al almacenamiento de Amazon EFS IA Para obtener más información, consulteGestión del ciclo de vida de Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Al crear un sistema de archivos de Amazon EFS, puede elegir si Amazon EFS replica los datos en varias zonas de disponibilidad (estándar) o almacena los datos de forma redundante en una única zona de disponibilidad. La clase de almacenamiento One Zone de Amazon EFS puede reducir los costes de almacenamiento en un margen significativo en comparación con las clases de almacenamiento estándar de Amazon EFS. Considere la posibilidad de utilizar la clase de almacenamiento One Zone de Amazon EFS para cargas de trabajo que no requieran resistencia Multi-AZ. Puede reducir aún más el coste del almacenamiento de una zona de Amazon EFS moviendo los archivos a los que se accede con poca frecuencia a Amazon EFS One Zone Infrequent Access. Para obtener más información, consulteAcceso poco frecuente de Amazon EFS.

Protección de los datos

Amazon EFS almacena los datos de forma redundante en varias zonas de disponibilidad para sistemas de archivos mediante clases de almacenamiento estándar. Si selecciona clases de almacenamiento de Amazon EFS One Zone, los datos se almacenan de forma redundante en una única zona de disponibilidad. Además, Amazon EFS está diseñado para proporcionar un 99,999999999% (11 9) de durabilidad durante un año determinado.

Al igual que con cualquier entorno, es una práctica recomendada tener una copia de seguridad y crear salvaguardias contra la eliminación accidental. En el caso de los datos de Amazon EFS, esa práctica recomendada incluye una copia de seguridad funcional y probada regularmente medianteAWS Backup. Los sistemas de archivos que utilizan clases de almacenamiento One Zone de Amazon EFS están configurados para realizar automáticamente copias de seguridad de los archivos de forma predeterminada en la creación del sistema de archivos, a menos que decida desactivar esta funcionalidad. Para obtener más información, consulteProtección de datos para Amazon EFSen laGuía del usuario de Amazon Elastic File System.

Casos de uso

Amazon EFS ofrece acceso compartido paralelo que aumenta y disminuye automáticamente a medida que se añaden y eliminan archivos. Como resultado, Amazon EFS es adecuado para cualquier aplicación que requiera un almacenamiento con funcionalidades como baja latencia, alto rendimiento y consistencia de lectura tras escritura. Amazon EFS es un back-end de almacenamiento ideal para aplicaciones que escalan horizontalmente y requieren un sistema de archivos compartido. Algunas cargas de trabajo como el análisis de datos, el procesamiento de contenido multimedia, la administración de contenido y el servicio web son algunos de los casos de uso comunes de Amazon EFS.

Un caso de uso en el que Amazon EFS podría no ser adecuado es para aplicaciones que requieren una latencia inferior a milisegundos. Esto es generalmente un requisito para los sistemas de bases de datos transaccionales. Recomendamos ejecutar pruebas de rendimiento del almacenamiento para determinar el impacto del uso de Amazon EFS para aplicaciones sensibles a la latencia. Si el rendimiento de las aplicaciones se degrada al utilizar Amazon EFS, considere Amazon EBS io2 Block Express, que proporciona latencia de E/S de baja varianza y submilisegundos en las instancias de Nitro. Para obtener más información, consulte Tipos de volumen de Amazon EBS en la Guía del usuario de Amazon EC2 para instancias de Linux.

Algunas aplicaciones fallan si su almacenamiento subyacente se cambia inesperadamente. Por lo tanto, Amazon EFS no es la mejor opción para estas aplicaciones. En su lugar, es posible que prefiera usar un sistema de almacenamiento que no permita el acceso simultáneo desde varios lugares.

Volúmenes de Docker

Los volúmenes Docker son una característica del tiempo de ejecución del contenedor Docker que permite que los contenedores conserven los datos mediante el montaje de un directorio

desde el sistema de archivos del host. Los controladores de volúmenes de Docker (denominados también «complementos») se usan para integrar volúmenes de contenedores con sistemas de almacenamiento externos como Amazon EBS. Los volúmenes de Docker solo se admiten cuando se alojan tareas de Amazon ECS en instancias de Amazon EC2.

Las tareas de Amazon ECS pueden utilizar volúmenes Docker para conservar los datos mediante volúmenes de Amazon EBS. Esto se realiza adjuntando un volumen de Amazon EBS a una instancia de Amazon EC2 y, a continuación, montando el volumen en una tarea mediante volúmenes Docker. Un volumen Docker se puede compartir entre varias tareas de Amazon ECS en el host.

La limitación de los volúmenes de Docker es que el sistema de archivos que utiliza la tarea está vinculado a la instancia específica de Amazon EC2. Si la instancia se detiene por cualquier motivo y la tarea se coloca en otra instancia, los datos se pierden. Puede asignar tareas a instancias para asegurarse de que los volúmenes de EBS asociados estén siempre disponibles para las tareas.

Para obtener más información, consulteVolúmenes de Dockeren laGuía del desarrollador de Amazon Elastic Container.

Ciclo de vida de los volúmenes de Amazon EBS

Existen dos patrones de uso clave con almacenamiento de contenedores y Amazon EBS. La primera es cuando una aplicación necesita conservar los datos y evitar la pérdida de datos cuando su contenedor termina. Un ejemplo de este tipo de aplicación sería una base de datos transaccional como MySQL. Cuando una tarea MySQL termina, se espera que otra tarea la reemplace. En este escenario, el ciclo de vida del volumen es independiente del ciclo de vida de la tarea. Cuando se utiliza EBS para conservar los datos del contenedor, se recomienda utilizar restricciones de ubicación de tareas para limitar la ubicación de la tarea a un único host con el volumen de EBS conectado.

El segundo es cuando el ciclo de vida del volumen es independiente del ciclo de vida de la tarea. Esto es especialmente útil para aplicaciones que requieren almacenamiento de alto rendimiento y baja latencia, pero no necesitan conservar los datos una vez finalizada la tarea. Por ejemplo, una carga de trabajo ETL que procesa grandes volúmenes de datos puede requerir un almacenamiento de alto rendimiento. Amazon EBS es adecuado para este tipo de carga de trabajo, ya que proporciona volúmenes de alto rendimiento que proporcionan hasta 256.000 IOPS. Cuando finaliza la tarea, la réplica de reemplazo se puede colocar de forma segura en cualquier host de Amazon EC2 del clúster. Siempre que la tarea tenga acceso a un back-end de almacenamiento que pueda cumplir sus requisitos de rendimiento, la tarea puede realizar su función. Por lo tanto, no se necesitan restricciones de ubicación de tareas en este caso.

Si las instancias de Amazon EC2 del clúster tienen varios tipos de volúmenes de Amazon EBS asociados a ellas, puede utilizar restricciones de ubicación de tareas para asegurarse de que las tareas se coloquen en instancias con un volumen adecuado de Amazon EBS adjunto. Por ejemplo, supongamos que un clúster tiene algunas instancias con ungp2volumen, mientras que otros usanio1Volúmenes de. Puede adjuntar atributos personalizados a instancias conio1y, a continuación, utilizar restricciones de ubicación de tareas para garantizar que las tareas intensivas de E/S se coloquen siempre en instancias de contenedor conio1Volúmenes de.

Los siguientes ejemplos deAWS CLIpara colocar atributos en una instancia de contenedor de Amazon ECS.

```
aws ecs put-attributes \
     --attributes name=EBS,value=io1,targetId=<your-container-instance-arn>
```

Disponibilidad de datos de Amazon EBS

Los contenedores suelen ser de corta duración, se crean con frecuencia y terminan a medida que las aplicaciones se escalan horizontalmente. Como práctica recomendada, puede ejecutar cargas de trabajo en varias zonas de disponibilidad para mejorar la disponibilidad de las aplicaciones. Amazon ECS proporciona una forma de controlar la ubicación de tareas mediante estrategias de ubicación de tareas y restricciones de ubicación de tareas. Cuando una carga de trabajo mantiene sus datos utilizando volúmenes de Amazon EBS, sus tareas deben colocarse en la misma zona de disponibilidad que el volumen de Amazon EBS. También se recomienda establecer una delimitación de ubicación que limite la zona de disponibilidad en la que se puede colocar una tarea. De este modo, se garantiza que las tareas y sus volúmenes correspondientes se encuentren siempre en la misma zona de disponibilidad.

Al ejecutar tareas independientes, puede controlar qué zona de disponibilidad se coloca la tarea estableciendo restricciones de ubicación mediante el atributo de zona de disponibilidad.

```
attribute:ecs.availability-zone == us-east-1a
```

Al ejecutar aplicaciones que podrían beneficiarse de la ejecución en varias zonas de disponibilidad, considere la posibilidad de crear un servicio Amazon ECS diferente para cada zona de disponibilidad. Esto garantiza que las tareas que necesitan un volumen de Amazon EBS se coloquen siempre en la misma zona de disponibilidad que el volumen asociado.

Recomendamos crear instancias de contenedor en cada zona de disponibilidad, adjuntando volúmenes de Amazon EBS mediantelniciar plantillasy añadiendoAtributos personalizadosa las

instancias para diferenciarlas de otras instancias de contenedor del clúster de Amazon ECS. Al crear servicios, configure las restricciones de ubicación de tareas para asegurarse de que Amazon ECS coloca las tareas en la zona de disponibilidad y la instancia adecuadas. Para obtener más información, consulteEjemplos de delimitación de ubicación de tareasen laGuía del desarrollador de Amazon Elastic Container.

Complementos de volumen de Docker

Los complementos Docker como Portworx proporcionan una abstracción entre el volumen Docker y el volumen de Amazon EBS. Estos complementos pueden crear dinámicamente un volumen de Amazon EBS cuando se inicia la tarea que necesita un volumen. Portworx también puede adjuntar un volumen a un nuevo host cuando un contenedor termina, y su réplica posterior se coloca en una instancia de contenedor diferente. También replica los datos de volumen de cada contenedor entre los nodos de Amazon ECS y entre las zonas de disponibilidad. Para obtener más información, consultePortworx.

Amazon FSx para el servidor de archivos de Windows

Amazon FSx for Windows File Server proporciona almacenamiento de archivos totalmente gestionado, altamente fiable y escalable al que se puede acceder a través del protocolo de bloque de mensajes de servidor (SMB) estándar del sector. Está construido en Windows Server y ofrece una amplia gama de características administrativas, como cuotas de usuario, restauración de archivos de usuario final e integración con Microsoft Active Directory (AD). Ofrece opciones de implementación Single-AZ y Multi-AZ, copias de seguridad totalmente administradas y cifrado de datos en reposo y en tránsito.

Amazon ECS admite el uso de Amazon FSx for Windows File Server en las definiciones de tareas de Amazon ECS Windows que permiten el almacenamiento persistente como punto de montaje a través del protocolo SMBv3 mediante una función SMB denominada GlobalMappings.

Para configurar la integración de Amazon FSx for Windows File Server y Amazon ECS, la instancia contenedor de Windows debe ser un miembro de dominio en un servicio de dominio de Active Directory (AD DS), alojado en unAWS Directory Service for Microsoft Active Directory, Active Directory local o Active Directory alojado en Amazon EC2.AWS Secrets Managerse utiliza para almacenar datos confidenciales como el nombre de usuario y la contraseña de una credencial de Active Directory que se utiliza para asignar el recurso compartido en la instancia contenedor de Windows.

Para utilizar volúmenes de sistema de archivos de Amazon FSx for Windows File Server para sus contenedores, debe especificar las configuraciones del volumen y el punto de montaje en su definición de tarea. A continuación, se incluye un fragmento de una definición de tarea que utiliza Amazon FSx for Windows File Server para el almacenamiento del contenedor.

```
{
 "containerDefinitions": [{
  "name": "container-using-fsx",
  "image": "iis:2",
  "entryPoint": [
   "powershell",
   "-command"
  ],
  "mountPoints": [{
   "sourceVolume": "myFsxVolume",
   "containerPath": "\\mount\\fsx",
   "readOnly": false
  }]
 }],
 "volumes": [{
  "fsxWindowsFileServerVolumeConfiguration": {
   "fileSystemId": "fs-ID",
   "authorizationConfig": {
    "domain": "ADDOMAIN.local",
    "credentialsParameter": "arn:aws:secretsmanager:us-
east-1:111122223333:secret:SecretName"
   },
   "rootDirectory": "share"
  }
 }]
}
```

Para obtener más información, consulteVolúmenes de Amazon FSx for Windows File Serveren laGuía del desarrollador de Amazon Elastic Container.

Controles de seguridad y acceso

Amazon FSx for Windows File Server ofrece las siguientes funciones de control de acceso que puede utilizar para garantizar que los datos almacenados en un sistema de archivos de Amazon FSx for Windows File Server solo se puede acceder a ellos desde las aplicaciones que lo necesitan.

Cifrado de datos

Amazon FSx for Windows File Server admite dos formas de cifrado para sistemas de archivos. Son el cifrado de datos en tránsito y en reposo. El cifrado de datos en tránsito es compatible con los recursos compartidos de archivos asignados en una instancia de contenedor compatible con el protocolo SMB 3.0 o posterior. El cifrado de los datos en reposo se activa automáticamente al crear un sistema de archivos de Amazon FSx. Amazon FSx cifra automáticamente los datos en tránsito mediante el cifrado SMB a medida que accede a su sistema de archivos sin necesidad de modificar sus aplicaciones. Para obtener más información, consulteCifrado de datos en Amazon FSxen laGuía del usuario de Amazon FSx for Windows File Server.

Control de acceso a nivel de carpeta mediante ACL de Windows

La instancia de Windows Amazon EC2 tiene acceso a recursos compartidos de archivos de Amazon FSx mediante credenciales de Active Directory. Utiliza listas de control de acceso (ACL) estándar de Windows para controlar el acceso a nivel de archivos y carpetas. Puede crear varias credenciales, cada una para una carpeta específica dentro del recurso compartido que se asigna a una tarea específica.

En el siguiente ejemplo, la tarea tiene acceso a la carpetaApp01mediante una credencial guardada en el Secrets Manager. Su nombre de recurso de Amazon (ARN) es1234.

```
"rootDirectory": "\\path\\to\\my\\data\App01",
"credentialsParameter": "arn-1234",
"domain": "corp.fullyqualified.com",
```

En otro ejemplo, una tarea tiene acceso a la carpetaApp02mediante una credencial guardada en el Administrador de Secretos. Su ARN es 6789.

```
"rootDirectory": "\\path\\to\\my\\data\App02",
"credentialsParameter": "arn-6789",
"domain": "corp.fullyqualified.com",
```

Casos de uso

Los contenedores no están diseñados para persistir los datos. Sin embargo, algunas aplicaciones de.NET contenedorizadas pueden requerir carpetas locales como almacenamiento persistente para guardar los resultados de las aplicaciones. Amazon FSx for Windows File Server ofrece una carpeta

local en el contenedor. Esto permite que varios contenedores lean y escriban en el mismo sistema de archivos respaldado por un recurso compartido SMB.

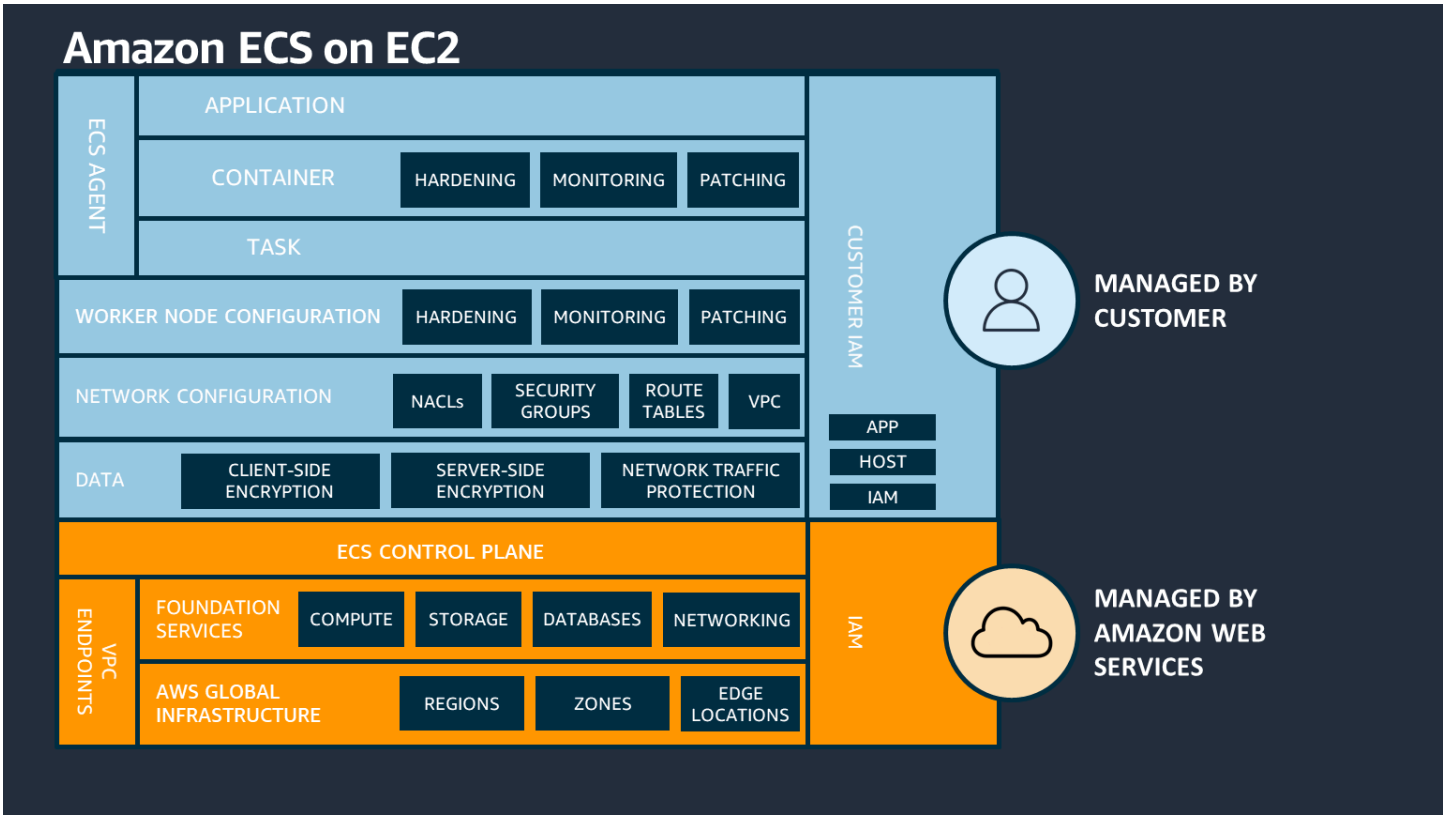
Prácticas recomendadas: seguridad

Esta guía proporciona recomendaciones de seguridad y cumplimiento para proteger su información, sistemas y otros activos que dependen de Amazon ECS. También presenta algunas evaluaciones de riesgos y estrategias de mitigación que puede utilizar para controlar mejor los controles de seguridad creados para los clústeres de Amazon ECS y las cargas de trabajo que admiten. Cada tema de esta guía comienza con una breve descripción general, seguida de una lista de recomendaciones y prácticas recomendadas que puede utilizar para proteger sus clústeres de Amazon ECS.

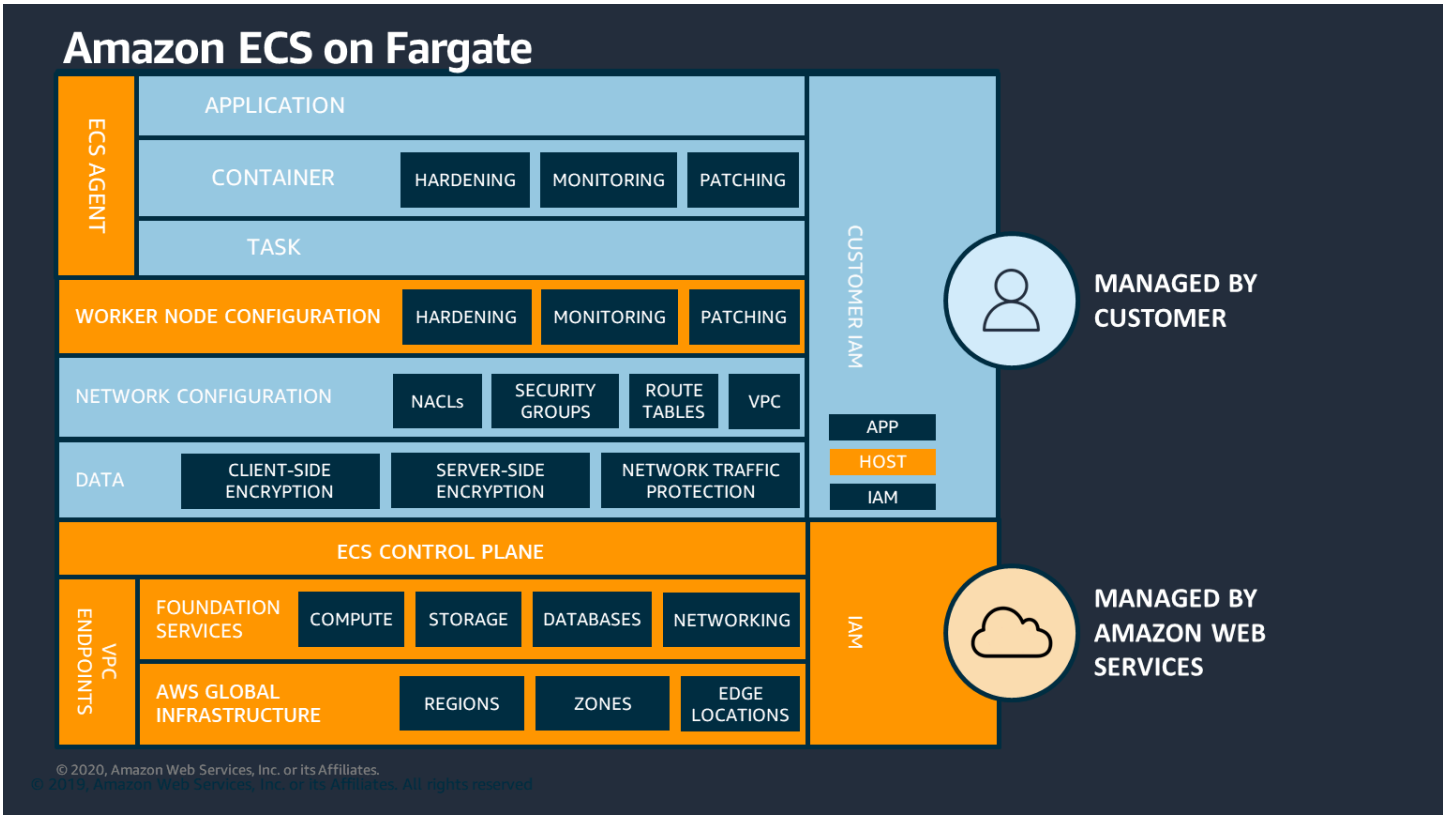
Temas

Modelo de responsabilidad compartida

La seguridad y el cumplimiento de un servicio gestionado como Amazon ECS es una reposibilidad compartida de usted yAWS. En general,AWSes responsable de la seguridad «de» la nube, mientras que usted, el cliente, es responsable de la seguridad «en» la nube.AWSes responsable de la gestión del plano de control de Amazon ECS, incluida la infraestructura necesaria para ofrecer un servicio seguro y fiable. Además, usted es en gran parte responsable de los temas de esta guía. Esto incluye la seguridad de datos, redes y tiempo de ejecución, así como el registro y la supervisión.



Con respecto a la seguridad de la infraestructura,AWSasume más responsabilidad porAWS Fargateque para otras instancias autogestionadas. Con Fargate,AWSadministra la seguridad de la instancia subyacente en la nube y el tiempo de ejecución que se utiliza para ejecutar sus tareas. Fargate también escala automáticamente su infraestructura en su nombre.



Antes de extender sus servicios a la nube, debe comprender qué aspectos de seguridad y cumplimiento de los que es responsable.

Para obtener más información sobre el modelo de responsabilidad compartida, consulteModelo de responsabilidad compartida.

AWS Identity and Access Management

Puede usarAWS Identity and Access Management(IAM) para administrar y controlar el acceso aAWSservicios y recursos mediante políticas basadas en reglas con fines de autenticación y autorización. Concretamente, a través de este servicio, controla el acceso aAWSmediante políticas que se aplican a los usuarios, grupos o roles de IAM. Entre estos tres, los usuarios de IAM son cuentas que pueden tener acceso a sus recursos. Además, un rol de IAM es un conjunto de permisos que puede asumir una identidad autenticada, que no está asociada con una identidad particular fuera de IAM. Para obtener más información, consulte¿Políticas y permisos en IAM?.

Gestión del acceso a Amazon ECS

Puede controlar el acceso a Amazon ECS creando y aplicando políticas de IAM. Estas políticas se componen de un conjunto de acciones que se aplican a un conjunto específico de recursos. La

acción de una política define la lista de operaciones (como las API de Amazon ECS) permitidas o denegadas, mientras que el recurso controla cuáles son los objetos de Amazon ECS a los que se aplica la acción. Se pueden agregar condiciones a una política para restringir su alcance. Por ejemplo, una directiva se puede escribir para permitir que sólo se realice una acción contra tareas con un conjunto determinado de etiquetas. Para obtener más información, consulteCómo funciona Amazon ECS con IAMen laGuía del desarrollador de Amazon Elastic Container.

Recommendations

Le recomendamos que realice las siguientes acciones al configurar las funciones y políticas de IAM.

Siga la política de acceso menos privilegiado

Cree directivas que tengan un ámbito para permitir a los usuarios realizar sus trabajos prescritos. Por ejemplo, si un desarrollador necesita detener periódicamente una tarea, cree una directiva que solo permita esa acción concreta. En el ejemplo siguiente sólo se permite a un usuario detener una tarea que pertenece a un determinado`task_family`En un clúster con un nombre de recurso de Amazon (ARN) específico. Hacer referencia a un ARN en una condición también es un ejemplo del uso de permisos de nivel de recursos. Puede utilizar los permisos de nivel de recurso para especificar el recurso al que desea aplicar una acción.

> **Note**
>
> Cuando haga referencia a un ARN en una política, utilice el nuevo formato ARN más largo. Para obtener más información, consulteNombres de recursos de Amazon (ARN) e IDen laGuía del desarrollador de Amazon Elastic Container.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:StopTask"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"
        }
```

```
      },
      "Resource": [
        "arn:aws:ecs:<region>:<aws_account_id>:task-definition/<task_family>:*"
      ]
    }
  ]
}
```

Deje que el recurso del clúster sirva como límite administrativo

Las políticas que tienen un ámbito demasiado estrecho pueden provocar una proliferación de roles y aumentar la sobrecarga administrativa. En lugar de crear roles que sólo tengan un ámbito de tareas o servicios concretos, cree roles que se asignen a clústeres y utilice el clúster como límite administrativo principal.

Aísle a los usuarios finales de la API de Amazon ECS mediante la creación de canalizaciones automatizadas

Puede limitar las acciones que los usuarios pueden utilizar creando canalizaciones que empaqueten e implementen aplicaciones automáticamente en clústeres de Amazon ECS. Esto delega efectivamente el trabajo de crear, actualizar y eliminar tareas en la canalización. Para obtener más información, consulteTutorial: Implementación estándar de Amazon ECS con CodePipelineen laAWS CodePipelineGuía del usuario.

Utilizar condiciones de política para una capa de seguridad adicional

Cuando necesite una capa de seguridad adicional, agregue una condición a su directiva. Esto puede ser útil si está realizando una operación con privilegios o cuando necesita restringir el conjunto de acciones que se pueden realizar con recursos concretos. La siguiente directiva de ejemplo requiere autorización de varios factores al eliminar un clúster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeleteCluster"
      ],
      "Condition": {
```

```
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      },
    "Resource": ["*"]
    }
  ]
}
```

Las etiquetas aplicadas a los servicios se propagan a todas las tareas que forman parte de ese servicio. Debido a esto, puede crear roles con el ámbito de los recursos de Amazon ECS con etiquetas específicas. En la siguiente directiva, una entidad de IAM inicia y detiene todas las tareas con una clave de etiqueta de`Department`y un valor de etiqueta de`Accounting`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:RunTask"
            ],
            "Resource": "arn:aws:ecs:*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Department": "Accounting"}
            }
        }
    ]
}
```

Auditoría periódica del acceso a las API de Amazon ECS

Un usuario puede cambiar las funciones. Después de cambiar las funciones, es posible que los permisos que se les concedieron anteriormente ya no se apliquen. Asegúrese de auditar quién tiene acceso a las API de Amazon ECS y si ese acceso todavía está garantizado. Considere la posibilidad de integrar IAM con una solución de administración del ciclo de vida del usuario que revoque automáticamente el acceso cuando un usuario abandona la organización. Para obtener más información, consulteDirectivas de auditoría de seguridad de Amazon ECSen laReferencia general de Amazon Web Services.

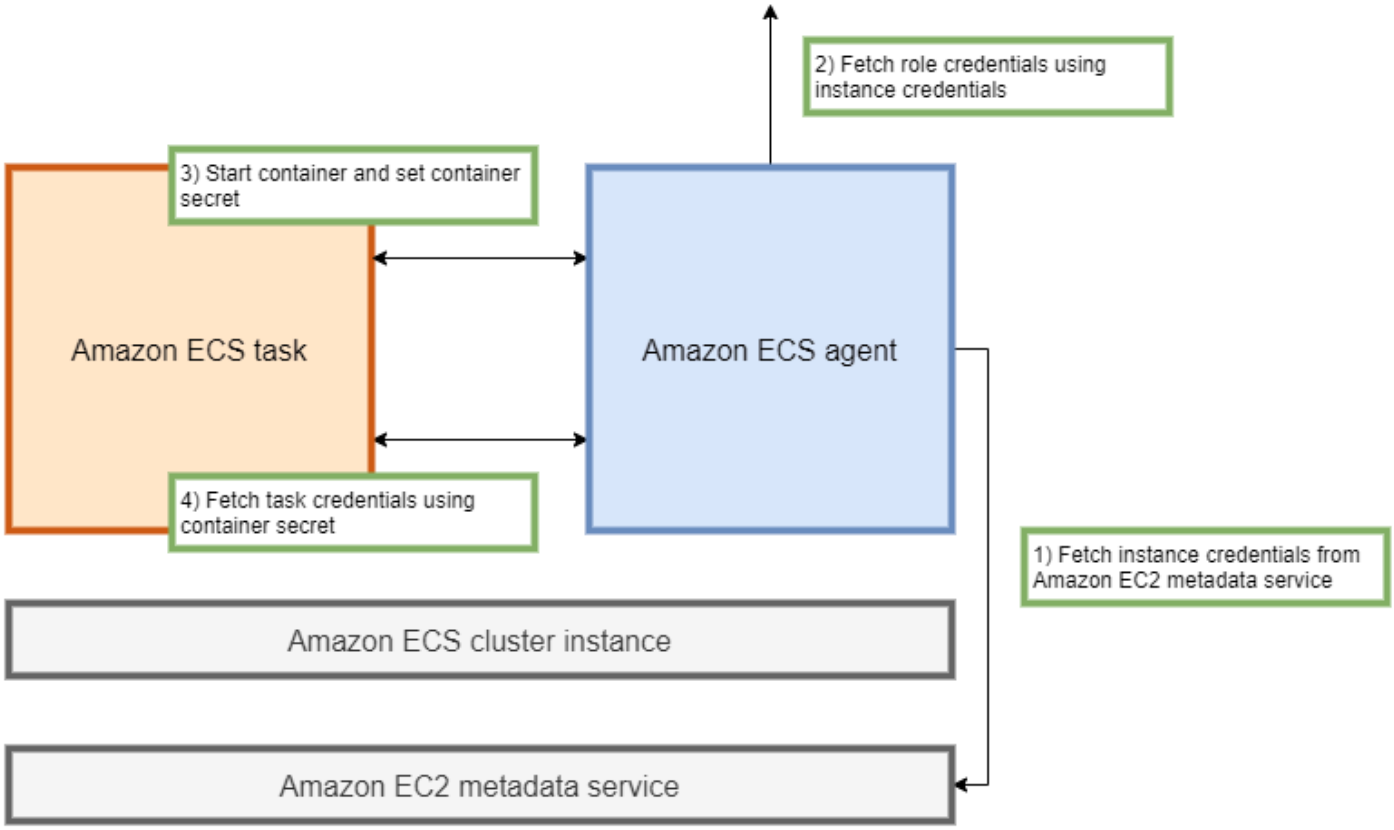
Uso de funciones de IAM con tareas de Amazon ECS

Le recomendamos que asigne a una tarea un rol de IAM. Su función se puede distinguir de la función de la instancia de Amazon EC2 en la que se ejecuta. La asignación de un rol a cada tarea se alinea con el principio de acceso menos privilegiado y permite un mayor control granular sobre acciones y recursos.

Al asignar roles de IAM a una tarea, debe utilizar la siguiente directiva de confianza para que cada una de las tareas pueda asumir un rol de IAM distinto del que utiliza la instancia de EC2. De esta forma, la tarea no hereda el rol de la instancia de EC2.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Cuando agrega un rol de tarea a una definición de tarea, el agente contenedor de Amazon ECS crea automáticamente un token con un identificador de credencial único (por ejemplo,`12345678-90ab-cdef-1234-567890abcdef`) para la tarea. Este token y las credenciales de rol se agregan a la caché interna del agente. El agente rellena la variable de entorno`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`en el contenedor con el URI del ID de credencial (por ejemplo,`/v2/credentials/12345678-90ab-cdef-1234-567890abcdef`).



Puede recuperar manualmente las credenciales de rol temporales desde el interior de un contenedor añadiendo la variable de entorno a la dirección IP del agente de contenedor de Amazon ECS y ejecutando el comando`curl`en la cadena resultante.

```
curl 192.0.2.0$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

El resultado esperado es el siguiente:

```
{
 "RoleArn": "arn:aws:iam::123456789012:role/SSMTaskRole-SSMFargateTaskIAMRole-
DASWWSF2WGD6",
 "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
 "Token": "IQoJb3JpZ2luX2VjEEM/Example==",
 "Expiration": "2021-01-16T00:51:53Z"
}
```

Las versiones más recientes de laAWSLos SDK obtienen automáticamente estas credenciales de la`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`variable al hacerAWSLlamadas al API.

El resultado incluye un par de claves de acceso que consta de un ID de clave de acceso secreta y una clave secreta que la aplicación utiliza para acceder aAWSde AWS. También incluye un token queAWSutiliza para comprobar que las credenciales son válidas. De forma predeterminada, las credenciales asignadas a tareas que utilizan roles de tarea son válidas durante seis horas. A continuación, el agente de contenedor de Amazon ECS los rota de forma automática.

Rol de ejecución de tareas

La función de ejecución de tareas se utiliza para conceder permiso al agente contenedor de Amazon ECS para llamar aAWSAcciones de la API en su nombre. Por ejemplo, cuando se utilizaAWS Fargate, Fargate necesita un rol de IAM que le permita extraer imágenes de Amazon ECR y escribir registros en CloudWatch Logs. También se requiere un rol de IAM cuando una tarea hace referencia a un secreto almacenado enAWS Secrets Manager, como un secreto de extracción de imagen.

> **Note**
>
> Si extraes imágenes como usuario autenticado, es menos probable que se vea afectado por los cambios que se produjeron enLímites de velocidad de extracción de Docker Hub. Para obtener más información, consulteAutenticación de registros privados para instancias de. Al utilizar Amazon ECR y Amazon ECR Public, puede evitar los límites impuestos por Docker. Si extrae imágenes de Amazon ECR, esto también ayuda a acortar los tiempos de extracción de la red y reduce los cambios en la transferencia de datos cuando el tráfico sale de la VPC.

> **Important**
>
> Cuando utilice Fargate, debe autenticarse en un registro de imágenes privado usando`repositoryCredentials`. No es posible establecer las variables de entorno del agente contenedor de Amazon ECS`ECS_ENGINE_AUTH_TYPE`or`ECS_ENGINE_AUTH_DATA`o modifique la`ecs.config`para las tareas alojadas en Fargate. Para obtener más información, consulteAutenticación privada del Registro para tareas.

Función de instancia de contenedor de Amazon EC2

El agente contenedor de Amazon ECS es un contenedor que se ejecuta en cada instancia de Amazon EC2 de un clúster de Amazon ECS. Se inicializa fuera de Amazon ECS mediante

elinitque está disponible en el sistema operativo. En consecuencia, no se pueden conceder permisos a través de un rol de tarea. En su lugar, los permisos deben asignarse a las instancias de Amazon EC2 en las que se ejecutan los agentes. La lista de acciones en el ejemploAmazonEC2ContainerServiceforEC2Roledebe otorgarse a laecsInstanceRole. Si no lo hace, las instancias no pueden unirse al clúster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTags",
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:UpdateContainerInstancesState",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

En esta política, elecrylogspermiten a los contenedores que se ejecutan en las instancias extraer imágenes de Amazon ECR y escribir registros en Amazon CloudWatch. Laecspermiten al agente registrar y anular el registro de instancias y comunicarse con el plano de control de Amazon ECS. De éstos, elecs:CreateClusteres opcional.

Roles vinculados a servicios

Puede utilizar la función vinculada a servicios de Amazon ECS para conceder permiso al servicio de Amazon ECS para llamar a otras API de servicio en su nombre. Amazon ECS necesita los permisos para crear y eliminar interfaces de red, registrar y anular el registro de destinos con un grupo de destino. También necesita los permisos necesarios para crear y eliminar políticas de escala de. Estos permisos se conceden a través del rol vinculado al servicio. Este rol se crea en su nombre la primera vez que utiliza el servicio.

> **Note**
>
> Si elimina sin querer el rol vinculado al servicio, puede volver a crearlo. Para obtener instrucciones, consulteCreación del rol vinculado a servicio.

Recommendations

Le recomendamos que haga lo siguiente al configurar las funciones y políticas de IAM de la tarea.

Bloquear el acceso a los metadatos de Amazon EC2

Cuando ejecute sus tareas en instancias de Amazon EC2, le recomendamos encarecidamente que bloquee el acceso a los metadatos de Amazon EC2 para evitar que los contenedores hereden la función asignada a dichas instancias. Si sus aplicaciones tienen que llamar a unAWS, utilice las funciones de IAM para las tareas.

Para evitar que las tareas se ejecuten enpuentepara acceder a los metadatos de Amazon EC2, ejecute el siguiente comando o actualice los datos de usuario de la instancia. Para obtener más instrucciones sobre cómo actualizar los datos de usuario de una instancia, consulte esteAWSArtículo de Support de. Para obtener más información sobre el modo de puente de definición de tareas, consultemodo de red de definición de tarea.

```
sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface
 docker+ --destination 192.0.2.0/32 --jump DROP
```

Para que este cambio persista después de un reinicio, ejecute el siguiente comando específico para su imagen de máquina de Amazon (AMI):

- Amazon Linux 2

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now
 iptables
```

- Amazon Linux

```
sudo service iptables save
```

Para las tareas que utilizanawsvpcmodo de red, establezca la variable de entornoECS_AWSVPC_BLOCK_IMDSDe atrueen la/etc/ecs/ecs.configfile.

Debe establecer elECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOSTvariable afalseen el archivo de configuración de ecs-agent para evitar que los contenedores que se ejecutan dentro delhostpara acceder a los metadatos de Amazon EC2.

Usar**awsvpc**Modo de red

Utilizar la redawsvpcpara restringir el flujo de tráfico entre diferentes tareas o entre sus tareas y otros servicios que se ejecutan dentro de su Amazon VPC. Esto añade una capa de seguridad adicional. Laawsvpcproporciona aislamiento de red a nivel de tarea para las tareas que se ejecutan en Amazon EC2. Es el modo predeterminado enAWS Fargate. Es el único modo de red que puede utilizar para asignar un grupo de seguridad a las tareas.

Utilizar el Asesor de acceso de IAM para refinar roles

Te recomendamos que elimines todas las acciones que nunca se hayan usado o que no se hayan utilizado durante algún tiempo. Esto evita que se produzcan accesos no deseados. Para ello, revise los resultados producidos por el Asesor de acceso de IAM y, a continuación, quite las acciones que nunca se usaron o que no se han utilizado recientemente. Para ello, siga los siguientes pasos.

Ejecute el siguiente comando para generar un informe que muestre la última información de acceso para la directiva a la que se hace referencia:

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

Utilice laJobIdque estaba en la salida para ejecutar el siguiente comando. Cuando haya terminado de hacer esto, puede ver los resultados del informe.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Para obtener más información, consulteAsesor de acceso de IAM.

MonitorizaciónAWS CloudTrailpor actividad sospechosa

Puede monitorearAWS CloudTrailpor cualquier actividad sospechosa. La mayoría deAWSLas llamadas a la API se registran enAWS CloudTrailComo acontecimientos. Son analizados porAWS CloudTrailPerspectivas, y se le avisa de cualquier comportamiento sospechoso asociado con`write`Llamadas al API. Esto podría incluir un pico en el volumen de llamadas. Estas alertas incluyen información como el momento en que se produjo la actividad inusual y el ARN de identidad principal que contribuyó a las API.

Puede identificar las acciones que realizan tareas con un rol de IAM enAWS CloudTrailmirando el`userIdentity`Propiedad. En el siguiente ejemplo, el`arn`incluye el nombre de la función asumida,`s3-write-go-bucket-role`, seguido del nombre de la tarea,`7e9894e088ad416eb5cab92afExample`.

```
"userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROA36C6WWEJ2YEXAMPLE:7e9894e088ad416eb5cab92afExample",
    "arn": "arn:aws:sts::123456789012:assumed-role/s3-write-go-bucket-
role/7e9894e088ad416eb5cab92afExample",
    ...
}
```

> **Note**
>
> Cuando las tareas que asumen un rol se ejecutan en instancias de contenedor de Amazon EC2, el agente contenedor de Amazon ECS registra una solicitud en el registro de auditoría del agente que se encuentra en una dirección en el`/var/log/ecs/audit.log.YYYY-MM-DD-HH`formato. Para obtener más información, consulteRegistro de funciones de IAM de tareasyRegistro de eventos de Insights para registros de seguimiento.

Seguridad de la red

La seguridad de red es un tema amplio que abarca varios subtemas. Estos incluyen cifrado en tránsito, segmentación y aislamiento de red, cortafuegos, enrutamiento de tráfico y observabilidad.

Cifrado en tránsito

El cifrado del tráfico de red impide que los usuarios no autorizados intercepten y lean datos cuando esos datos se transmiten a través de una red. Con Amazon ECS, el cifrado de red se puede implementar de cualquiera de las siguientes formas.

- Con una malla de servicio (TLS):

  conAWS App Mesh, puede configurar conexiones TLS entre los proxies de Envoy que se implementan con extremos de malla. Dos ejemplos son los nodos virtuales y las puertas de enlace virtuales. Los certificados TLS pueden provenir deAWS Certificate Manager(ACM). O bien, puede provenir de su propia autoridad de certificación privada.

  - Habilitar seguridad de la capa de transporte (TLS)
  - Habilite el cifrado de tráfico entre servicios enAWS App Meshusar certificados ACM o certificados proporcionados por el cliente
  - Tutorial de TLS ACM
  - Tutorial del archivo TLS
  - Envoy
- Uso de instancias Nitro:

  De forma predeterminada, el tráfico se cifra automáticamente entre los siguientes tipos de instancias de Nitro: C5n, C5dn, M5dn, M5dn, M5dn, R5dn y R5dn. El tráfico no se cifra cuando se enruta a través de una puerta de enlace de tránsito, un equilibrador de carga o un intermediario similar.

  - Cifrado en tránsito
  - Novedades de la contabilidad de 2019
  - Esta charla de Re:Inforce 2019
- Utilizar la indicación de nombre de servidor (SNI) con un Application Load Balancer:

  El Application Load Balancer (ALB) y el Network Load Balancer (NLB) admiten la indicación de nombre de servidor (SNI). Mediante el uso de SNI, puede colocar varias aplicaciones seguras detrás de un solo oyente. Para esto, cada uno tiene su propio certificado TLS. Le recomendamos que aprovisione certificados para el balanacer de carga usandoAWS Certificate Manager(ACM) y, a continuación, agréguelos a la lista de certificados del agente de escucha. LaAWSEl balanceador de carga utiliza un algoritmo de selección de certificados inteligentes con SNI. Si el nombre de host proporcionado por un cliente coincide con un solo certificado de la lista de certificados, el

balanceador de carga elige dicho certificado. Si un nombre de host proporcionado por un cliente coincide con varios certificados de la lista, el balanceador de carga selecciona un certificado que el cliente puede admitir. Algunos ejemplos incluyen un certificado autofirmado o un certificado generado a través de ACM.

- SNI con Application Load Balancer
- SNI con Network Load Balancer

- Cifrado integral con certificados TLS:

  Esto implica implementar un certificado TLS con la tarea. Esto puede ser un certificado autofirmado o un certificado de una autoridad de certificación de confianza. Puede obtener el certificado haciendo referencia a un secreto para el certificado. De lo contrario, puede optar por ejecutar un contenedor que emita una solicitud de firma de certificados (CSR) a ACM y, a continuación, monta el secreto resultante en un volumen compartido.

  - Mantener la seguridad de la capa de transporte hasta los contenedores mediante el Network Load Balancer con Amazon ECS, parte 1
  - Mantenimiento de la seguridad de la capa de transporte (TLS) hasta la parte 2 del contenedor: Uso de AWS Private Certificate Authority

Integración en red de las tareas

Las siguientes recomendaciones están teniendo en cuenta el funcionamiento de Amazon ECS. Amazon ECS no utiliza una red superpuesta. En su lugar, las tareas se configuran para funcionar en diferentes modos de red. Por ejemplo, las tareas que están configuradas para usar`bridge`adquirir una dirección IP no enrutable de una red Docker que se ejecuta en cada host. Las tareas que están configuradas para usar el`awsvpc`El modo de red adquiere una dirección IP de la subred del host. Tareas configuradas con`host`utilizar la interfaz de red del host.`awsvpc`es el modo de red preferido. Esto se debe a que es el único modo que puede usar para asignar grupos de seguridad a tareas. También es el único modo que está disponible paraAWS Fargateen Amazon ECS.

Grupos de seguridad de las tareas

Le recomendamos que configure sus tareas para que utilicen la`awsvpc`Modo de red. Después de configurar la tarea para que utilice este modo, el agente de Amazon ECS aprovisiona y conecta automáticamente una interfaz de red elástica (ENI) a la tarea. Cuando se aprovisiona el ENI, la tarea se inscribe en unAWSGrupo de seguridad. El grupo de seguridad funciona como un firewall virtual que puede utilizar para controlar el tráfico entrante y saliente.

Malla de servicio y seguridad mutua de la capa de transporte (MTL)

Puede utilizar una malla de servicio comoAWS App Meshpara controlar el tráfico de red. De forma predeterminada, un nodo virtual solo puede comunicarse con sus backends de servicio configurados, como los servicios virtuales con los que se comunicará el nodo virtual. Si un nodo virtual necesita comunicarse con un servicio fuera de la malla, puede usar la herramientaALLOW_ALLo creando un nodo virtual dentro de la malla para el servicio externo. Para obtener más información, consulteKubernetes Tutorial para la salida.

App Mesh también le ofrece la posibilidad de utilizar Mutual Transport Layer Security (MTL) donde tanto el cliente como el servidor se autentican mutuamente mediante certificados. La comunicación subsiguiente entre el cliente y el servidor se cifran mediante TLS. Al requerir MTL entre servicios en una malla, puede verificar que el tráfico proviene de una fuente de confianza. Para obtener más información, consulte los siguientes temas:

- Autenticación MTL
- Tutorial de MTLs Secret Discovery Service (SDS)
- Tutorial de archivos de MTLS

AWS PrivateLink

AWS PrivateLinkes una tecnología de red que le permite crear endpoints privados para diferentesAWS, incluido Amazon ECS. Los endpoints son necesarios en entornos aislados donde no hay Internet Gateway (IGW) conectado a Amazon VPC y no hay rutas alternativas a Internet. UsoAWS PrivateLinkgarantiza que las llamadas al servicio Amazon ECS permanezcan dentro de la VPC de Amazon y no atraviesan Internet. Para obtener instrucciones sobre cómo crearAWS PrivateLinkpara Amazon ECS y otros servicios relacionados, consulteInterfaz Amazon ECS puntos de enlace de la VPC de Amazon.

> **Important**
>
> AWS Fargatelas tareas no requieren unAWS PrivateLinkpara Amazon ECS.

Tanto Amazon ECR como Amazon ECS admiten políticas de endpoints. Estas directivas le permiten refinar el acceso a las API de un servicio. Por ejemplo, podría crear una política de endpoints para Amazon ECR que solo permita que las imágenes se envíen a los registros, en

particularAWSCuentas. Una política como esta podría utilizarse para evitar que los datos se exfiltren a través de imágenes de contenedor y, al mismo tiempo, permitir que los usuarios envíen a los registros autorizados de Amazon ECR. Para obtener más información, consulteUsar políticas de punto de enlace de la VPC.

La siguiente política permite a todos losAWSen su cuenta para realizar todas las acciones contra sus repositorios de Amazon ECR:

```
{
  "Statement": [
    {
      "Sid": "LimitECRAccess",
      "Principal": "*",
      "Action": "*",
      "Effect": "Allow",
      "Resource": "arn:aws:ecr:region:your_account_id:repository/*"
    },
  ]
}
```

Puede mejorar aún más esto estableciendo una condición que utilice el nuevo`PrincipalOrgID`Propiedad. Esto evita la inserción y extracción de imágenes por parte de una entidad de IAM que no forma parte de suAWS Organizations. Para obtener más información, consulteaws:PrincipalOrgID.

Recomendamos aplicar la misma política tanto a la`com.amazonaws.region.ecr.dkr`y la`com.amazonaws.region.ecr.api`Puntos de enlace de .

Configuración del agente de contenedores de Amazon ECS

El archivo de configuración del agente de contenedor de Amazon ECS incluye varias variables de entorno relacionadas con la seguridad de la red.`ECS_AWSVPC_BLOCK_IMDS`y`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST`se utilizan para bloquear el acceso de una tarea a los metadatos de Amazon EC2.`HTTP_PROXY`se utiliza para configurar el agente para enrutar a través de un proxy HTTP para conectarse a Internet. Para obtener instrucciones sobre cómo configurar el agente y el tiempo de ejecución de Docker para enrutar a través de un proxy, consulteConfiguración de proxy HTTP.

> **Important**
>
> Esta configuración no está disponible cuando usaAWS Fargate.

Recommendations

Le recomendamos que realice lo siguiente al configurar Amazon VPC, los equilibradores de carga y la red.

Utilizar el cifrado de red cuando corresponda

Debe utilizar el cifrado de red cuando corresponda. Algunos programas de cumplimiento de normas, como PCI DSS, requieren que se cifren los datos en tránsito si los datos contienen datos del titular de la tarjeta. Si la carga de trabajo tiene requisitos similares, configure el cifrado de red.

Los navegadores modernos advierten a los usuarios cuando se conectan a sitios inseguros. Si su servicio está dirigido por un equilibrador de carga orientado al público, use TLS/SSL para cifrar el tráfico desde el navegador del cliente al equilibrador de carga y volver a cifrarlo en el back-end si es necesario.

Usar**`awsvpc`**modo de red y grupos de seguridad cuando necesite controlar el tráfico entre tareas o entre tareas y otros recursos de red

Debe utilizar`awsvpc`el modo de red y los grupos de seguridad cuando necesite controlar el tráfico entre tareas y entre tareas y otros recursos de red. Si el servicio detrás de un ALB, utilice grupos de seguridad para permitir únicamente el tráfico entrante de otros recursos de red utilizando el mismo grupo de seguridad que su ALB. Si su aplicación está detrás de un NLB, configure el grupo de seguridad de la tarea para que sólo permita el tráfico entrante desde el rango CIDR de Amazon VPC y las direcciones IP estáticas asignadas a la NLB.

Los grupos de seguridad también deben utilizarse para controlar el tráfico entre tareas y otros recursos dentro de la VPC de Amazon, como las bases de datos de Amazon RDS.

Crear clústeres en Amazon VPC independientes cuando el tráfico de red debe estar estrictamente aislado

Debe crear clústeres en Amazon VPC independientes cuando el tráfico de red deba estar estrictamente aislado. Evite ejecutar cargas de trabajo que tengan estrictos requisitos de seguridad

en clústeres con cargas de trabajo que no tienen que cumplir con esos requisitos. Cuando el aislamiento estricto de la red es obligatorio, cree clústeres en Amazon VPC independientes y exponga los servicios de forma selectiva a otras VPC de Amazon mediante los endpoints de Amazon VPC. Para obtener más información, consultePuntos de conexión de la VPC de Amazon.

ConfiguraciónAWS PrivateLinkpuntos finales cuando estén justificados

Debe configurarAWS PrivateLinkpuntos finales cuando esté justificado. Si su política de seguridad le impide adjuntar una puerta de Internet Gateway (IGW) a sus VPC de Amazon, configureAWS PrivateLinkpara Amazon ECS y otros servicios como Amazon ECR,AWS Secrets Managery Amazon CloudWatch.

Utilice los registros de flujo de Amazon VPC para analizar el tráfico hacia y desde las tareas de larga ejecución

Debe utilizar los registros de flujo de Amazon VPC para analizar el tráfico hacia y desde tareas de larga duración. Tareas que utilizan`awsvpc`modo de red obtener su propio ENI. De este modo, puede supervisar el tráfico que va hacia y desde tareas individuales mediante los registros de flujo de Amazon VPC. Una actualización reciente de Amazon VPC Flow Logs (v3) enriquece los registros con metadatos de tráfico, incluidos el ID de vpc, el ID de subred y el ID de instancia. Estos metadatos se pueden utilizar para ayudar a reducir una investigación. Para obtener más información, consulteLogs de flujo de Amazon VPC.

> **Note**
>
> Debido a la naturaleza temporal de los contenedores, los registros de flujo no siempre pueden ser una forma eficaz de analizar patrones de tráfico entre diferentes contenedores o contenedores y otros recursos de red.

Administración de secretos

Los secretos, como las claves API y las credenciales de base de datos, son utilizados con frecuencia por las aplicaciones para obtener acceso a otros sistemas. A menudo consisten en un nombre de usuario y una contraseña, un certificado o una clave API. El acceso a estos secretos debe restringirse a entidades de IAM específicas que utilizan IAM y se inyectan en contenedores en tiempo de ejecución.

Los secretos se pueden inyectar sin problemas en contenedores desdeAWS Secrets ManagerEl almacén de parámetros Amazon EC2 Systems Manager. Estos secretos se pueden hacer referencia en su tarea como cualquiera de los siguientes.

1. Se hace referencia a ellas como variables de entorno que utilizan la propiedad`secrets`parámetro de definición de contenedor.
2. Se les hace referencia como`secretOptions`si su plataforma de registro requiere autenticación. Para obtener más información, consulteOpciones de configuración de registros de.
3. Se les hace referencia como secretos extraídos por imágenes que usan la función`repositoryCredentials`parámetro de definición de contenedor si el registro del que se extrae el contenedor requiere autenticación. Utilice este método al extraer imágenes de Docker Hub. Para obtener más información, consulteAutenticación privada del Registro para tareas.

Recommendations

Le recomendamos que haga lo siguiente al configurar la gestión de secretos.

UsarAWS Secrets Managero el almacén de parámetros Amazon EC2 Systems Manager para almacenar materiales secretos

Debe almacenar de forma segura las claves de la API, las credenciales de la base de datos y otros materiales secretos enAWS Secrets Managero como un parámetro cifrado en el almacén de parámetros Amazon EC2 Systems Manager. Estos servicios son similares porque ambos son almacenes de clave-valor administrados que usanAWS KMSpara cifrar datos confidenciales.AWS Secrets Manager, sin embargo, también incluye la capacidad de rotar automáticamente secretos, generar secretos aleatorios y compartir secretos a través deAWSCuentas. Si considera que estas características importantes, utiliceAWS Secrets Managerde lo contrario, utilice parámetros cifrados.

> **Note**
>
> Tareas que hacen referencia a un secreto deAWS Secrets ManagerEl almacén de parámetros Amazon EC2 Systems Manager requieren unRol de ejecución de tareascon una política que otorga a Amazon ECS acceso al secreto deseado y, si procede, alAWS KMSClave utilizada para cifrar y descifrar ese secreto.

> **Important**
>
> Los secretos a los que se hace referencia en las tareas no se rotan automáticamente. Si su secreto cambia, debe forzar una nueva implementación o iniciar una nueva tarea para recuperar el valor secreto más reciente. Para obtener más información, consulte los siguientes temas:
>
> - AWS Secrets Manager: Inyección de datos como variables de entorno
> - Almacén de parámetros Amazon EC2 Systems Manager: Inyección de datos como variables de entorno

Recuperación de datos de un bucket cifrado de Amazon S3

Debido a que el valor de las variables de entorno puede filtrarse inadvertidamente en los registros y se revelan al ejecutar`docker inspect`, debe almacenar secretos en un depósito cifrado de Amazon S3 y utilizar funciones de tarea para restringir el acceso a esos secretos. Al hacerlo, su aplicación debe escribirse para leer el secreto del bucket de Amazon S3. Para obtener instrucciones, consulteEstablecer el comportamiento del cifrado predeterminado del lado del servidor para los buckets de Amazon S3.

Monte el secreto en un volumen usando un contenedor sidecar

Debido a que existe un riesgo elevado de fuga de datos con variables de entorno, debe ejecutar un contenedor sidecar que lea sus secretos deAWS Secrets Managery escribirlos en un volumen compartido. Este contenedor puede ejecutarse y salir antes del contenedor de la aplicación mediantePedido de contenedores de Amazon ECS. Al hacer esto, el contenedor de la aplicación se monta posteriormente el volumen donde se escribió el secreto. Al igual que el método de depósito de Amazon S3, su aplicación debe escribirse para leer el secreto del volumen compartido. Dado que el volumen se circunscribe a la tarea, el volumen se elimina automáticamente después de que se detenga la tarea. Para ver un ejemplo de contenedor sidecar, consulte laaws-secret-sidecar-inyectorProyecto de.

> **Note**
>
> En Amazon EC2, el volumen en el que se escribe el secreto se puede cifrar con unAWS KMSclave administrada por el cliente. EnAWS Fargate, el almacenamiento por volumen se cifra automáticamente mediante una clave administrada por servicio.

Recursos adicionales

- Transferir secretos a los contenedores en una tarea de Amazon ECS
- Cámaraes un contenedor para almacenar secretos en el almacén de parámetros Amazon EC2 Systems Manager

Compliance

Su responsabilidad de conformidad al utilizar Amazon ECS se determina en función de la confidencialidad de los datos, los objetivos de conformidad de su empresa, así como de la legislación y los reglamentos aplicables.

AWSproporciona los siguientes recursos para ayudar con el cumplimiento de normas:

- Guías de inicio rápido de seguridad y cumplimiento: Estas guías de implementación tratan consideraciones sobre arquitectura y ofrecen pasos para implementar los entornos de referencia centrados en la seguridad y la conformidad enAWS.
- Documento técnico de Architecting for HIPAA Security and Compliance: Este documento técnico describe cómo las empresas pueden utilizarAWSpara crear aplicaciones que se ajusten al estándar HIPAA.
- AWSServicios en el ámbito del programa de conformidad: Esta lista contiene elAWSServicios en el ámbito de los programas de conformidad específicos. Para obtener más información, consulteAWSProgramas de conformidad.

Normas de Seguridad de Datos del Sector de las Tarjetas de Pago (PCI DSS

Es importante que comprenda el flujo completo de datos de los titulares de tarjetas (CHD) dentro del entorno al adherirse a PCI DSS. El flujo CHD determina la aplicabilidad del DSS PCI, define los límites y componentes de un entorno de datos de titulares de tarjetas (CDE) y, por lo tanto, el alcance de una evaluación PCI DSS. La determinación precisa del alcance de PCI DSS es clave para definir la postura de seguridad y, en última instancia, una evaluación exitosa. Los clientes deben disponer de un procedimiento para determinar el alcance que garantice su integridad y detecte cambios o desviaciones del ámbito.

La naturaleza temporal de las aplicaciones en contenedores proporciona complejidades adicionales al auditar configuraciones. Como resultado, los clientes deben mantener un conocimiento de todos los parámetros de configuración de contenedores para garantizar que se cumplan los requisitos de cumplimiento a lo largo de todas las fases del ciclo de vida de un contenedor.

Para obtener información adicional sobre cómo lograr el cumplimiento de PCI DSS en Amazon ECS, consulte los siguientes documentos técnicos.

- Arquitecting on Amazon ECS para la conformidad con PCI DSS
- Arquitectura para el alcance y la segmentación de PCI DSS en AWS

HIPAA (Ley de Portabilidad y Responsabilidad de Health os Médicos de EE. UU)

El uso de Amazon ECS con cargas de trabajo que procesan información de estado protegida (PHI) no requiere configuración adicional. Amazon ECS actúa como un servicio de orquestación que coordina el lanzamiento de contenedores en Amazon EC2. No funciona con o sobre datos dentro de la carga de trabajo que se está orquestando. De acuerdo con las regulaciones de la HIPAA y laAWSAddendum de Business Associate, PHI debe cifrarse en tránsito y en reposo cuando se acceda a través de contenedores lanzados con Amazon ECS.

Varios mecanismos para cifrar en reposo están disponibles con cadaAWS, como Amazon S3, Amazon EBS yAWS KMS. Puede implementar una red de superposición (como VNS3 o Weave Net) para garantizar el cifrado completo de la PHI transferida entre contenedores o para proporcionar una capa de cifrado redundante. También debe habilitarse el registro completo y todos los registros de contenedor deben dirigirse a Amazon CloudWatch. Para obtener más información, consulteArquitecting for HIPAA Security and Compliance.

Recommendations

Debe involucrar a los propietarios del programa de cumplimiento dentro de su empresa con anticipación y utilizar elAWSModelo de responsabilidad compartidapara identificar la propiedad del control de cumplimiento para el éxito con los programas de cumplimiento pertinentes.

Registro y monitorización

El registro y la supervisión son un aspecto importante del mantenimiento de la fiabilidad, la disponibilidad y el rendimiento de Amazon ECS y suAWSSoluciones de.AWSproporciona varias herramientas para monitorizar sus recursos de Amazon ECS y responder a posibles incidentes.

- Alarmas de Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CloudTrail Registros

Puede configurar los contenedores de las tareas para enviar información de registro a los Amazon CloudWatch Logs. Si utiliza laAWS FargatePara sus tareas, puede ver los registros de los contenedores. Si utiliza el tipo de lanzamiento de Amazon EC2, puede ver distintos registros desde sus contenedores en una ubicación cómoda. Esto también evita que los registros de contenedor ocupen espacio en disco en sus instancias de contenedor.

Para obtener más información acerca de los Amazon CloudWatch Logs, consulteMonitoree registros de instancias Amazon EC2en laGuía del usuario de Amazon CloudWatch. Para obtener instrucciones sobre cómo enviar registros de contenedor desde sus tareas a los Amazon CloudWatch Logs, consulteMediante la`awslogs`controlador de registro.

Registro de contenedores con bit fluido

AWSproporciona una imagen de Fluent Bit con complementos para los Amazon CloudWatch Logs y Amazon Kinesis Data Firehose. Esta imagen proporciona la capacidad de enrutar registros a los destinos Amazon CloudWatch y Amazon Kinesis Data Firehose (que incluyen Amazon S3, Amazon Elasticsearch Service y Amazon Redshift). Recomendamos usar Fluent Bit como router de registro porque tiene una tasa de utilización de recursos más baja que Fluentd. Para obtener más información, consulteAmazon CloudWatch Logs para bits fluidosyAmazon Kinesis Data Firehose para bits fluidos.

LaAWSpara la imagen Fluent Bit (Bit Fluent) está disponible en:

- Galería pública de Amazon ECR en Amazon ECR
- Repositorio de Amazon ECR(en la mayoría de las regiones de alta disponibilidad)
- Docker Hub

A continuación se muestra la sintaxis que se va a utilizar para la CLI de Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:tag
```

Por ejemplo, puede extraer la últimaAWSpara la imagen de bit fluido usando este comando Docker CLI:

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:latest
```

Consulte también las siguientes entradas de blog para obtener más información sobre Fluent Bit y características relacionadas:

- Bit fluido para Amazon EKS en AWS Fargate
- Registro centralizado de contenedores con bit fluido
- Creación de un agregador de soluciones de registro escalable conAWS Fargate, Fluentd y Amazon Kinesis Data Firehose

Enrutamiento de registros personalizado - FireLens para Amazon ECS

Con FireLens para Amazon ECS, puede utilizar parámetros de definición de tareas para enviar registros a unAWSo servicio deAWSDestino de red de socios (APN) para el almacenamiento y el análisis de registros. FireLens funciona con Fluentd y Fluent Bit. Proporcionamos elAWSpara la imagen de bit fluido. O bien, puede utilizar su propia imagen Fluentd o Fluent Bit.

Debe tener en cuenta las siguientes condiciones y consideraciones al utilizar FireLens para Amazon ECS:

- FireLens para Amazon ECS es compatible con tareas alojadas tanto enAWS Fargatey Amazon EC2.
- FireLens para Amazon ECS es compatible conAWS CloudFormationplantillas de. Para obtener más información, consulteFirelensConfiguration de AWS::ECS::TaskDefinitionen laAWS CloudFormationGuía del usuario.
- En el caso de las tareas que utilizan el`bridge`Para ello, los contenedores con la configuración FireLens deben iniciarse antes de que se inicie cualquiera de los contenedores de aplicación que se basan en él. Para controlar el orden en que se inicien los contenedores, utilice las condiciones de dependencia en la definición de la tarea. Para obtener más información, consulteDependencia de contenedor.

Seguridad de AWS Fargate

Le recomendamos que tenga en cuenta las siguientes prácticas recomendadas al utilizarAWS Fargate.

UsarAWS KMSpara cifrar el almacenamiento efímero

Debe tener su almacenamiento efímero encriptado porAWS KMS. Para las tareas de Amazon ECS alojadas enAWS FargateUso de la versión de plataforma`1.4.0`o posterior, cada tarea de recibe 20 GB de almacenamiento efímero. La cantidad de almacenamiento no se puede ajustar. Para estas tareas que se inicien a partir del 28 de mayo de 2020, el almacenamiento efímero se cifra con un algoritmo de cifrado AES-256 utilizando una clave de cifrado administrada porAWS Fargate.

Ejemplo: Iniciar una tarea Amazon ECS enAWS FargateVersión 1.4.0 de la plataforma con cifrado efímero

El siguiente comando iniciará una tarea de Amazon ECS enAWS FargateVersión de la plataforma 1.4. Dado que esta tarea se inicia como parte del clúster de Amazon ECS, utiliza los 20 GB de almacenamiento efímero que se cifra automáticamente.

```
aws ecs run-task --cluster clustername \
  --task-definition taskdefinition:version \
  --count 1
  --launch-type "FARGATE" \
  --platform-version 1.4.0 \
  --network-configuration
 "awsvpcConfiguration={subnets=[subnetid],securityGroups=[securitygroupid]}" \
  --region region
```

Capacidad de SYS_PTRACE para el seguimiento de syscall del kernel

Docker proporciona la configuración predeterminada de las capacidades de Linux que se agregan o eliminan del contenedor. Para obtener más información acerca de las capacidades disponibles, consultePrivilegio de ejecución y capacidades de Linuxen laEjecute Docker.

Tareas que se inician enAWS Fargatesolo admite agregar el`SYS_PTRACE`Capacidad del kernel.

Consulte el video tutorial a continuación que muestra cómo usar esta función a través de SysdigFalcoProyecto de.

#ContainersFromTheCouch - Solución de problemas deAWS FargateTarea con la capacidad SYS_PTRACE

El código discutido en el video anterior se puede encontrar en GitHubAquí.

Seguridad de tareas y contenedores

Debe considerar la imagen del contenedor como su primera línea de defensa contra un ataque. Una imagen insegura y mal construida puede permitir a un atacante escapar de los límites del contenedor y obtener acceso al host. Debe hacer lo siguiente para mitigar el riesgo de que esto suceda.

Recommendations

Le recomendamos que haga lo siguiente al configurar las tareas y los contenedores.

Cree imágenes mínimas o utilice imágenes distroless

Comience eliminando todos los binarios extraños de la imagen del contenedor. Si está utilizando una imagen desconocida de Docker Hub, inspeccione la imagen para hacer referencia al contenido de cada una de las capas del contenedor. Puede utilizar una aplicación comoBucearPara ello.

También puede utilizardesangustiadoimágenes que solo incluyen su aplicación y sus dependencias en tiempo de ejecución. No contienen gestores de paquetes ni shells. Las imágenes sin problemas mejoran la «señal al ruido de los escáneres y reducen la carga de establecer la procedencia justo a lo que necesitas». Para obtener más información, consulte la documentación de GitHub sobredesangustiado.

Docker tiene un mecanismo para crear imágenes a partir de una imagen reservada y mínima conocida comorasguño. Información de Formore, consulteCrear una imagen principal simple usandorasguñoen la documentación de Docker. Con lenguajes como Go, puede crear un binario vinculado estático y hacer referencia a él en su Dockerfile. El siguiente ejemplo muestra cómo lograr esto.

```
############################
# STEP 1 build executable binary
############################
FROM golang:alpine AS builder
# Install git.
# Git is required for fetching the dependencies.
RUN apk update && apk add --no-cache git
WORKDIR $GOPATH/src/mypackage/myapp/
```

```
COPY . .
# Fetch dependencies.
# Using go get.
RUN go get -d -v
# Build the binary.
RUN go build -o /go/bin/hello
############################
# STEP 2 build a small image
############################
FROM scratch
# Copy our static executable.
COPY --from=builder /go/bin/hello /go/bin/hello
# Run the hello binary.
ENTRYPOINT ["/go/bin/hello"]
This creates a container image that consists of your application and nothing else,
 making it extremely secure.
```

El ejemplo anterior es también un ejemplo de una compilación de varias etapas. Estos tipos de compilaciones son atractivos desde el punto de vista de la seguridad, ya que puede usarlos para minimizar el tamaño de la imagen final enviada al registro del contenedor. Las imágenes de contenedor carentes de herramientas de compilación y otros binarios extraños mejoran la postura de seguridad al reducir la superficie de ataque de la imagen. Para obtener más información acerca de las compilaciones de varias etapas, consultecreación de compilaciones de varias etapas.

Escanea tus imágenes en busca de vulnerabilidades

Al igual que sus homólogos de máquinas virtuales, las imágenes de contenedores pueden contener binarios y bibliotecas de aplicaciones con vulnerabilidades o desarrollar vulnerabilidades con el tiempo. La mejor manera de protegerse contra exploits es escanear regularmente sus imágenes con un escáner de imágenes. Las imágenes almacenadas en Amazon ECR se pueden escanear en formato push o bajo demanda (una vez cada 24 horas). Amazon ECR utiliza actualmenteClair, una solución de escaneo de imágenes de código abierto. Después de escanear una imagen, los resultados se registran en la secuencia de eventos de Amazon ECR en Amazon EventBridge. También puede ver los resultados de un análisis desde la consola de Amazon ECR o llamando alDescribirImagesEscanhallazgosAPI DE. Imágenes con`HIGH`or`CRITICAL`debe eliminarse o reconstruirse. Si una imagen que se ha implementado desarrolla una vulnerabilidad, debe reemplazarse lo antes posible.

Docker Desktop Edge versión 2.3.6.0o posterior puedescanimágenes locales. Las exploraciones están alimentadas porSnyk, un servicio de seguridad de aplicaciones. Cuando se descubren

vulnerabilidades, Snyk identifica las capas y dependencias con la vulnerabilidad en Dockerfile. También recomienda alternativas seguras como usar una imagen base más delgada con menos vulnerabilidades o actualizar un paquete en particular a una versión más reciente. Mediante el análisis Docker, los desarrolladores pueden resolver posibles problemas de seguridad antes de enviar sus imágenes al registro.

- Automatizar el cumplimiento de las imágenes mediante Amazon ECR yAWS Security Hubexplica cómo mostrar la información de vulnerabilidad de Amazon ECR enAWS Security Huby automatizar la corrección bloqueando el acceso a imágenes vulnerables.

Eliminar permisos especiales de las imágenes

Los indicadores de derechos de accesosetuidysetgidpermiten ejecutar un ejecutable con los permisos del propietario o grupo del ejecutable. Elimine todos los binarios con estos derechos de acceso de la imagen, ya que estos binarios se pueden utilizar para escalar privilegios. Considere eliminar todos los shells y utilidades comocurlque se pueden usar con fines maliciosos. Encontrará los archivos consetuidysetgidAcceso mediante el siguiente comando.

```
find / -perm /6000 -type f -exec ls -ld {} \;
```

Para quitar estos permisos especiales de estos archivos, agregue la siguiente directiva a la imagen del contenedor.

```
RUN find / -xdev -perm /6000 -type f -exec chmod a-s {} \; || true
```

Crear un conjunto de imágenes seleccionadas

En lugar de permitir a los desarrolladores crear sus propias imágenes, cree un conjunto de imágenes revisadas para las diferentes pilas de aplicaciones de su organización. Al hacerlo, los desarrolladores pueden renunciar a aprender a componer archivos Dockery concentrarse en escribir código. A medida que los cambios se fusionan en su base de código, una canalización CI/CD puede compilar automáticamente el activo y luego almacenarlo en un repositorio de artefactos. Y, por último, copie el artefacto en la imagen adecuada antes de enviarlo a un registro Docker como Amazon ECR. Por lo menos debe crear un conjunto de imágenes base desde las que los desarrolladores pueden crear sus propios Dockerfiles. Debe evitar extraer imágenes de Docker Hub. No siempre sabes lo que hay en la imagen y alrededor de una quinta parte de las 1000 mejores imágenes tienen

vulnerabilidades. Puede encontrar una lista de esas imágenes y sus vulnerabilidades enhttps://vulnerablecontainers.org/.

Analizar paquetes de aplicaciones y bibliotecas en busca de vulnerabilidades

El uso de bibliotecas de código abierto es ahora común. Al igual que con los sistemas operativos y los paquetes del sistema operativo, estas bibliotecas pueden tener vulnerabilidades. Como parte del ciclo de vida del desarrollo, estas bibliotecas deben analizarse y actualizarse cuando se encuentren vulnerabilidades críticas.

Docker Desktop realiza exploraciones locales usando Snyk. También se puede utilizar para encontrar vulnerabilidades y posibles problemas de licencias en bibliotecas de código abierto. Se puede integrar directamente en los flujos de trabajo de desarrolladores, lo que le permite mitigar los riesgos que plantean las bibliotecas de código abierto. Para obtener más información, consulte los siguientes temas:

- Herramientas de seguridad de aplicaciones de código abiertoincluye una lista de herramientas para detectar vulnerabilidades en aplicaciones.
- Hoja de trucos de digitalización de Docker

Realizar análisis de código estático

Debe realizar un análisis de código estático antes de crear una imagen de contenedor. Se realiza contra su código fuente y se utiliza para identificar errores de codificación y código que podría ser explotado por un actor malicioso, como inyecciones de errores. SonarQubees una opción popular para pruebas de seguridad de aplicaciones estáticas (SAST), con soporte para una variedad de lenguajes de programación diferentes.

Ejecutar contenedores como un usuario no root

Debe ejecutar contenedores como un usuario que no sea root. De forma predeterminada, los contenedores se ejecutan como`root`a menos que el`USER`está incluida en su Dockerfile. Las capacidades predeterminadas de Linux asignadas por Docker restringen las acciones que se pueden ejecutar como`root`, pero solo marginalmente. Por ejemplo, un contenedor que se ejecuta como`root`todavía no tiene permiso para acceder a los dispositivos.

Como parte de su canalización CI/CD, debe pelusas Dockerfiles para buscar el`USER`y fallar la compilación si falta. Para obtener más información, consulte los siguientes temas:

- Dockerfile-pelusaes una herramienta de código abierto de RedHat que se puede utilizar para comprobar si el archivo se ajusta a las mejores prácticas.
- Hadolintes otra herramienta para crear imágenes Docker que se ajusten a las mejores prácticas.

Usar un sistema de archivos raíz de sólo lectura

Debe utilizar un sistema de archivos raíz de sólo lectura. El sistema de archivos raíz de un contenedor se puede escribir de forma predeterminada. Cuando configura un contenedor con unR0(sólo lectura) sistema de archivos raíz que le obliga a definir explícitamente dónde se pueden conservar los datos. Esto reduce la superficie de ataque porque no se puede escribir en el sistema de archivos del contenedor a menos que se otorguen permisos específicamente.

> **Note**
>
> Tener un sistema de archivos raíz de sólo lectura puede causar problemas con ciertos paquetes del sistema operativo que esperan poder escribir en el sistema de archivos. Si está planeando usar sistemas de archivos raíz de solo lectura, pruebe a fondo de antemano.

Configurar tareas con límites de CPU y memoria (Amazon EC2)

Debe configurar tareas con límites de CPU y memoria para minimizar el siguiente riesgo. Los límites de recursos de una tarea establecen un límite superior para la cantidad de CPU y memoria que pueden reservar todos los contenedores de una tarea. Si no se establecen límites, las tareas tienen acceso a la CPU y la memoria del host. Esto puede causar problemas en los que las tareas implementadas en un host compartido pueden privar a otras tareas de recursos del sistema.

> **Note**
>
> Amazon ECS enAWS Fargaterequieren que especifique límites de CPU y memoria porque utiliza estos valores para fines de facturación. Una tarea que acapara todos los recursos del sistema no es un problema para Amazon ECS Fargate porque cada tarea se ejecuta en su propia instancia dedicada. Si no especifica un límite de memoria, Amazon ECS asigna un mínimo de 4 MB a cada contenedor. Del mismo modo, si no se establece ningún límite de CPU para la tarea, el agente contenedor de Amazon ECS le asigna un mínimo de 2 CPU.

Utilizar etiquetas inmutables con Amazon ECR

Con Amazon ECR, puede y debe utilizar la configuración de imágenes con etiquetas inmutables. Esto evita enviar una versión modificada o actualizada de una imagen al repositorio de imágenes con una etiqueta idéntica. Esto protege contra un atacante que empuje una versión comprometida de una imagen sobre su imagen con la misma etiqueta. Mediante el uso de etiquetas inmutables, se obliga efectivamente a empujar una nueva imagen con una etiqueta diferente para cada cambio.

Evite ejecutar contenedores como privilegiados (Amazon EC2)

Debe evitar ejecutar contenedores como privilegiados. En segundo plano, los contenedores se ejecutan como`privileged`se ejecutan con privilegios extendidos en el host. Esto significa que el contenedor hereda todas las capacidades de Linux asignadas `arooten` el host. Su uso debe estar severamente restringido o prohibido. Recomendamos configurar la variable de entorno de agente de contenedor de Amazon ECS`ECS_DISABLE_PRIVILEGED`De `atrue`para evitar que los contenedores se ejecuten como`privileged`en hosts particulares si`privileged`no es necesario. También puede utilizarAWS Lambdapara analizar las definiciones de tareas para el uso de la herramienta`privileged`Parámetro.

> **Note**
>
> Ejecutar un contenedor como`privileged`no es compatible con Amazon ECSAWS Fargate.

Eliminar capacidades innecesarias de Linux del contenedor

A continuación se muestra una lista de las capacidades predeterminadas de Linux asignadas a los contenedores Docker. Para obtener más información acerca de cada función, consulteInformación general sobre las capacidades de Linux.

```
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_FOWNER, CAP_FSETID, CAP_KILL,
CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_NET_BIND_SERVICE,
CAP_NET_RAW, CAP_SYS_CHROOT, CAP_MKNOD, CAP_AUDIT_WRITE,
CAP_SETFCAP
```

Si un contenedor no requiere todas las capacidades del núcleo de Docker enumeradas anteriormente, considere soltarlas del contenedor. Para obtener más información acerca de cada

función de Docker, consulteKernalCapabilities. Puede averiguar qué capacidades están en uso haciendo lo siguiente:

- Instale el paquete del SOlibcap-ngy ejecute la`pscap`para enumerar las capacidades que cada proceso está utilizando.
- También puede utilizarcapshpara descifrar qué capacidades está usando un proceso.
- ConsulteCapacidades de Linux 101para obtener más información.

Utilice una clave gestionada por el cliente (CMK) para cifrar las imágenes enviadas a Amazon ECR

Debe utilizar una clave gestionada por el cliente (CMK) para invalidar las imágenes que se envían a Amazon ECR. Las imágenes que se envían a Amazon ECR se cifran automáticamente en reposo con unAWS Key Management Service(AWS KMS) clave administrada. Si prefiere utilizar su propia clave, Amazon ECR admite ahoraAWS KMScifrado con claves administradas por el cliente (CMK). Antes de habilitar el cifrado del lado del servidor con un CMK, revise las consideraciones enumeradas en la documentación deCifrado en reposo.

Seguridad de tiempo de ejecución

Su seguridad de tiempo de ejecución proporciona protección activa a los contenedores mientras se están ejecutando. La idea es detectar y evitar que se produzcan actividades maliciosas en sus contenedores.

Con informática segura (`seccomp`) puede evitar que una aplicación contenedorizada realice ciertos syscalls al kernel del sistema operativo host subyacente. Mientras que el sistema operativo Linux tiene unos cientos de llamadas al sistema, la mayoría de ellas no son necesarias para ejecutar contenedores. Al restringir qué syscalls puede hacer un contenedor, puede disminuir eficazmente la superficie de ataque de su aplicación.

Para comenzar con seccomp, puede usar`strace`para generar un seguimiento de pila para ver qué llamadas al sistema está realizando su aplicación. Puede utilizar una herramienta como`syscall2seccomp`para crear un perfil seccomp a partir de los datos recopilados del seguimiento de la pila. Para obtener más información, consultestraceysyscall2seccomp.

A diferencia del módulo de seguridad SELinux, seccomp no puede aislar contenedores entre sí. Sin embargo, protege el núcleo del host de syscalls no autorizados. Funciona interceptando syscalls y permitiendo que pasen solo aquellos que han sido permitidos en la lista. Docker tiene

unapredeterminadaperfil seccomp adecuado para la mayoría de las cargas de trabajo de propósito general.

> **Note**
>
> También es posible crear sus propios perfiles para cosas que requieren privilegios adicionales.

AppArmor es un módulo de seguridad de Linux similar a seccomp, pero restringe las capacidades de un contenedor incluyendo el acceso a partes del sistema de archivos. Se puede ejecutar en`enforcement`o`complain`Modo. Dado que crear perfiles de AppArmor puede ser un reto, le recomendamos que utilice una herramienta combane. Para obtener más información sobre AppArmor, consulte laAppArmor(Se ha creado el certificado).

> **Important**
>
> AppArmor sólo está disponible para las distribuciones Ubuntu y Debian de Linux.

Recommendations

Le recomendamos que realice las siguientes acciones al configurar la seguridad en tiempo de ejecución.

Utilice una solución de terceros para la defensa en tiempo de ejecución

Utilice una solución de terceros para la defensa en tiempo de ejecución. Si está familiarizado con el funcionamiento de la seguridad de Linux, cree y administre perfiles seccomp y AppArmor. Ambos son proyectos de código abierto. De lo contrario, considere utilizar un servicio de terceros diferente en su lugar. La mayoría utiliza el aprendizaje automático para bloquear o alertar sobre actividades sospechosas. Para obtener una lista de soluciones de terceros disponibles, consulteAWS Marketplacepara Contenedores.

Agregar o eliminar capacidades de Linux mediante políticas seccomp

Use seccomp para tener un mayor control sobre las capacidades de Linux y para evitar errores de comprobación de syscall. Seccomp funciona como un filtro syscall que revoca el permiso para ejecutar ciertos syscalls o para usar agruments específicos.

AWSSocios

Puede utilizar cualquiera de las siguientesAWSProductos asociados para añadir seguridad y características adicionales a sus cargas de trabajo de Amazon ECS. Para obtener más información, consulteSocios de Amazon ECS.

Aqua Security

Puede usarAqua Securitypara proteger sus aplicaciones nativas de la nube desde el desarrollo hasta la producción. Aqua Cloud Native Security Platform se integra con sus recursos nativos de la nube y herramientas de orquestación para proporcionar seguridad transparente y automatizada. Puede prevenir actividades sospechosas y ataques en tiempo real, y ayudar a hacer cumplir las políticas y simplificar el cumplimiento normativo.

Palo Alto Networks

Palo Alto Networksproporciona seguridad y protección para sus hosts, contenedores e infraestructura sin servidor en la nube y durante todo el ciclo de vida de desarrollo y software.

Twistlock es suministrado por Palo Alto Networks y puede integrarse con Amazon ECS FireLens. Con él, tiene acceso a registros e incidentes de seguridad de alta fidelidad que se agregan sin problemas en variosAWSServicios de . Estos incluyen Amazon CloudWatch, Amazon Athena y Amazon Kinesis. Twistlock protege las cargas de trabajo que se implementan enAWSServicios de contenedores.

Sysdig

Puede usarSysdigpara ejecutar cargas de trabajo nativas de la nube seguras y compatibles en escenarios de producción. Sysdig Secure DevOps Platform cuenta con funciones integradas de seguridad y cumplimiento normativo para proteger sus cargas de trabajo nativas de la nube, y también ofrece escalabilidad, rendimiento y personalización de nivel empresarial.

Historial de documentos de la Guía de Prácticas recomendadas de Amazon ECS

En la siguiente tabla se describen las versiones de documentación de la Guía de Prácticas recomendadas de Amazon ECS.

update-history-change	update-history-description	update-history-date
Prácticas recomendadas de seguridad	Se han añadido prácticas recomendadas para la gestión de la seguridad para cargas de trabajo de Amazon ECS.	26 de mayo de 2021
Mejores prácticas de escalado automático y administración de capacidad	Se han añadido prácticas recomendadas para el escalado automático y la gestión de la capacidad para cargas de trabajo de Amazon ECS.	14 de mayo de 2021
Prácticas recomendadas de almacenamiento	Se han añadido prácticas recomendadas para el almacenamiento persisten te de cargas de trabajo de Amazon ECS.	7 de mayo de 2021
Prácticas recomendadas de redes	Se han añadido prácticas recomendadas para la gestión de redes para cargas de trabajo de Amazon ECS.	6 de abril de 2021
Versión inicial	Lanzamiento inicial de la Guía de Prácticas recomendadas de Amazon ECS	6 de abril de 2021

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.