aws

Guia de práticas recomendadas do

Amazon Elastic Container Service

Amazon Elastic Container Service: Guia de práticas recomendadas do

Table of Contents

Introduction

O Amazon Elastic Container Service (Amazon ECS) é um serviço de gerenciamento de contêineres altamente escalável e rápido que facilita a execução, a interrupção e o gerenciamento de contêineres em um cluster. Este guia aborda muitas das práticas recomendadas operacionais mais importantes e, ao mesmo tempo, explica os principais tópicos subjacentes ao funcionamento dos aplicativos baseados no Amazon ECS. O objetivo é fornecer uma abordagem concreta e acionável para operar e solucionar problemas de aplicativos baseados no Amazon ECS.

Este guia será revisado regularmente para incorporar novas práticas recomendadas do Amazon ECS. Se você tiver alguma dúvida ou comentário sobre qualquer um dos conteúdos deste guia, crie um problema no repositório do GitHub. Para obter mais informações, consulteGuia de práticas recomendadas do Amazon ECSno GitHub

- Melhores práticas — Networking
- Práticas recomendadas - Auto scaling e gerenciamento de capacidade
- Práticas recomendadas - Armazenamento persistente
- Práticas recomendadas - Segurança

Melhores práticas — Networking

Os aplicativos modernos são tipicamente criados a partir de vários componentes distribuídos que se comunicam entre si. Por exemplo, um aplicativo móvel ou Web pode se comunicar com um ponto final de API e a API pode ser alimentada por vários microsserviços que se comunicam pela Internet.

Este guia apresenta as práticas recomendadas para criar uma rede na qual os componentes de seu aplicativo possam se comunicar entre si de forma segura e escalável.

Tópicos

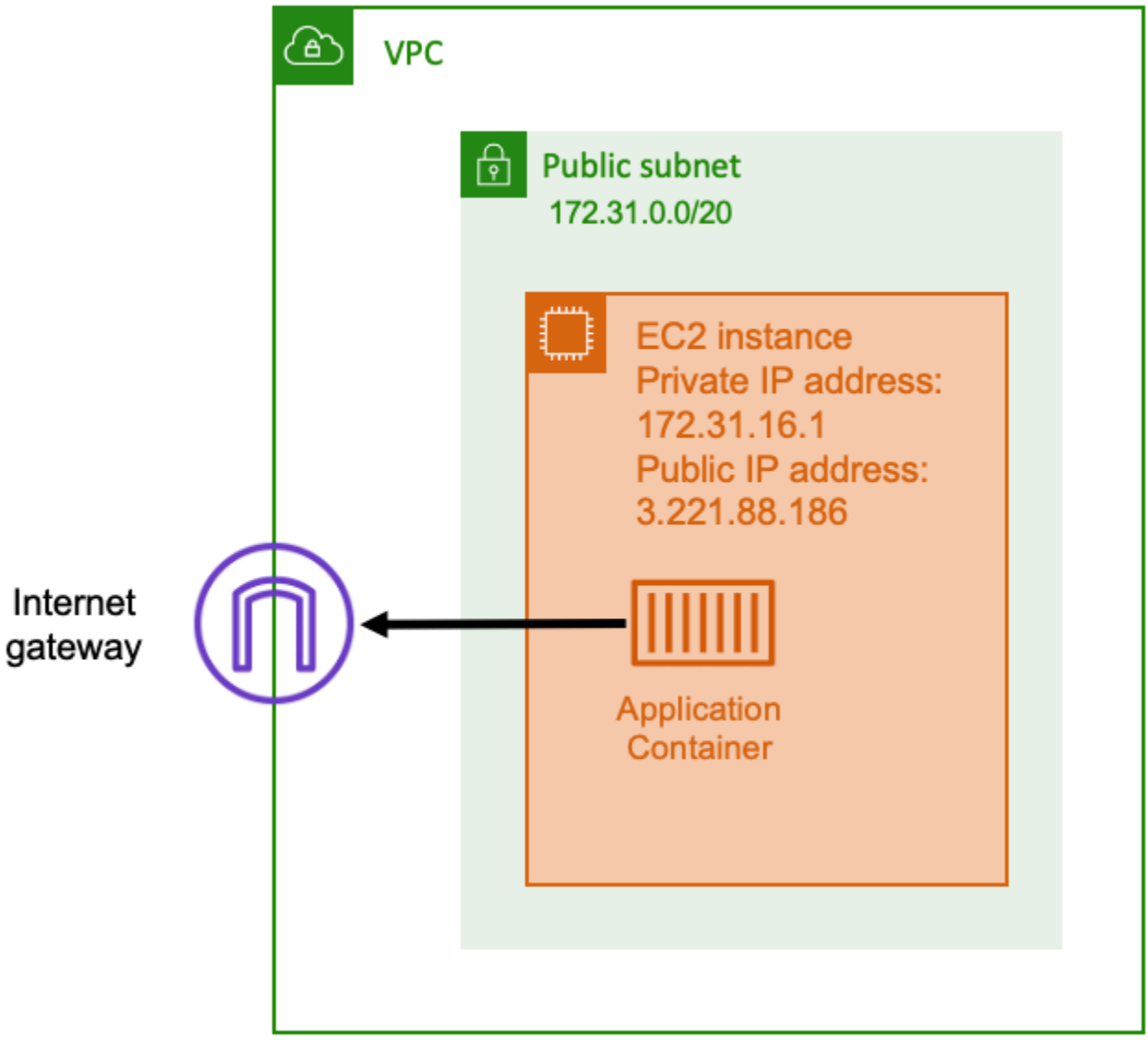
Conexão com a Internet

A maioria dos aplicativos em contêineres tem pelo menos alguns componentes que precisam de acesso de saída à Internet. Por exemplo, o back-end de um aplicativo móvel requer acesso de saída às notificações por push.

O Amazon Virtual Private Cloud tem dois métodos principais para facilitar a comunicação entre sua VPC e a Internet.

Tópicos

Usando uma sub-rede pública e um gateway de Internet



Ao usar uma sub-rede pública que tem uma rota para um gateway da Internet, o aplicativo contêineres pode ser executado em um host dentro de uma VPC em uma sub-rede pública. O host que executa o contêiner recebe um endereço IP público. Este endereço IP público é roteável a partir da Internet. Para obter mais informações, consulteGateways da InternetnoGuia do usuário da Amazon VPC.

Essa arquitetura de rede facilita a comunicação direta entre o host que executa seu aplicativo e outros hosts na internet. A comunicação é bidirecional. Isso significa que você não só pode estabelecer uma conexão de saída com qualquer outro host na Internet, mas outros hosts na Internet

também podem tentar se conectar ao seu host. Portanto, você deve prestar muita atenção ao seu grupo de segurança e regras de firewall. Isso é para garantir que outros hosts na Internet não possam abrir nenhuma conexão que você não deseja que seja aberta.

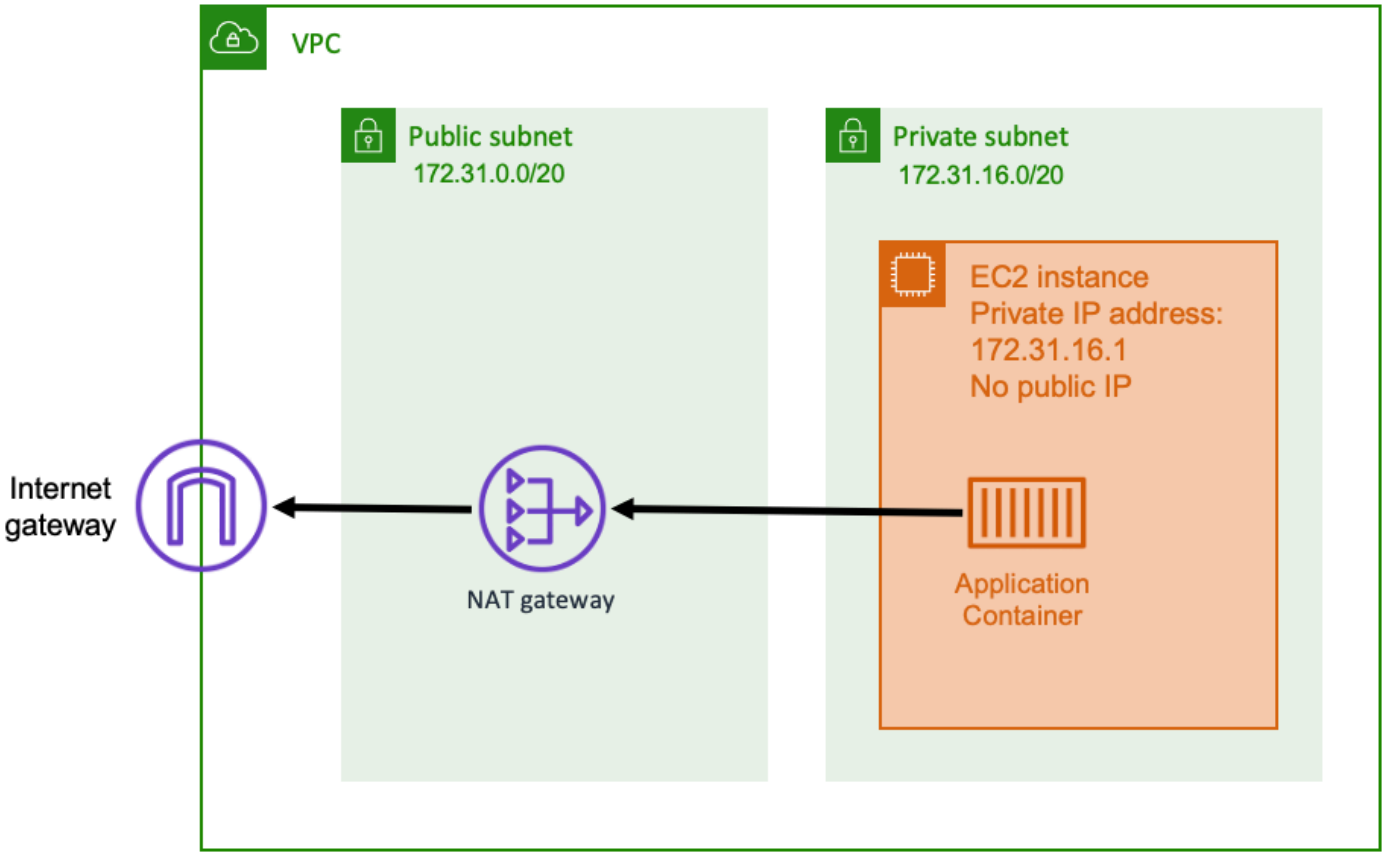
Por exemplo, se seu aplicativo estiver sendo executado no Amazon EC2, verifique se a porta 22 para acesso SSH não está aberta. Caso contrário, sua instância poderia receber tentativas constantes de conexão SSH de bots maciços na internet. Estes bots arrastam através de endereços IP públicos. Depois de encontrar uma porta SSH aberta, eles tentam senhas de força bruta para tentar acessar sua instância. Devido a isso, muitas organizações limitam o uso de sub-redes públicas e preferem ter a maioria, se não todos, de seus recursos dentro de sub-redes privadas.

O uso de sub-redes públicas para redes é adequado para aplicativos públicos que exigem grandes quantidades de largura de banda ou latência mínima. Os casos de uso aplicáveis incluem serviços de streaming de vídeo e jogos.

Essa abordagem de rede é suportada tanto quando você usa o Amazon ECS no Amazon EC2 quanto quando o usa emAWS Fargate.

- Usando o Amazon EC2: você pode executar instâncias do EC2 em uma sub-rede pública. O Amazon ECS usa essas instâncias do EC2 como capacidade de cluster, e todos os contêineres que estão sendo executados nas instâncias podem usar o endereço IP público subjacente do host para redes de saída. Isso se aplica tanto ao`hostebridge`modos de rede. No entanto, o`awsvpc`O modo de rede não fornece ENIs de tarefa com endereços IP públicos. Portanto, eles não podem fazer uso direto de um gateway de internet.
- Usando Fargate — Ao criar seu serviço do Amazon ECS, especifique sub-redes públicas para a configuração de rede do seu serviço e certifique-se de que oAtribuir endereço IP públicoestá habilitada. Cada tarefa Fargate é conectada em rede na sub-rede pública e tem seu próprio endereço IP público para comunicação direta com a Internet.

Usando uma sub-rede privada e um gateway NAT



Usando uma sub-rede privada e um gateway NAT, você pode executar seu aplicativo em contêineres em um host que esteja em uma sub-rede privada. Como tal, esse host tem um endereço IP privado que é roteável dentro de sua VPC, mas não é roteável a partir da Internet. Isso significa que outros hosts dentro da VPC podem fazer conexões com o host usando seu endereço IP privado, mas outros hosts na Internet não podem fazer nenhuma comunicação de entrada para o host.

Com uma sub-rede privada, você pode usar um gateway de conversão de endereços de rede (NAT) para permitir que um host dentro de uma sub-rede privada se conecte à internet. Os hosts na Internet recebem uma conexão de entrada que parece estar vindo do endereço IP público do gateway NAT que está dentro de uma sub-rede pública. O gateway NAT é responsável por servir como uma ponte entre a Internet e a VPC privada. Essa configuração geralmente é preferida por motivos de segurança, pois significa que sua VPC está protegida contra acesso direto por invasores na Internet. Para obter mais informações, consulteGateways NATnoGuia do usuário da Amazon VPC.

Essa abordagem de rede privada é adequada para cenários em que você deseja proteger seus contêineres contra acesso externo direto. Cenários aplicáveis incluem sistemas de processamento

de pagamentos ou contêineres que armazenam dados de usuários e senhas. Você será cobrado para criar e usar um gateway NAT em sua conta. Serão aplicadas taxas de uso por hora do gateway NAT e de processamento de dados. Para fins de redundância, você deve ter um gateway NAT em cada Zona de disponibilidade. Dessa forma, a perda de disponibilidade de uma única Zona de disponibilidade não compromete sua conectividade de saída. Por isso, se você tiver uma pequena carga de trabalho, poderá ser mais econômico usar sub-redes privadas e gateways NAT.

Essa abordagem de rede é compatível tanto ao usar o Amazon ECS no Amazon EC2 quanto ao usá-lo emAWS Fargate.

- Usando o Amazon EC2: você pode executar instâncias do EC2 em uma sub-rede privada. Os contêineres que são executados nesses hosts EC2 usam a rede de hosts subjacente e as solicitações de saída passam pelo gateway NAT.
- Usando Fargate — Ao criar seu serviço do Amazon ECS, especifique sub-redes privadas para a configuração de rede do seu serviço e não ative a opçãoAtribuir endereço IP públicoOpção. Cada tarefa do Fargate é hospedada em uma sub-rede privada. Seu tráfego de saída é roteado por qualquer gateway NAT associado a essa sub-rede privada.

Receber conexões de entrada pela Internet

Se você executar um serviço público, você deve aceitar o tráfego de entrada da Internet. Por exemplo, seu site público deve aceitar solicitações HTTP de entrada de navegadores. Nesse caso, outros hosts na Internet também devem iniciar uma conexão de entrada com o host do seu aplicativo.

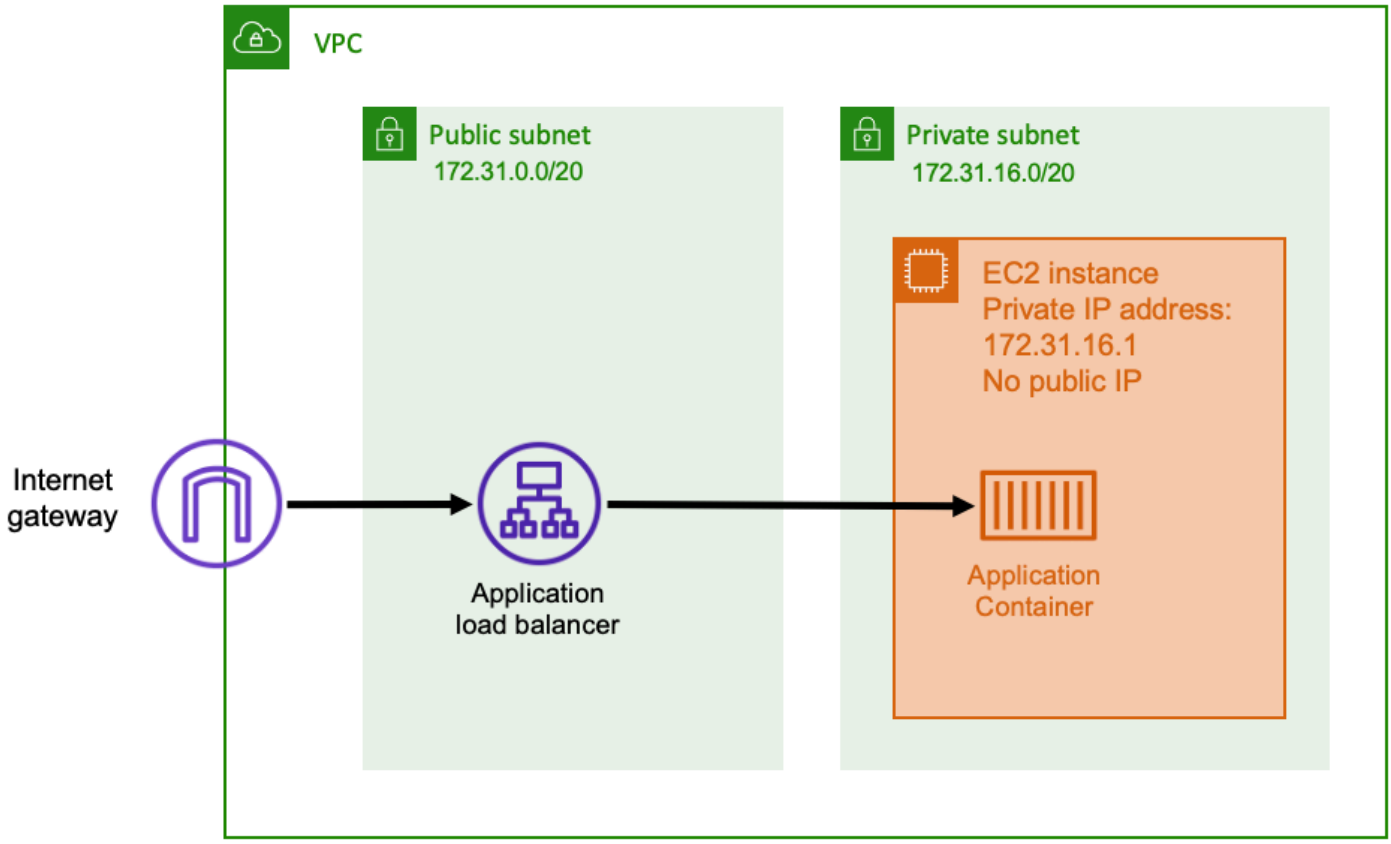
Uma abordagem para esse problema é iniciar seus contêineres em hosts que estão em uma sub-rede pública com um endereço IP público. No entanto, não recomendamos isso para aplicativos de grande escala. Para isso, uma melhor abordagem é ter uma camada de entrada escalável que fica entre a internet e seu aplicativo. Para essa abordagem, você pode usar qualquer um dosAWSlistados nesta seção como uma entrada.

Tópicos

- Balanceador de carga de aplicações
- Load balancer de rede
- API HTTP do Amazon API Gateway

Balanceador de carga de aplicações

Um Application Load Balancer funciona na camada de aplicativo. É a sétima camada do modelo Open Systems Interconnection (OSI). Isso torna um Application Load Balancer adequado para serviços HTTP públicos. Se você tiver um site ou uma API REST HTTP, um Application Load Balancer é um load balancer adequado para essa carga de trabalho. Para obter mais informações, consulteO que é Application Load Balancer?noGuia do usuário para Application Load Balancers.



Com essa arquitetura, você cria um Application Load Balancer em uma sub-rede pública para que ele tenha um endereço IP público e possa receber conexões de entrada da Internet. Quando o Application Load Balancer recebe uma conexão de entrada ou, mais especificamente, uma solicitação HTTP, ele abre uma conexão com o aplicativo usando seu endereço IP privado. Em seguida, ele encaminha a solicitação através da conexão interna.

Um Application Load Balancer tem as seguintes vantagens.

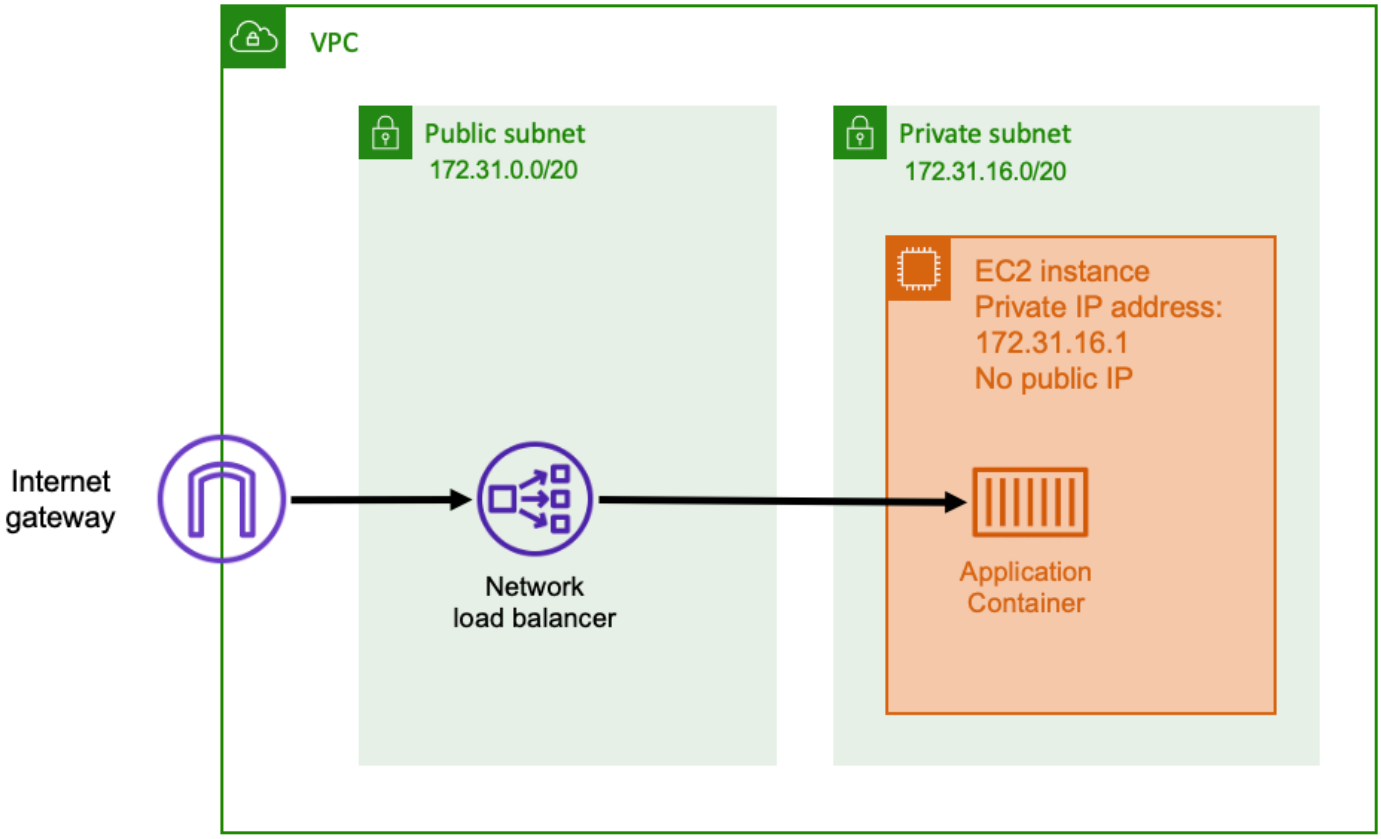
- Terminação SSL/TLS — um Application Load Balancer pode sustentar a comunicação HTTPS segura e certificados para comunicações com clientes. Opcionalmente, ele pode encerrar a

conexão SSL no nível do balanceador de carga para que você não precise manipular certificados em seu próprio aplicativo.

- Roteamento avançado — Um Application Load Balancer pode ter vários nomes de host DNS. Ele também tem recursos avançados de roteamento para enviar solicitações HTTP de entrada para diferentes destinos com base em métricas como o nome do host ou o caminho da solicitação. Isso significa que você pode usar um único Application Load Balancer como a entrada para muitos serviços internos diferentes, ou até mesmo microsserviços em caminhos diferentes de uma API REST.
- Suporte gRPC e websockets — Um Application Load Balancer pode lidar com mais do que apenas HTTP. Ele também pode balancear a carga de serviços baseados em gRPC e websocket, com suporte a HTTP/2.
- Segurança — Um Application Load Balancer ajuda a proteger seu aplicativo contra tráfego mal-intencionado. Ele inclui recursos como mitigações de sincronização HTTP e é integrado ao AWS Web Application Firewall (AWS WAF).AWS WAFpode filtrar ainda mais o tráfego mal-intencionado que pode conter padrões de ataque, como injeção de SQL ou script entre sites.

Load balancer de rede

Um Network Load Balancer funciona na quarta camada do modelo Open Systems Interconnection (OSI). Ele é adequado para protocolos ou cenários não-HTTP em que a criptografia de ponta a ponta é necessária, mas não tem os mesmos recursos específicos de HTTP de um Application Load Balancer. Portanto, um Network Load Balancer é mais adequado para aplicativos que não usam HTTP. Para obter mais informações, consulteO que é um Network Load Balancer?noGuia do usuário para Network Load Balancers.



Quando um Network Load Balancer é usado como entrada, ele funciona de forma semelhante a um Application Load Balancer. Isso ocorre porque ele é criado em uma sub-rede pública e tem um endereço IP público que pode ser acessado pela Internet. Em seguida, o Network Load Balancer abre uma conexão com o endereço IP privado do host que está executando o contêiner e envia os pacotes do lado público para o lado privado.

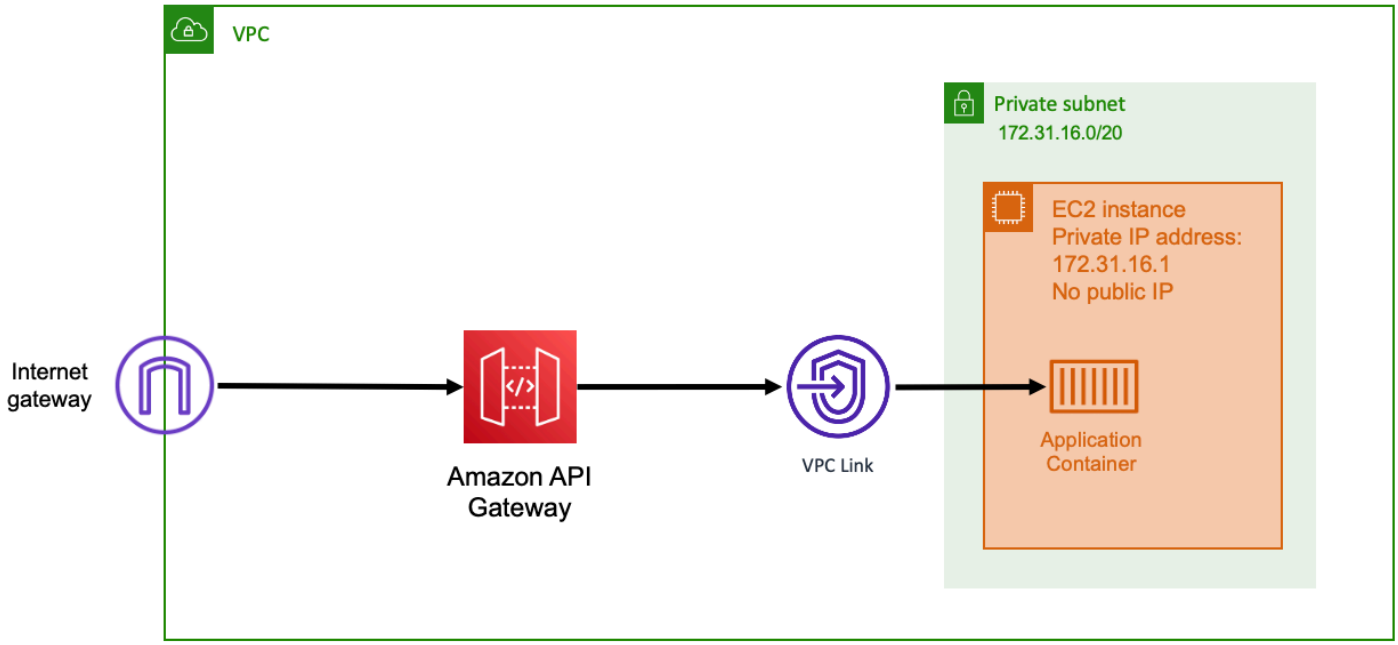
Como o Network Load Balancer opera em um nível inferior da pilha de rede, ele não tem o mesmo conjunto de recursos que o Application Load Balancer possui. No entanto, ele tem os seguintes recursos importantes.

- Criptografia de ponta a ponta — Como um Network Load Balancer opera na quarta camada do modelo OSI, ele não lê o conteúdo dos pacotes. Isso o torna adequado para comunicações de balanceamento de carga que precisam de criptografia de ponta a ponta.
- Criptografia TLS — Além da criptografia de ponta a ponta, o Network Load Balancer também pode encerrar conexões TLS. Dessa forma, seus aplicativos de back-end não precisam implementar seu próprio TLS.

- Suporte a UDP — como um Network Load Balancer opera na quarta camada do modelo OSI, ele é adequado para cargas de trabalho não HTTP e protocolos diferentes do TCP.

API HTTP do Amazon API Gateway

A API HTTP do Amazon API Gateway é um servidor menos entrada que é adequado para aplicativos HTTP com intermitências repentinas em volumes de solicitação ou volumes de solicitação baixos. Para obter mais informações, consulteO que é o Amazon API Gateway?noGuia do desenvolvedor do API Gateway.



O modelo de preço para o Application Load Balancer e o Network Load Balancer incluem um preço por hora para manter os load balancers disponíveis para aceitar conexões de entrada em todos os momentos. Em contraste, o API Gateway cobra por cada solicitação separadamente. Isto tem o efeito de que, se não forem apresentados pedidos, não haverá encargos. Em cargas de tráfego alto, um Application Load Balancer ou Network Load Balancer pode lidar com um volume maior de solicitações a um preço por solicitação mais barato do que o API Gateway. No entanto, se você tiver um número baixo de solicitações no geral ou tiver períodos de baixo tráfego, o preço cumulativo para usar o API Gateway deve ser mais econômico do que pagar uma cobrança horária para manter um load balancer que está sendo subutilizado.

Funções do API Gateway usando um link da VPC que permite que oAWSpara se conectar a hosts dentro da sub-rede privada de sua VPC, usando seu endereço IP privado. Ele pode detectar esses

endereços IP privados observandoAWS Cloud Mapregistros de descoberta de serviço que são gerenciados pela descoberta de serviço do Amazon ECS.

O API Gateway é compatível com os seguintes recursos.

- Encerramento SSL/TLS
- Roteando diferentes caminhos HTTP para diferentes microsserviços de back-end

Além dos recursos anteriores, o API Gateway também suporta o uso de autorizadores Lambda personalizados que você pode usar para proteger sua API de uso não autorizado. Para obter mais informações, consulteObservações de campo: APIs baseadas em contêiner sem servidor com o Amazon ECS e o Amazon API Gateway.

Escolhendo um modo de rede

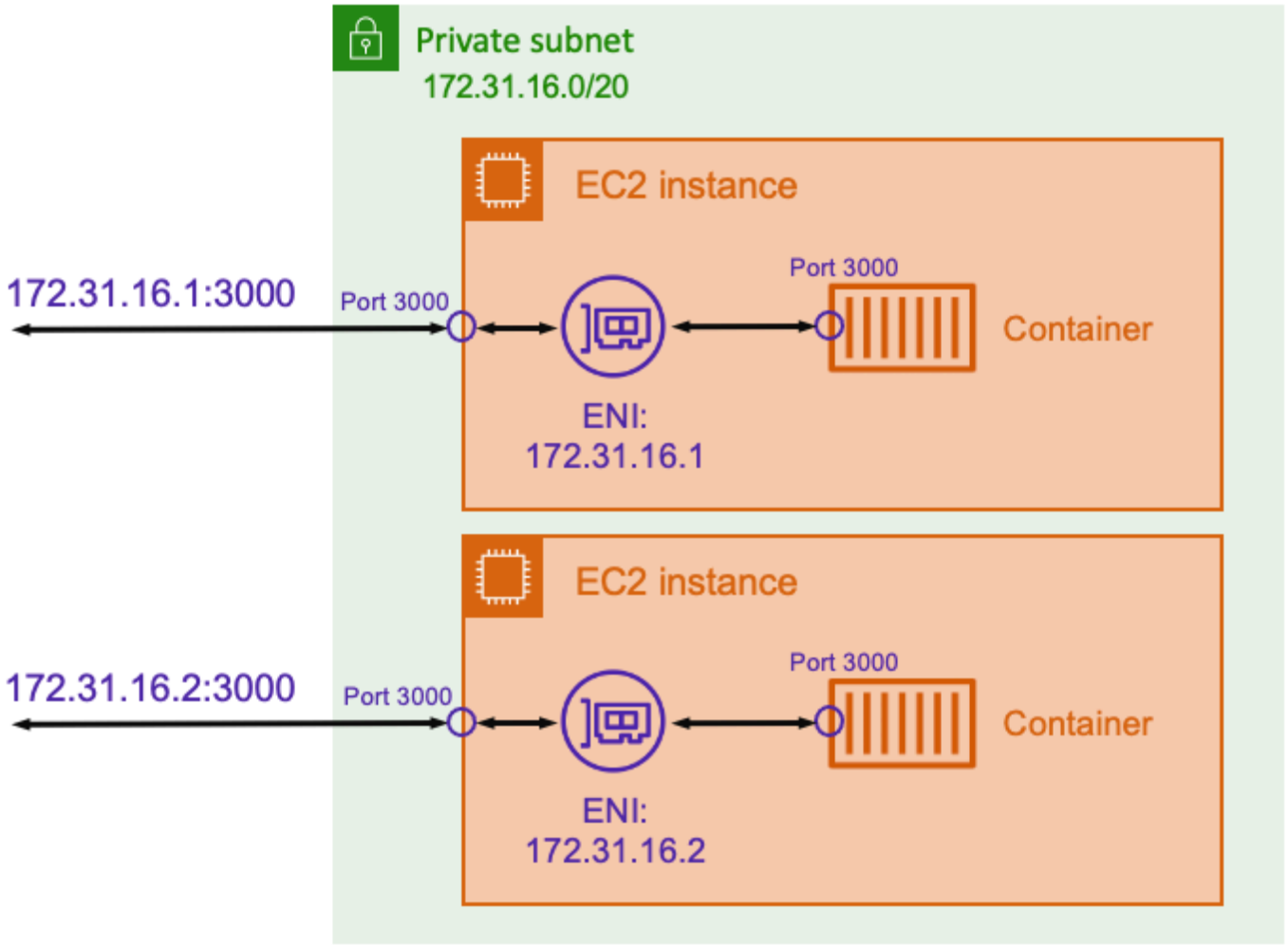
As abordagens mencionadas anteriormente para arquitetar conexões de rede de entrada e saída podem ser aplicadas a qualquer uma de suas cargas de trabalho noAWS, mesmo que não estejam dentro de um contêiner. Ao executar contêineres noAWS, você precisa considerar outro nível de rede. Uma das principais vantagens do uso de contêineres é que você pode empacotar vários contêineres em um único host. Ao fazer isso, você precisa escolher como deseja conectar os contêineres que estão sendo executados no mesmo host. A seguir estão as opções para escolher.

Tópicos

- Modo de host
- Modo de ponte
- Modo AWSVPC

Modo de host

O`host`é o modo de rede mais básico compatível com o Amazon ECS. Usando o modo host, a rede do contêiner é vinculada diretamente ao host subjacente que está executando o contêiner.



Suponha que você esteja executando um contêiner Node.js com um aplicativo Express que escuta na porta`3000`semelhante ao ilustrado no diagrama anterior. Quando o`host`for usado, o contêiner recebe tráfego na porta 3000 usando o endereço IP da instância do host subjacente do Amazon EC2. Não recomendamos o uso deste modo.

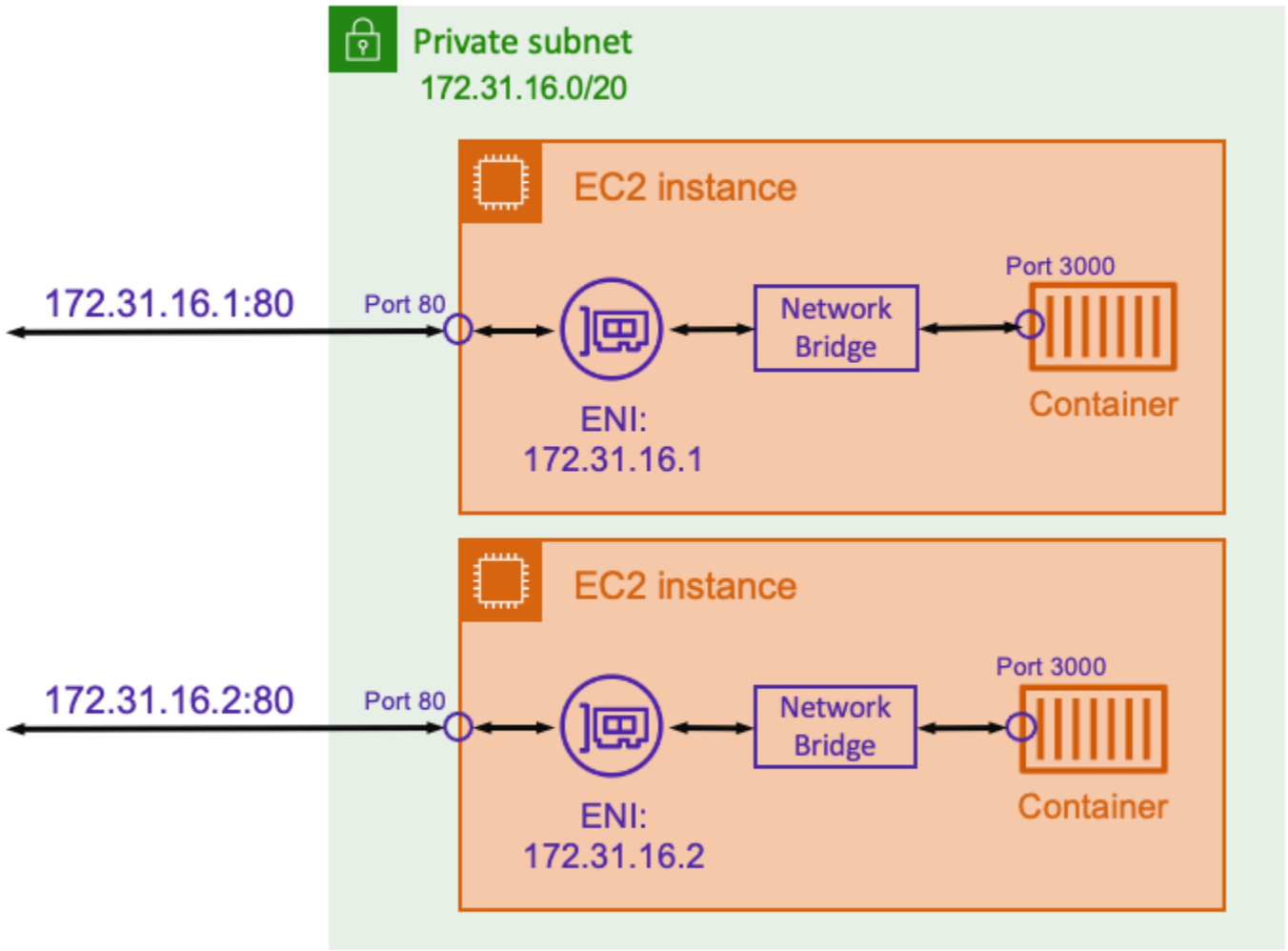
Há desvantagens significativas em usar esse modo de rede. Você não pode executar mais de uma única instanciação de uma tarefa em cada host. Isso ocorre porque somente a primeira tarefa pode ser vinculada à porta necessária na instância do Amazon EC2. Também não há como remapear uma porta de contêiner quando ela está usando`host`Modo de rede. Por exemplo, se um aplicativo precisar escutar um número de porta específico, você não poderá remapear o número da porta diretamente. Em vez disso, você deve gerenciar quaisquer conflitos de porta alterando a configuração do aplicativo.

Há também implicações de segurança ao usar o`host`Modo de rede. Esse modo permite que os contêineres representem o host e permite que os contêineres se conectem a serviços de rede de loopback privados no host.

O`host`O modo de rede só é compatível com tarefas do Amazon ECS hospedadas em instâncias do Amazon EC2. Não há suporte ao uso do Amazon ECS no Fargate.

Modo de ponte

com`bridge`, você está usando uma ponte de rede virtual para criar uma camada entre o host e a rede do contêiner. Dessa forma, você pode criar mapeamentos de porta que remapeiam uma porta de host para uma porta de contêiner. Os mapeamentos podem ser estáticos ou dinâmicos.

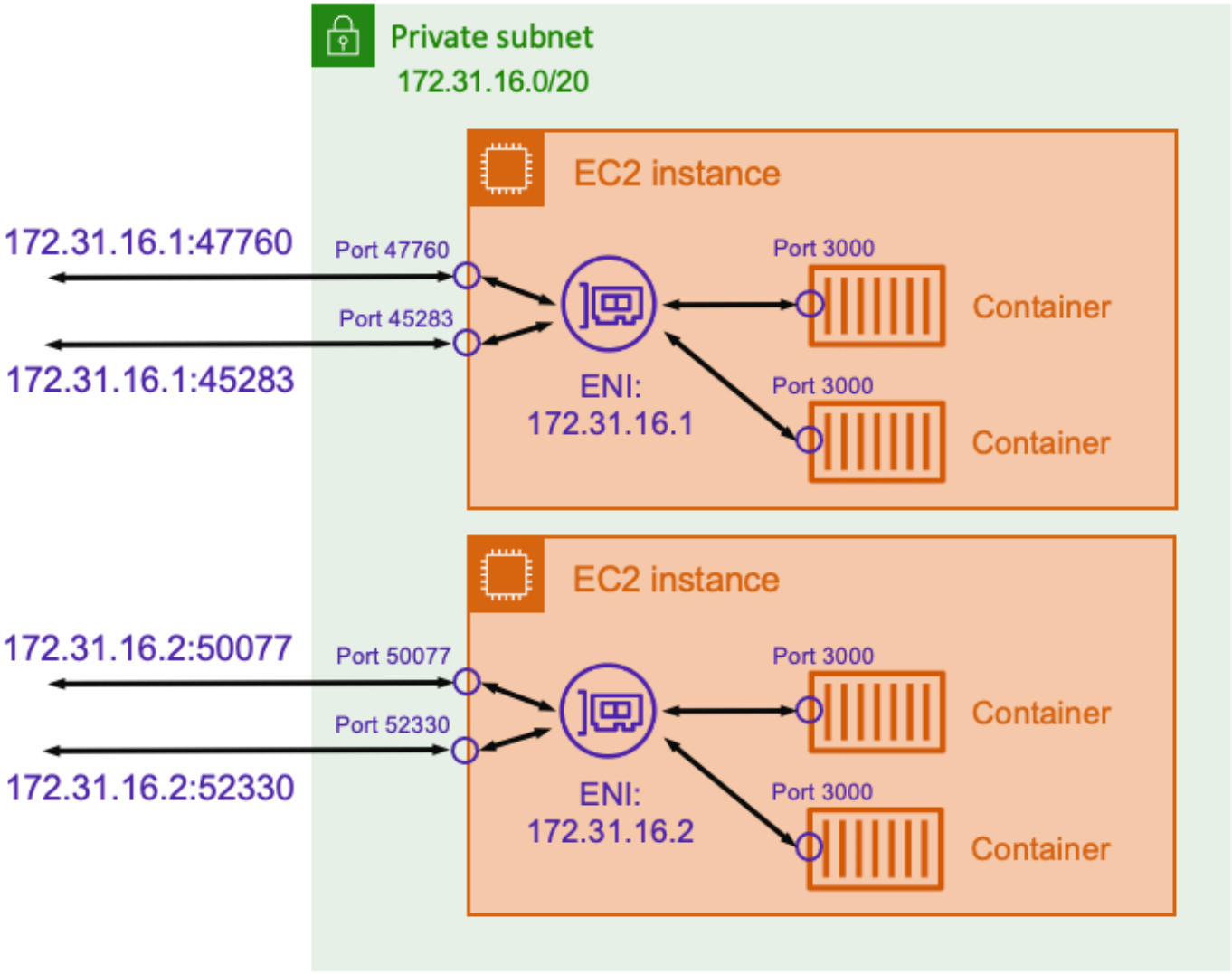


Com um mapeamento de porta estática, você pode definir explicitamente qual porta host deseja mapear para uma porta de contêiner. Usando o exemplo acima, port`80`no host está sendo mapeado

para a porta`3000`No contêiner. Para se comunicar com o aplicativo em contêiner, você envia tráfego para a porta`80`O endereço IP da instância do Amazon EC2. Do ponto de vista do aplicativo em contêineres, ele vê que o tráfego de entrada na porta`3000`.

Se você quiser apenas alterar a porta de tráfego, então mapeamentos de porta estáticos são adequados. No entanto, isso ainda tem a mesma desvantagem que usar o`host`Modo de rede. Você não pode executar mais de uma única instanciação de uma tarefa em cada host. Isso ocorre porque um mapeamento de porta estática só permite que um único contêiner seja mapeado para a porta 80.

Para resolver esse problema, considere usar o`bridge`O modo de rede com um mapeamento de porta dinâmico, conforme mostrado no diagrama a seguir.



Ao não especificar uma porta de host no mapeamento de portas, você pode fazer com que o Docker escolha uma porta aleatória e não utilizada no intervalo de portas efêmeras e a atribua como porta de host pública para o contêiner. Por exemplo, o aplicativo Node.js escutando na porta`3000`no contêiner pode ser atribuída uma porta aleatória de alto número, como`47760`no host do Amazon EC2. Isso significa que você pode executar várias cópias desse contêiner no host. Além disso, cada contêiner pode ser atribuído sua própria porta no host. Cada cópia do contêiner recebe tráfego na porta`3000`. No entanto, os clientes que enviam tráfego para esses contêineres usam as portas de host atribuídas aleatoriamente.

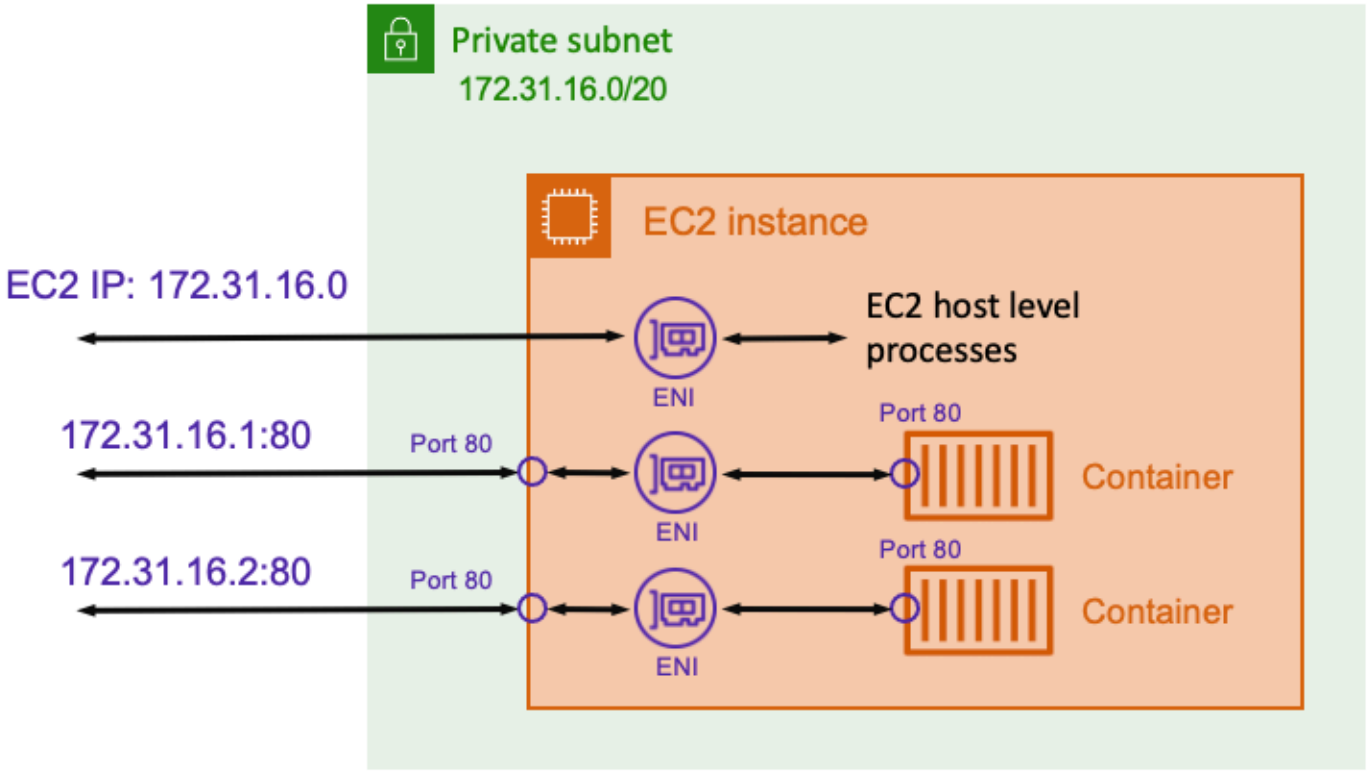
O Amazon ECS ajuda você a acompanhar as portas atribuídas aleatoriamente para cada tarefa. Ele faz isso atualizando automaticamente os grupos de destino do balanceador de carga eAWS Cloud Mappara ter a lista de endereços IP de tarefas e portas. Isso facilita o uso de serviços que operam usando`bridge`com portas dinâmicas.

No entanto, uma desvantagem de usar o`bridge`é que é difícil bloquear serviço para comunicação de serviço. Como os serviços podem ser atribuídos a qualquer porta aleatória e não utilizada, é necessário abrir amplos intervalos de portas entre hosts. No entanto, não é fácil criar regras específicas para que um determinado serviço só possa se comunicar com um outro serviço específico. Os serviços não têm portas específicas para utilizar para regras de rede de grupos de segurança.

O`bridge`O modo de rede só é compatível com tarefas do Amazon ECS hospedadas em instâncias do Amazon EC2. Não há suporte ao uso do Amazon ECS no Fargate.

Modo AWSVPC

Com o`awsvpc`, o Amazon ECS cria e gerencia uma interface de rede elástica (ENI) para cada tarefa e cada tarefa recebe seu próprio endereço IP privado dentro da VPC. Este ENI é separado dos anfitriões subjacentes ENI. Se uma instância do Amazon EC2 estiver executando várias tarefas, o ENI de cada tarefa também será separado.



No exemplo anterior, a instância do Amazon EC2 é atribuída a um ENI. O ENI representa o endereço IP da instância do EC2 usada para comunicações de rede no nível do host. Cada tarefa tem também um ENI correspondente e um endereço IP privado. Como cada ENI é separado, cada contêiner pode se vincular à porta`80`sobre a tarefa ENI. Portanto, você não precisa acompanhar os números das portas. Em vez disso, você pode enviar tráfego para a porta`80`No endereço IP da tarefa ENI.

A vantagem de usar o`awsvpc`é que cada tarefa tem um grupo de segurança separado para permitir ou negar tráfego. Isso significa que você tem maior flexibilidade para controlar as comunicações entre tarefas e serviços em um nível mais granular. Você também pode configurar uma tarefa para negar tráfego de entrada de outra tarefa localizada no mesmo host.

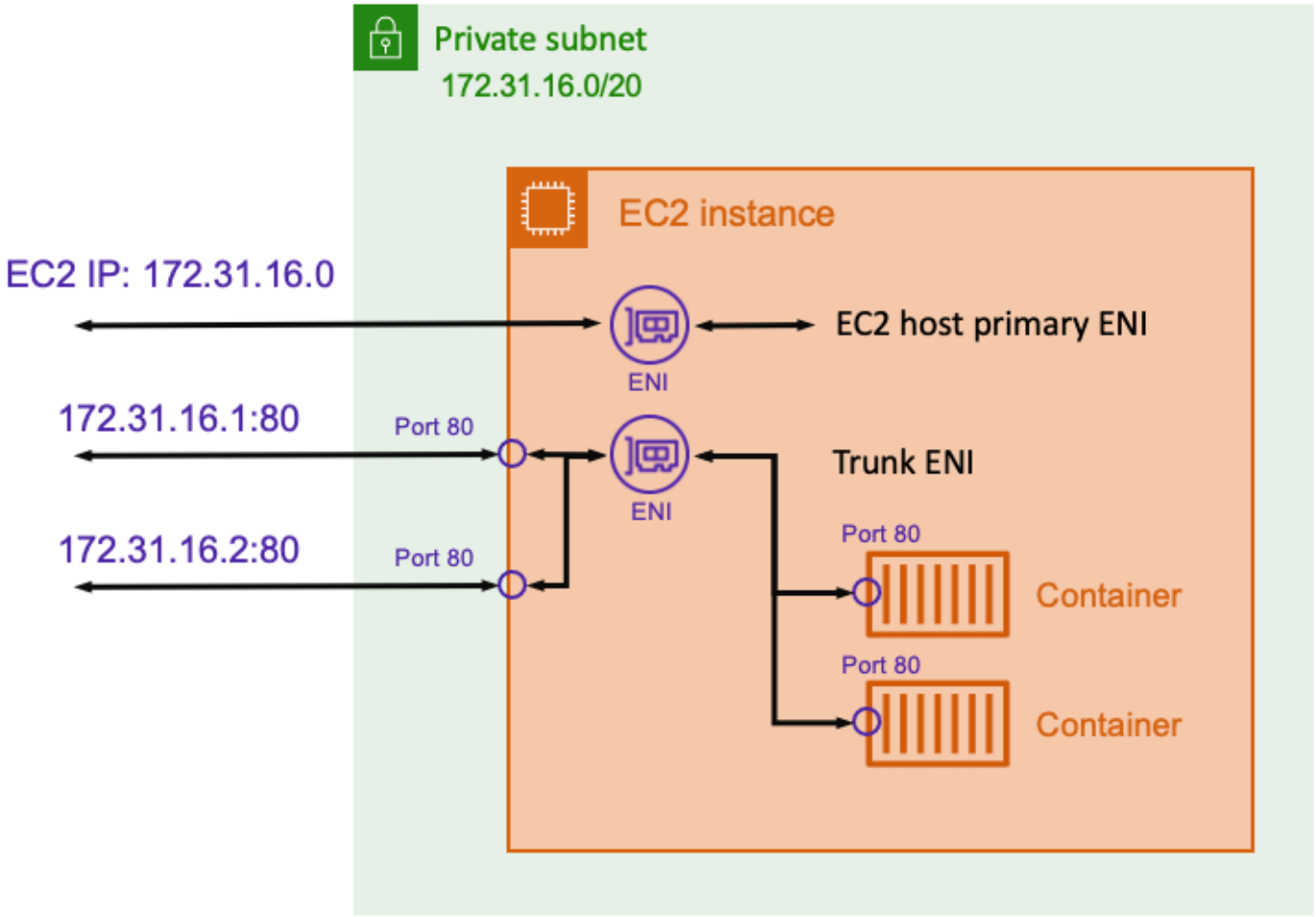
O`awsvpc`é compatível com tarefas do Amazon ECS hospedadas no Amazon EC2 e no Fargate. Tenha em atenção que, ao usar Fargate, o`awsvpc`é necessário.

Ao usar o`awsvpc`há alguns desafios que você deve estar atento.

Aumento da densidade da tarefa com Entroncamento ENI

A maior desvantagem de usar o`awsvpc`O modo de rede com tarefas hospedadas em instâncias do Amazon EC2 é que as instâncias EC2 têm um limite para o número de ENIs que podem ser

anexadas a elas. Isso limita quantas tarefas você pode colocar em cada instância. O Amazon ECS fornece o recurso de entroncamento ENI que aumenta o número de ENIs disponíveis para obter mais densidade de tarefas.



Ao usar o entroncamento ENI, dois anexos ENI são usados por padrão. O primeiro é o ENI principal da instância, que é usado para qualquer processo em nível de host. O segundo é o tronco ENI, que o Amazon ECS cria. Esse recurso só é compatível com tipos de instância do Amazon EC2 específicos.

Considere este exemplo. Sem entroncamento ENI, um`c5.large`que tem duas vCPUs só pode hospedar duas tarefas. No entanto, com o entroncamento ENI, um`c5.large`que tem duas vCPUs pode hospedar até dez tarefas. Cada tarefa tem um endereço IP e um grupo de segurança diferentes. Para obter mais informações sobre os tipos de instância disponíveis e sua densidade, consulteTipos de instância do Amazon EC2 compatíveisnoAmazon Elastic Container Service Guia do desenvolvedor.

O entroncamento ENI não tem impacto no desempenho do tempo de execução em termos de latência ou largura de banda. No entanto, aumenta o tempo de inicialização da tarefa. Você deve garantir que, se o entroncamento ENI for usado, suas regras de dimensionamento automático e outras cargas de trabalho que dependem do tempo de inicialização da tarefa ainda funcionarão como você espera.

Para obter mais informações, consulteTroncamento da interface de redenoAmazon Elastic Container Service Guia do desenvolvedor.

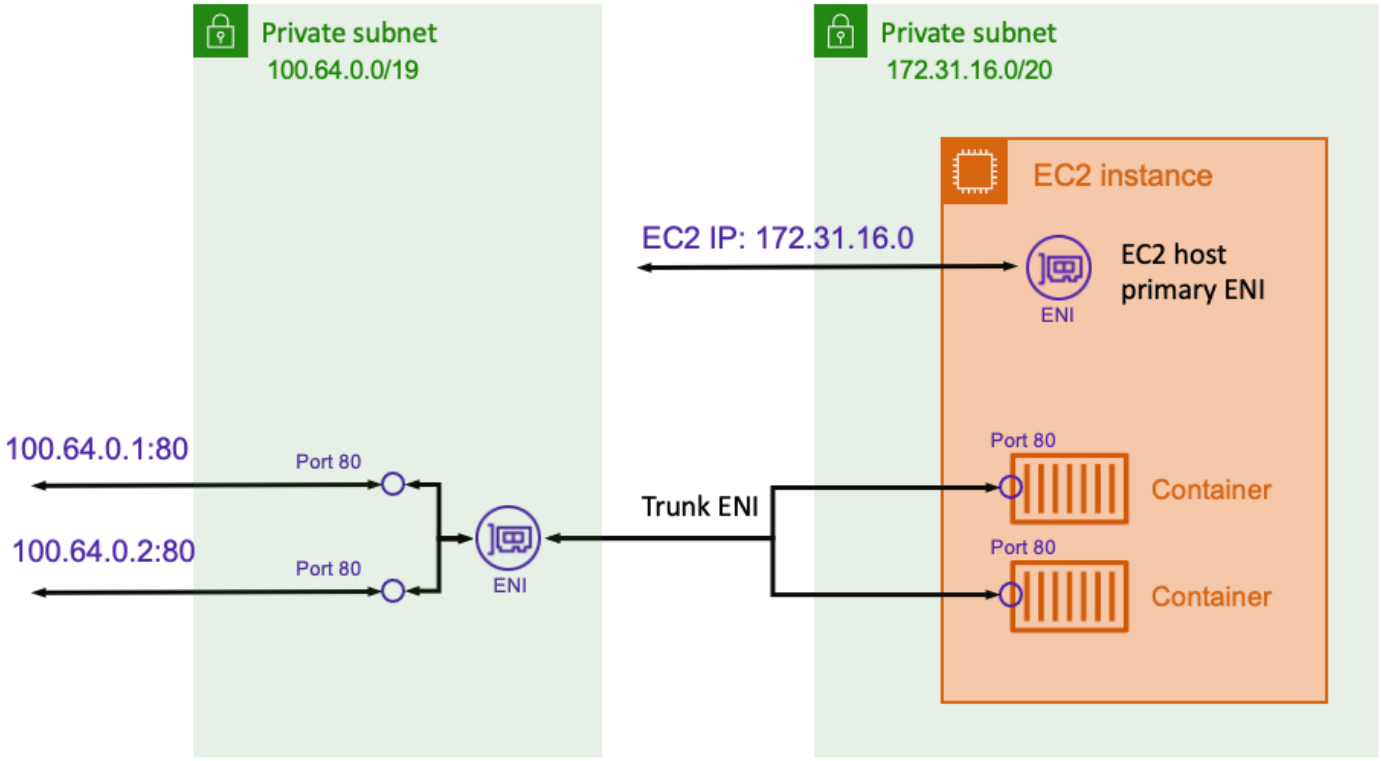
Prevenir o esgotamento de endereços IP

Ao atribuir um endereço IP separado a cada tarefa, você pode simplificar sua infraestrutura geral e manter grupos de segurança que oferecem um ótimo nível de segurança. No entanto, essa configuração pode levar ao esgotamento do IP.

A VPC padrão em seuAWStem sub-redes pré-provisionadas que têm um`/20`Intervalo CIDR. Isso significa que cada sub-rede tem 4.091 endereços IP disponíveis. Observe que vários endereços IP dentro do`/20`O intervalo são reservados para uso específico da AWS. Considere este exemplo. Você distribui seus aplicativos em três sub-redes em três zonas de disponibilidade para alta disponibilidade. Nesse caso, você pode usar aproximadamente 12.000 endereços IP nas três sub-redes.

Usando entroncamento ENI, cada instância do Amazon EC2 executada requer dois endereços IP. Um endereço IP é usado para o ENI primário e o outro endereço IP é usado para o ENI tronco. Cada tarefa do Amazon ECS na instância requer um endereço IP. Se você estiver iniciando uma carga de trabalho extremamente grande, poderá ficar sem endereços IP disponíveis. Isso pode resultar em falhas de execução do Amazon EC2 ou falhas de execução de tarefas. Esses erros ocorrem porque os ENIs não podem adicionar endereços IP dentro da VPC se não houver endereços IP disponíveis.

Ao usar o`awsvpc`, você deve avaliar seus requisitos de endereço IP e garantir que seus intervalos de CIDR de sub-rede atendam às suas necessidades. Se você já começou a usar uma VPC que tem sub-redes pequenas e começa a ficar sem espaço de endereço, você pode adicionar uma sub-rede secundária.



Usando o entroncamento ENI, o CNI da Amazon VPC pode ser configurado para usar ENIs em um espaço de endereço IP diferente do host. Ao fazer isso, você pode fornecer ao seu host do Amazon EC2 e às suas tarefas diferentes intervalos de endereços IP que não se sobrepõem. No diagrama de exemplo, o endereço IP do host do EC2 está em uma sub-rede que tem o`172.31.16.0/20`Intervalo de IP No entanto, as tarefas que estão em execução no anfitrião são atribuídos endereços IP no`100.64.0.0/19`Intervalo. Usando dois intervalos de IP independentes, você não precisa se preocupar com tarefas consumindo muitos endereços IP e não deixando endereços IP suficientes para instâncias.

Usando o modo de pilha dupla IPv6

O`awsvpc`é compatível com VPCs configuradas para o modo de pilha dupla IPv6. Uma VPC usando o modo de pilha dupla pode se comunicar por IPv4, IPv6 ou ambos. Cada sub-rede na VPC pode ter um intervalo IPv4 CIDR e um intervalo IPv6 CIDR. Para obter mais informações, consulteEndereçamento IP na sua VPCnoGuia do usuário da Amazon VPC.

Não é possível desativar o suporte IPv4 para sua VPC e sub-redes para resolver problemas de exaustão do IPv4. No entanto, com o suporte IPv6, você pode usar alguns novos recursos, especificamente o gateway da Internet somente de saída. Um gateway de internet somente de saída

permite que tarefas usem seu endereço IPv6 roteável publicamente para iniciar conexões de saída à internet. Mas o gateway da Internet somente de saída não permite conexões da Internet. Para obter mais informações, consulteGateways da Internet apenas de saídanoGuia do usuário da Amazon VPC.

Conexão aoAWSserviços de dentro de sua VPC

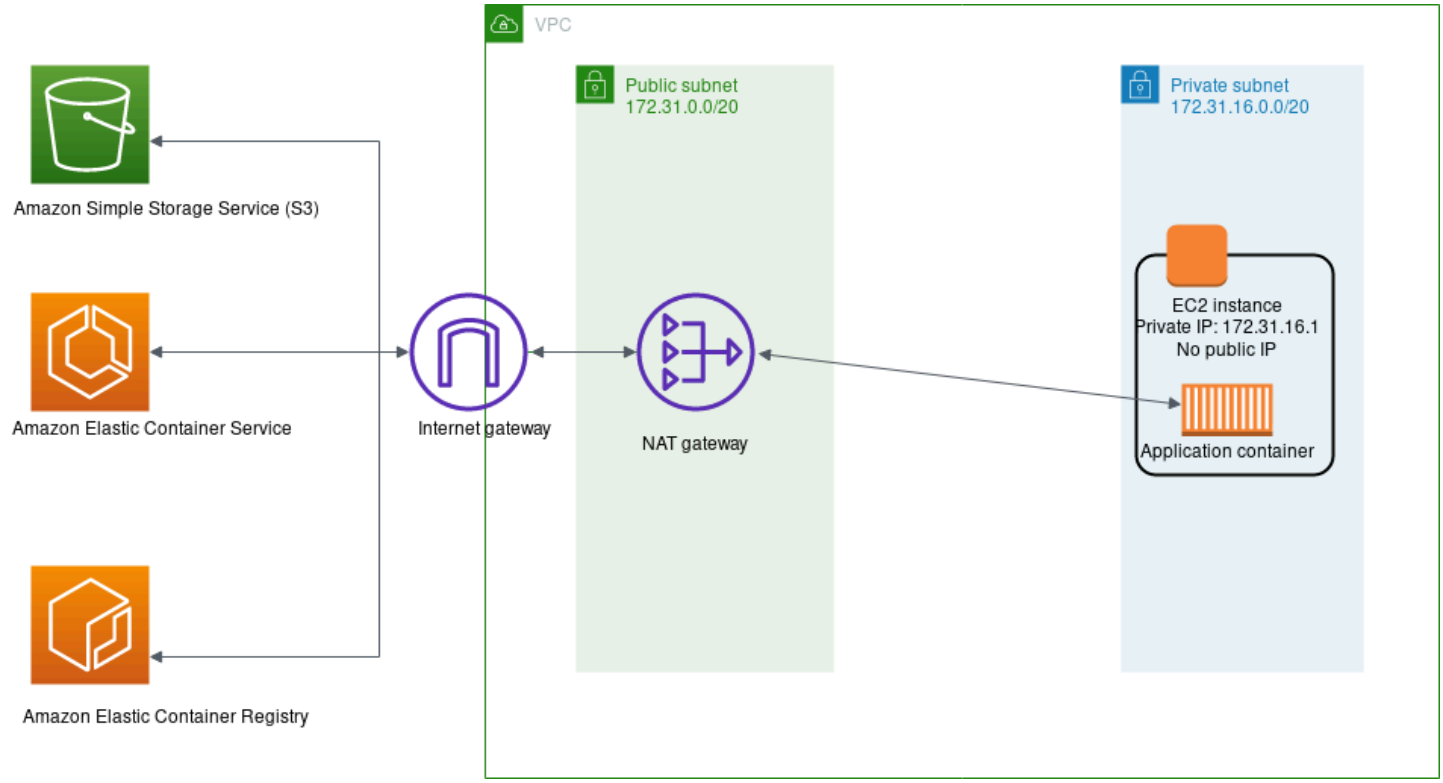
Para que o Amazon ECS funcione corretamente, o agente de contêiner ECS executado em cada host deve se comunicar com o plano de controle do Amazon ECS. Se você estiver armazenando suas imagens de contêiner no Amazon ECR, os hosts do Amazon EC2 devem se comunicar com o endpoint de serviço do Amazon ECR e com o Amazon S3, onde as camadas de imagem são armazenadas. Se você usar outrosAWSpara seu aplicativo em contêineres, como dados persistentes armazenados no DynamoDB, verifique se esses serviços também têm o suporte de rede necessário.

Tópicos

- gateway NAT
- AWS PrivateLink

gateway NAT

Usar um gateway NAT é a maneira mais fácil de garantir que suas tarefas do Amazon ECS possam acessar outrosAWSServiços da . Para obter mais informações sobre essa abordagem, consulteUsando uma sub-rede privada e um gateway NAT.



A seguir estão as desvantagens de usar essa abordagem:

- Não é possível limitar com quais destinos o gateway NAT pode se comunicar. Você também não pode limitar quais destinos seu pneu backend pode se comunicar sem interromper todas as comunicações de saída de sua VPC.
- Os gateways NAT cobram por cada GB de dados que passam. Se você usar o gateway NAT para baixar arquivos grandes do Amazon S3 ou fazer um alto volume de consultas de banco de dados para o DynamoDB, será cobrado por cada GB de largura de banda. Além disso, os gateways NAT oferecem suporte a 5 Gbps de largura de banda e escalam automaticamente até 45 Gbps. Se você rotear através de um único gateway NAT, os aplicativos que exigem conexões de largura de banda muito alta podem encontrar restrições de rede. Como solução alternativa, você pode dividir sua carga de trabalho em várias sub-redes e fornecer a cada sub-rede seu próprio gateway NAT.

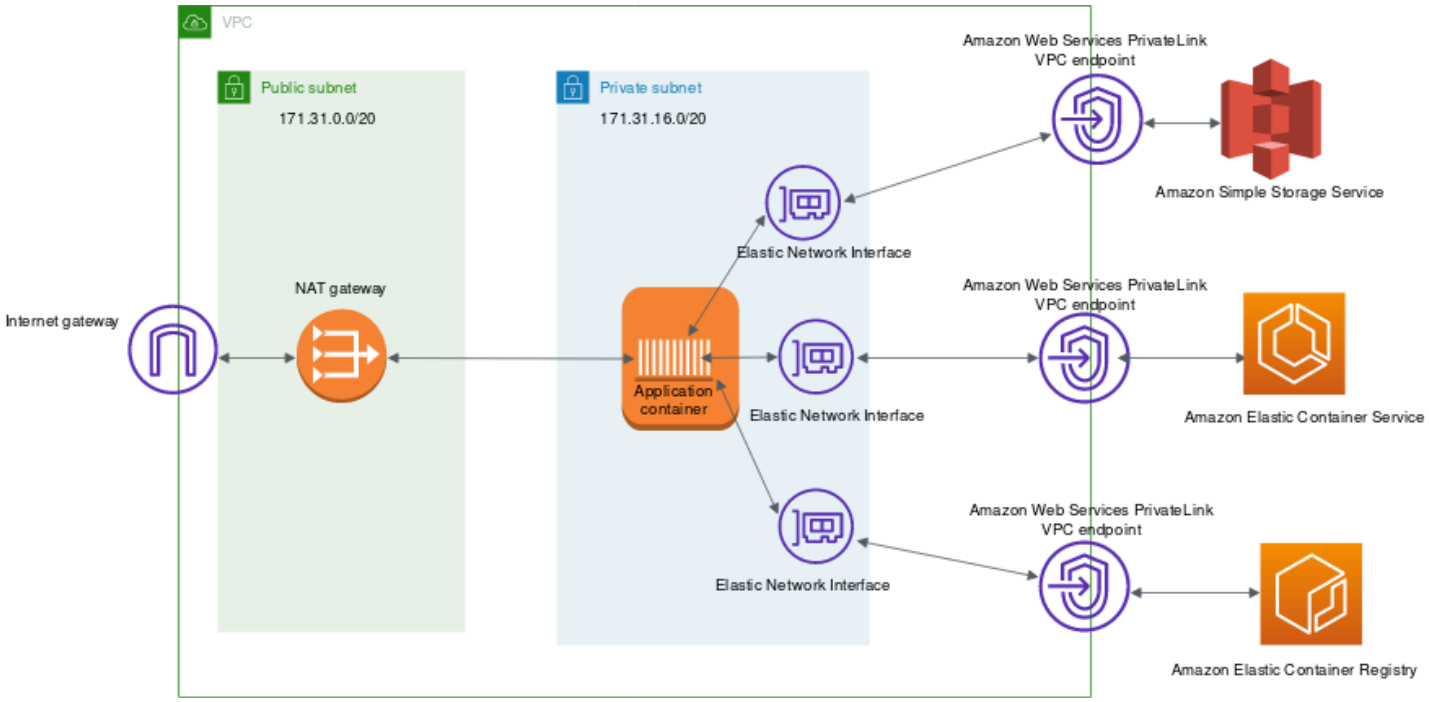
AWS PrivateLink

AWS PrivateLinkfornece conectividade privada entre VPCs,AWSe suas redes locais sem expor seu tráfego à Internet pública.

Uma das tecnologias usadas para fazer isso é o endpoint da VPC. Um VPC endpoint permite conexões privadas entre sua VPC e o suporteAWSServiços do VPC endpoint. O tráfego entre a sua

VPC e os outros serviços não sai da rede da Amazon. Um VPC endpoint não exige um gateway da Internet, gateway privado virtual, dispositivo NAT, conexão VPN ouAWS Direct Connectconexão. As instâncias do Amazon EC2 na sua VPC não exigem que endereços IP públicos se comuniquem com recursos no serviço.

O diagrama a seguir mostra como a comunicação com oAWSfunciona quando você está usando endpoints de VPC em vez de um gateway de Internet.AWS PrivateLinkprovisiona interfaces elásticas de rede (ENIs) dentro da sub-rede, e regras de roteamento VPC são usadas para enviar qualquer comunicação para o nome do host do serviço através do ENI, diretamente para o destinoAWSserviçoServiço do Esse tráfego não precisa mais usar o gateway NAT ou o gateway de Internet.



Veja a seguir alguns dos endpoints comuns da VPC usados com o serviço Amazon ECS.

- VPC endpoint de gateway do S3
- Endpoint de VPC do DynamoDB
- Endpoint de VPC do Amazon ECS
- VPC endpoint do Amazon ECR

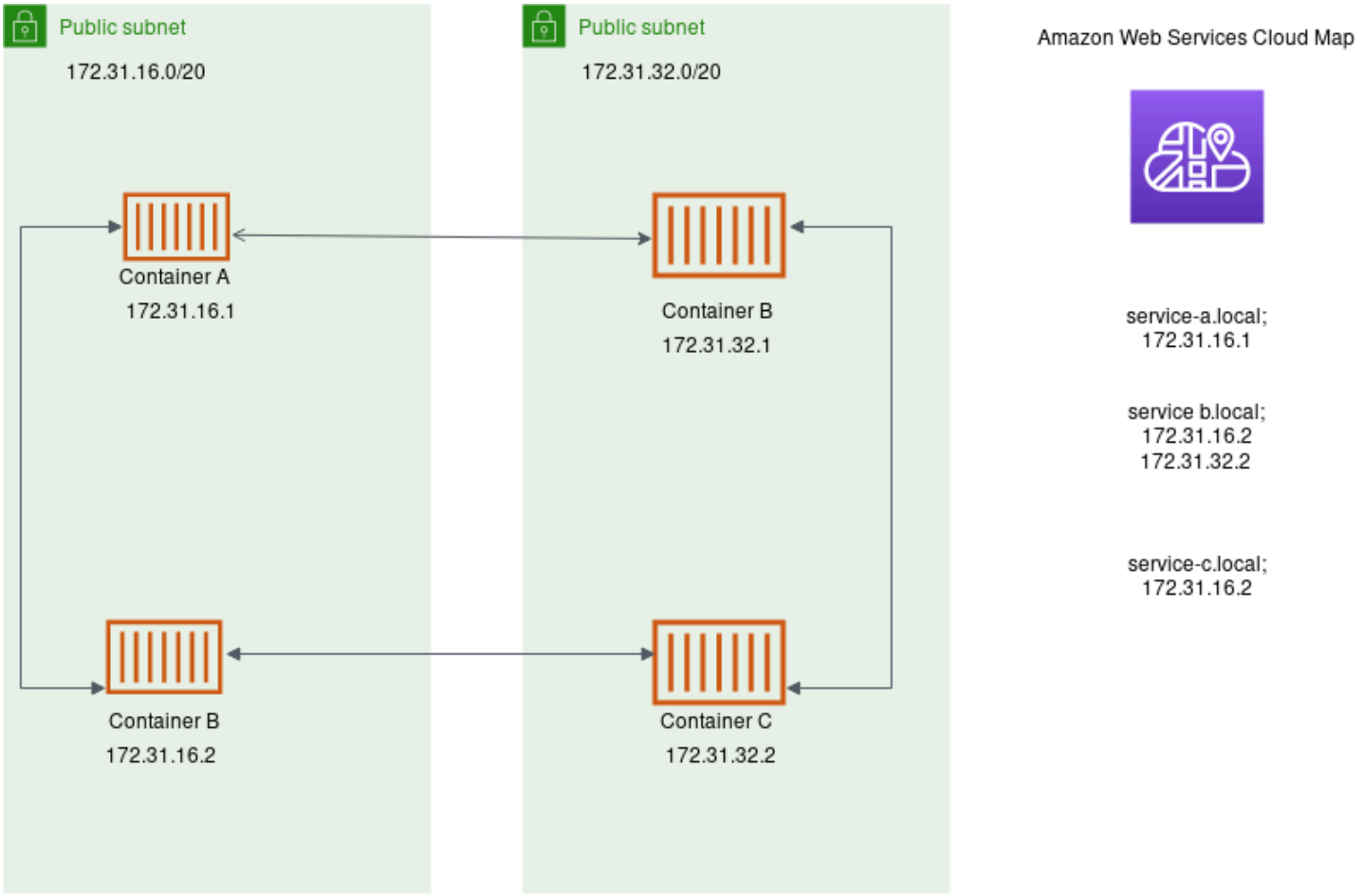
Muitos outrosAWSOs serviços oferecem suporte a VPC endpoints. Se você fizer uso pesado de qualquerAWS, você deve procurar a documentação específica para esse serviço e como criar um endpoint da VPC para esse tráfego.

Rede entre serviços do Amazon ECS em uma VPC

Usando contêineres do Amazon ECS em uma VPC, você pode derramar aplicativos monolíticos em partes separadas que podem ser implantadas e dimensionadas independentemente em um ambiente seguro. No entanto, pode ser um desafio garantir que todas essas peças, dentro e fora de uma VPC, possam se comunicar entre si. Existem várias abordagens para facilitar a comunicação, todas com diferentes vantagens e desvantagens.

Usando a descoberta de serviço

Uma abordagem para a comunicação de serviço para serviço é a comunicação direta usando a descoberta de serviços. Nessa abordagem, você pode usar oAWS Cloud Mapintegração de descoberta de serviços com o Amazon ECS. Usando a descoberta de serviços, o Amazon ECS sincroniza a lista de tarefas iniciadas comAWS Cloud Map, que mantém um nome de host DNS que resolve os endereços IP internos de uma ou mais tarefas desse serviço específico. Outros serviços na Amazon VPC podem usar esse nome de host DNS para enviar tráfego diretamente para outro contêiner usando seu endereço IP interno. Para obter mais informações, consulteDescoberta de serviçonoAmazon Elastic Container Service Guia do desenvolvedor.



No diagrama anterior, existem três serviços.`serviceA`tem um contêiner e se comunica com`serviceB`, que tem dois contêineres.`serviceB`também deve se comunicar com`serviceC`, que tem um contêiner. Cada contêiner em todos esses três serviços pode usar os nomes DNS internos doAWS Cloud Mappara localizar os endereços IP internos de um contêiner do serviço downstream com o qual ele precisa se comunicar.

Essa abordagem para comunicação serviço-a-serviço oferece baixa latência. À primeira vista, também é simples, pois não há componentes extras entre os contêineres. O tráfego viaja diretamente de um contêiner para o outro contêiner.

Esta abordagem é adequada ao usar`oawsvpc`, onde cada tarefa tem seu próprio endereço IP exclusivo. A maioria dos softwares só suporta o uso de DNSA, que resolvem diretamente para endereços IP. Ao usar`oawsvpc`, o endereço IP para cada tarefa é umAregistro. No entanto, se você estiver usando`bridge`, vários contêineres podem estar compartilhando o mesmo endereço IP. Além disso, mapeamentos de porta dinâmicos fazem com que os contentores sejam atribuídos aleatoriamente números de porta nesse único endereço IP. Neste ponto, umAnão é mais suficiente para a descoberta de serviços. Você também deve usar um`SRV`registro. Esse tipo de registro pode

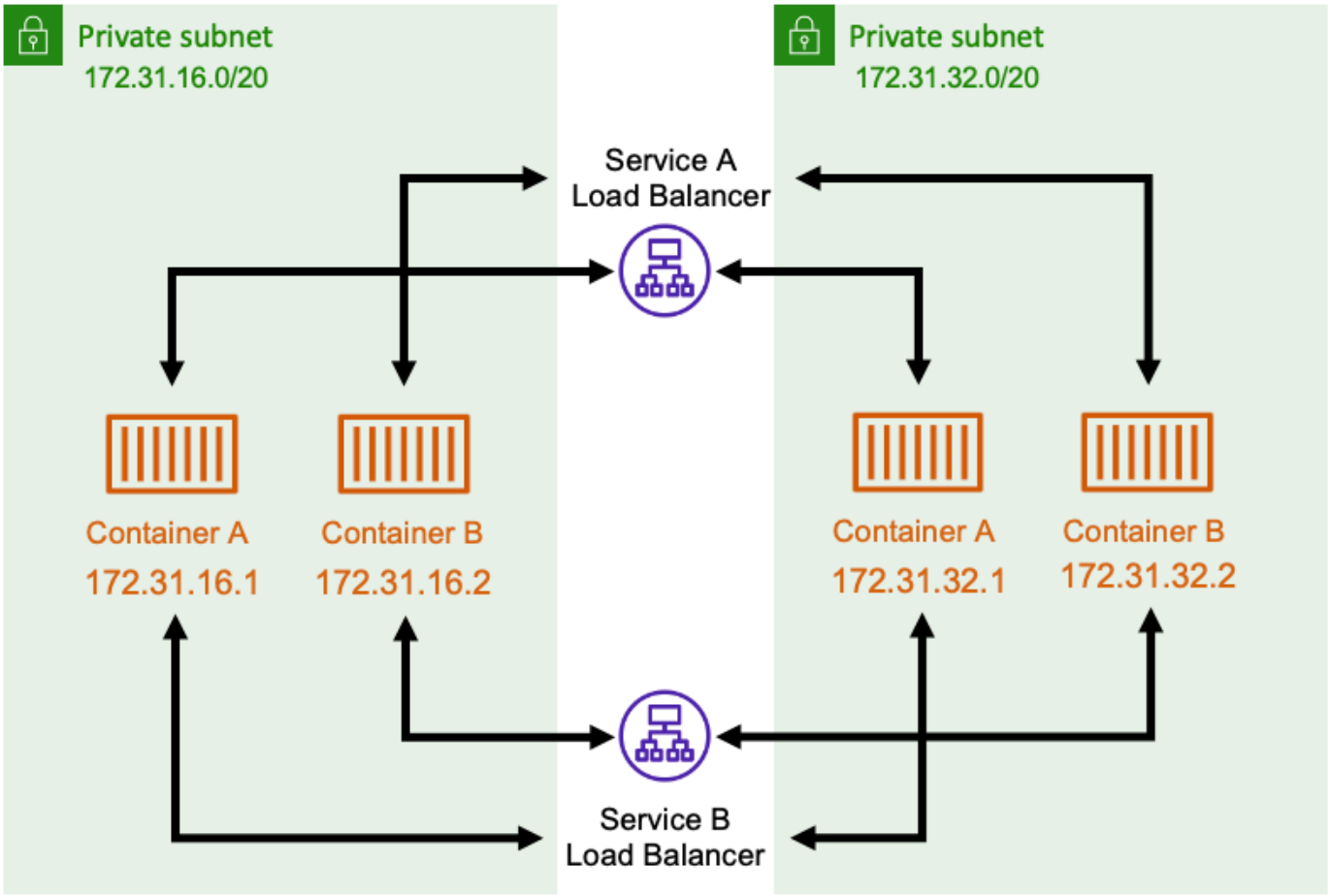
acompanhar os endereços IP e os números de porta, mas requer que você configure os aplicativos adequadamente. Alguns aplicativos pré-criados que você usa podem não suportar`SRV`registros.

Outra vantagem do`awsvpc`é que você tem um grupo de segurança exclusivo para cada serviço. Você pode configurar esse grupo de segurança para permitir conexões de entrada somente dos serviços upstream específicos que precisam falar com esse serviço.

A principal desvantagem da comunicação direta de serviço a serviço usando a descoberta de serviço é que você deve implementar lógica extra para ter novas tentativas e lidar com falhas de conexão. Os registros DNS têm um período de tempo de vida útil (TTL) que controla quanto tempo eles são armazenados em cache. Leva algum tempo para que o registro DNS seja atualizado e para o cache expirar para que seus aplicativos possam pegar a versão mais recente do registro DNS. Portanto, seu aplicativo pode acabar resolvendo o registro DNS para apontar para outro contêiner que não está mais lá. Seu aplicativo precisa lidar com novas tentativas e ter lógica para ignorar backends ruins.

Usando um load balancer interno

Outra abordagem para a comunicação serviço-a-serviço é usar um balanceador de carga interno. Um balanceador de carga interno existe inteiramente dentro de sua VPC e só é acessível a serviços dentro de sua VPC.



O balanceador de carga mantém a alta disponibilidade implantando recursos redundantes em cada sub-rede. Quando um contêiner de`serviceA`precisa se comunicar com um contêiner de`serviceB`, ele abre uma conexão com o load balancer. Em seguida, o balanceador de carga abre uma conexão com um contêiner de`service B`. O load balancer serve como local centralizado para gerenciar todas as conexões entre cada serviço.

Se um contêiner de`serviceB`pára, então o balanceador de carga pode remover esse contêiner do pool. O load balancer também faz verificações de saúde em relação a cada destino downstream em seu pool e pode remover automaticamente alvos inválidos do pool até que eles se tornem íntegros novamente. Os aplicativos não precisam mais estar cientes de quantos contêineres a jusante existem. Eles só abrem as conexões com o load balancer.

Essa abordagem é vantajosa para todos os modos de rede. O balanceador de carga pode acompanhar os endereços IP de tarefas ao usar o comando`awsvpc`, bem como combinações mais avançadas de endereço IP e porta ao usar o`bridge`Modo de rede. Ele distribui o tráfego

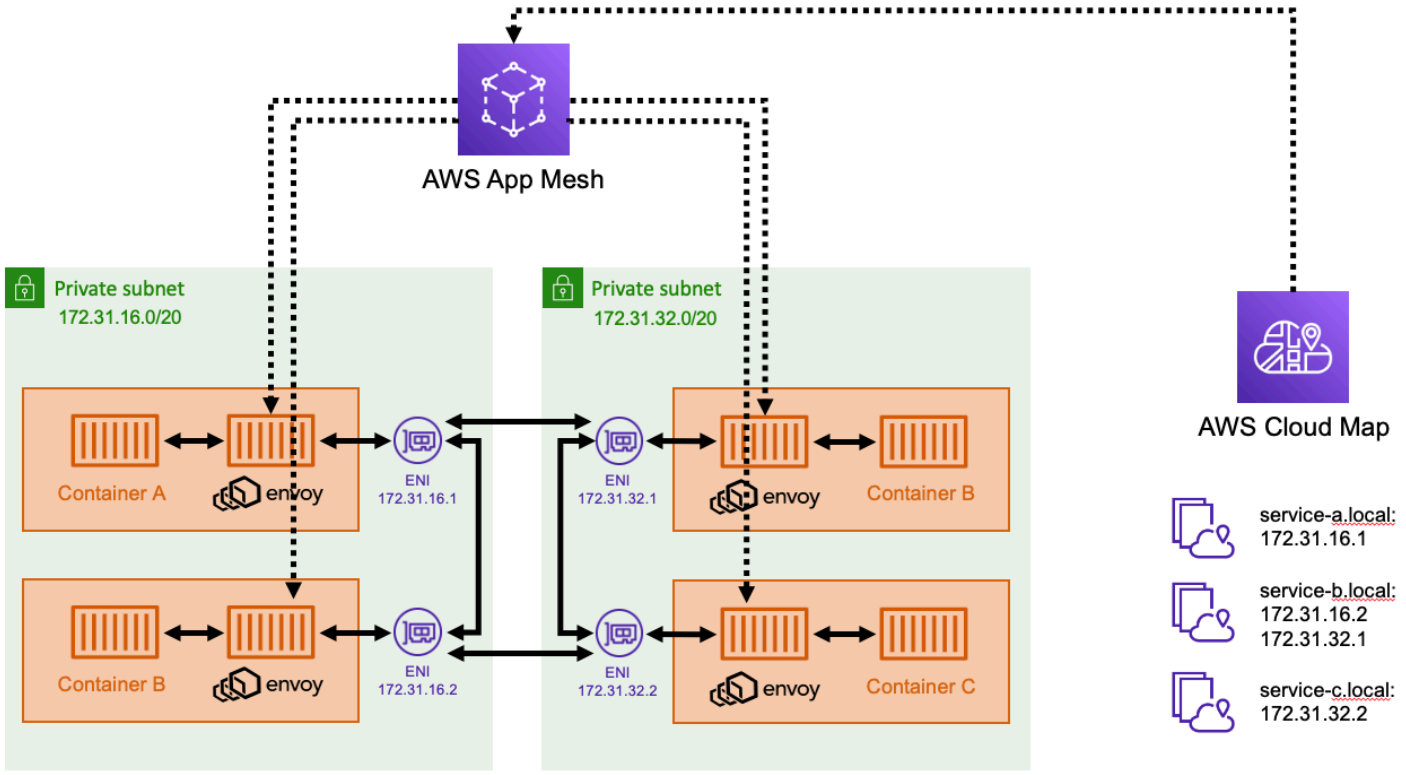
uniformemente por todas as combinações de endereços IP e portas, mesmo que vários contêineres estejam hospedados na mesma instância do Amazon EC2, apenas em portas diferentes.

A única desvantagem desta abordagem é o custo. Para ser altamente disponível, o balanceador de carga precisa ter recursos em cada zona de disponibilidade. Isso adiciona custo extra devido à sobrecarga de pagar pelo balanceador de carga e pela quantidade de tráfego que passa pelo balanceador de carga.

No entanto, você pode reduzir os custos indiretos fazendo com que vários serviços compartilhem um balanceador de carga. Isso é particularmente adequado para serviços REST que usam um Application Load Balancer. Você pode criar regras de roteamento baseadas em caminho que roteiam o tráfego para serviços diferentes. Por exemplo,`/api/user/*`pode rotear para um contêiner que faz parte do`user`serviçoServiço do`/api/order/*`pode rotear para o`order`serviçoServiço do Com essa abordagem, você paga apenas por um Application Load Balancer e tem um URL consistente para sua API. No entanto, você pode dividir o tráfego para vários microsserviços no back-end.

Usando uma malha de serviço

AWS App Meshé uma malha de serviço que pode ajudá-lo a gerenciar um grande número de serviços e ter melhor controle de como o tráfego é roteado entre os serviços. O App Mesh funciona como um intermediário entre a descoberta de serviços básicos e o balanceamento de carga. Com o App Mesh, os aplicativos não interagem diretamente entre si, mas também não usam um balanceador de carga centralizado. Em vez disso, cada cópia da sua tarefa é acompanhada por um sidecar proxy Enviado. Para obter mais informações, consulte O que é o AWS App Mesh no Guia do usuário do AWS App Mesh.



No diagrama anterior, cada tarefa tem um sidecar de proxy do Enviado. Este sidecar é responsável por proxy todo o tráfego de entrada e saída para a tarefa. O plano de controle App Mesh usaAWS Cloud Mappara obter a lista de serviços disponíveis e os endereços IP de tarefas específicas. Em seguida, o App Mesh fornece a configuração para o sidecar proxy Envoy. Essa configuração inclui a lista de contêineres disponíveis aos quais podem ser conectados. O sidecar proxy Enviado também realiza verificações de saúde contra cada alvo para garantir que eles estejam disponíveis.

Essa abordagem fornece os recursos da descoberta de serviços, com a facilidade do balanceador de carga gerenciado. Os aplicativos não implementam tanta lógica de balanceamento de carga em seu código porque o sidecar proxy Envoy manipula esse balanceamento de carga. O proxy Envoy pode ser configurado para detectar falhas e repetir solicitações com falha. Além disso, ele também pode ser configurado para usar MTLs para criptografar o tráfego em trânsito e garantir que seus aplicativos estejam se comunicando com um destino verificado.

Existem poucas diferenças entre um proxy do Envoy e um load balancer. Em suma, com o proxy Envoy, você é responsável por implantar e gerenciar seu próprio sidecar proxy Envoy. O sidecar de proxy Envoy usa parte da CPU e da memória que você aloca para a tarefa do Amazon ECS. Isso adiciona alguma sobrecarga ao consumo de recursos da tarefa e carga de trabalho operacional adicional para manter e atualizar o proxy quando necessário.

O App Mesh e um proxy Envoy permitem uma latência extremamente baixa entre as tarefas. Isso ocorre porque o proxy Envoy é executado colocado em cada tarefa. Há apenas uma instância para o salto de rede da instância, entre um proxy Envoy e outro proxy Envoy. Isso significa que há também menos sobrecarga de rede em comparação com o uso de balanceadores de carga. Ao usar balanceadores de carga, há dois saltos de rede. O primeiro é da tarefa upstream para o load balancer e o segundo é do load balancer para a tarefa downstream.

Serviços de rede emAWScontas e VPCs

Se você fizer parte de uma organização com várias equipes e divisões, provavelmente implantará serviços de forma independente em VPCs separadas dentro de umAWSou em VPCs que estão associados a váriosAWScontas. Não importa qual a maneira de implantar seus serviços, recomendamos que você complete seus componentes de rede para ajudar a rotear o tráfego entre VPCs. Para isso, váriosAWSpodem ser usados para complementar seus componentes de rede existentes.

- AWS Transit Gateway — Você deve considerar esse serviço de rede primeiro. Esse serviço serve como um hub central para rotear suas conexões entre Amazon VPCs,AWSe redes locais. Para obter mais informações, consulteO que é um gateway de trânsito?noGuia de gateways de trânsito da Amazon VPC.
- Suporte à Amazon VPC e VPN — Você pode usar esse serviço para criar conexões VPN site a site para conectar redes locais à sua VPC. Para obter mais informações, consulte O que é o AWS Site-to-Site VPN? no Guia do usuário do AWS Site-to-Site VPN.
- Amazon VPC — Você pode usar o peering da Amazon VPC para ajudá-lo a conectar várias VPCs, na mesma conta ou entre contas. Para obter mais informações, consulte O que é emparelhamento de VPC? no Amazon VPC Peering Guide.
- VPCs compartilhadas — você pode usar uma VPC e sub-redes VPC em váriosAWScontas. Para obter mais informações, consulteTrabalhar com VPCs compartilhadasnoGuia do usuário da Amazon VPC.

Otimização e solução de problemas

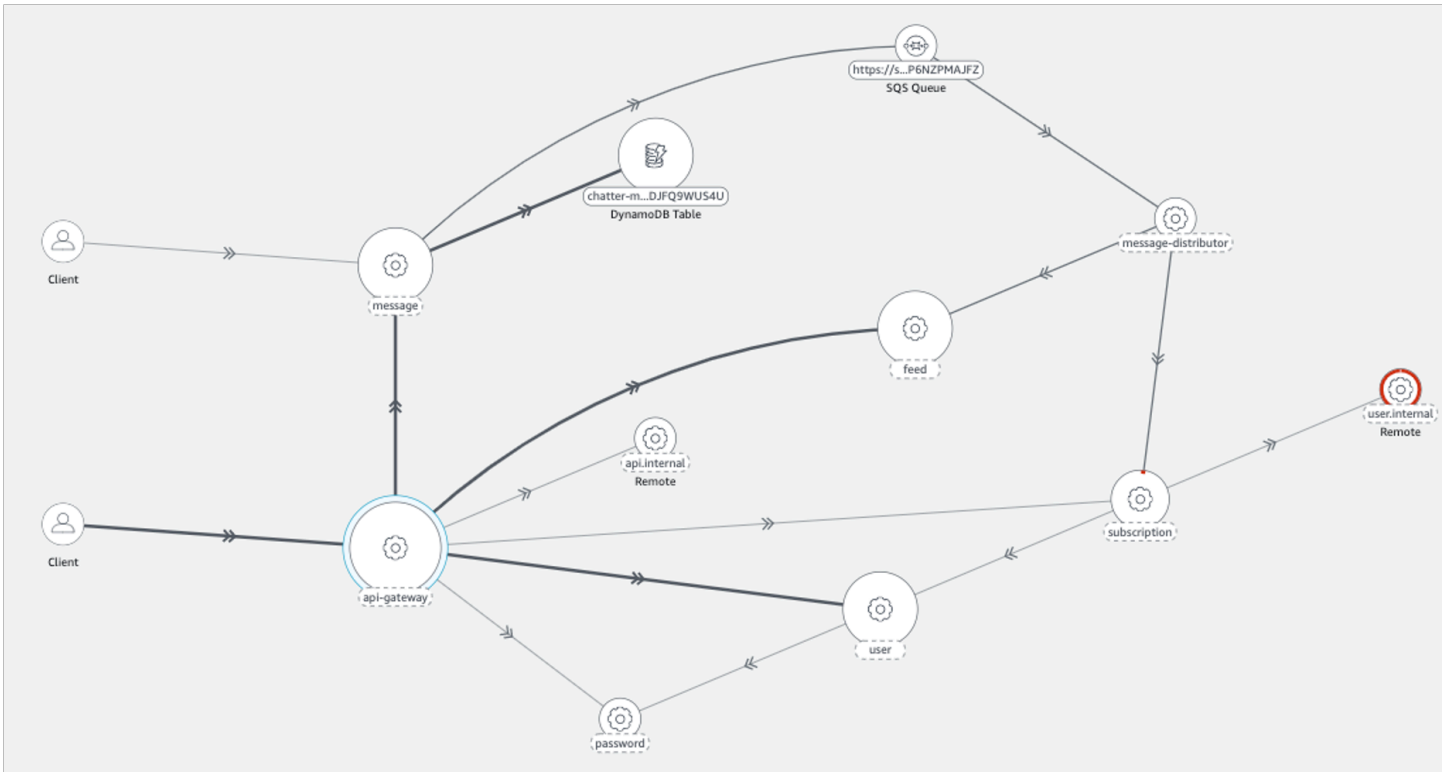
Os seguintes serviços e recursos podem ajudá-lo a obter insights sobre suas configurações de rede e serviço. Você pode usar essas informações para solucionar problemas de rede e otimizar melhor os serviços.

Insights do CloudWatch

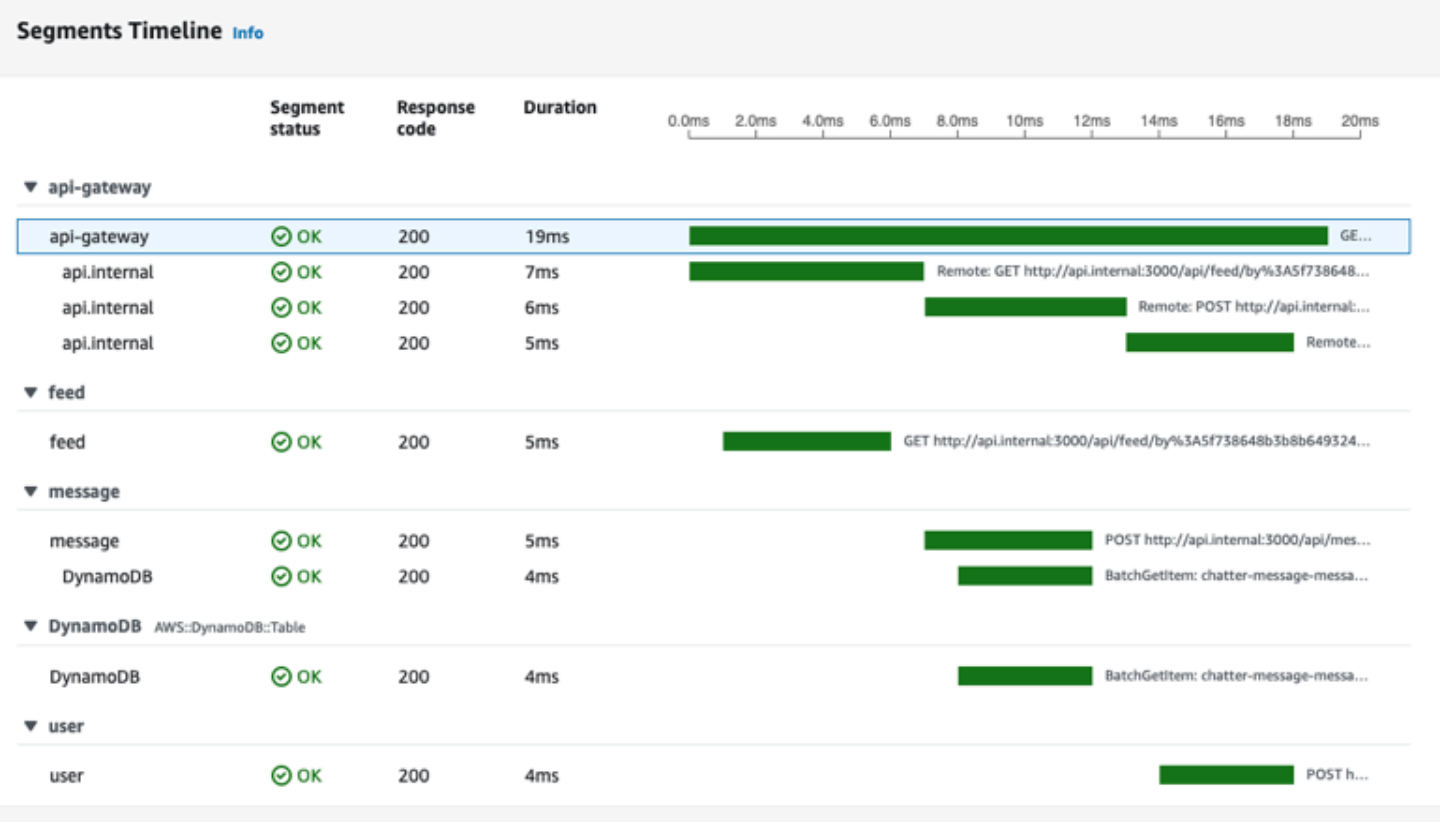
O CloudWatch Container Insights coleta, agrega e resume métricas e logs de seus aplicativos e microsserviços em contêineres. As métricas incluem a utilização de recursos, como CPU, memória, disco e rede. Eles estão disponíveis nos painéis automáticos do CloudWatch. Para obter mais informações, consulteConfigurando o Container Insights no Amazon ECSnoGuia do usuário do Amazon CloudWatch.

AWS X-Ray

AWS X-Rayé um serviço de rastreamento que você pode usar para coletar informações sobre as solicitações de rede feitas pelo aplicativo. Você pode usar o SDK para instrumentar seu aplicativo e capturar temporizações e códigos de resposta de tráfego entre seus serviços e entre seus serviços eAWSEndpoints de serviço do. Para obter mais informações, consulteO que é oAWS X-RaynoAWS X-RayGuia do desenvolvedor.



Você também pode explorarAWS X-Raygráficos de como seus serviços se conectam uns com os outros. Ou, use-os para explorar estatísticas agregadas sobre o desempenho de cada link de serviço para serviço. Por último, você pode aprofundar qualquer transação específica para ver como segmentos que representam chamadas de rede estão associados a essa transação específica.



Você pode usar esses recursos para identificar se há um gargalo de rede ou se um serviço específico dentro de sua rede não está funcionando conforme esperado.

VPC Flow Logs

Você pode usar os logs de fluxo da Amazon VPC para analisar o desempenho da rede e depurar problemas de conectividade. Com os logs de fluxo da VPC habilitados, você pode capturar um log de todas as conexões em sua VPC. Isso inclui conexões com interfaces de rede associadas ao Elastic Load Balancing, Amazon RDS, gateways NAT e outras chavesAWSserviços que você pode estar usando. Para obter mais informações, consulte Logs de fluxo da VPC no Guia do usuário da Amazon VPC.

Dicas de ajuste de rede

Existem algumas configurações que você pode ajustar para melhorar sua rede.

nofile ulimit

Se você espera que seu aplicativo tenha alto tráfego e manipule muitas conexões simultâneas, você deve levar em conta a cota do sistema para o número de arquivos permitidos. Quando há muitos

soquetes de rede abertos, cada um deve ser representado por um descritor de arquivo. Se a cota do descritor de arquivo for muito baixa, ela limitará os soquetes de rede. Isso resulta em falhas de conexões ou erros. Você pode atualizar a cota específica do contêiner para o número de arquivos na definição de tarefa do Amazon ECS. Se você estiver executando o Amazon EC2 (em vez deAWS Fargate), talvez você também precise ajustar essas cotas em sua instância subjacente do Amazon EC2.

sysctl net

Outra categoria de configuração ajustável é o`sysctl`Configurações de rede. Você deve consultar as configurações específicas para sua distribuição Linux de escolha. Muitas dessas configurações ajustam o tamanho dos buffers de leitura e gravação. Isso pode ajudar em algumas situações ao executar instâncias do Amazon EC2 em grande escala que têm muitos contêineres neles.

Práticas recomendadas - Auto scaling e gerenciamento de capacidade

O Amazon ECS é usado para executar cargas de trabalho de aplicativos em contêineres de todos os tamanhos. Isso inclui os extremos de ambientes de teste mínimos e grandes ambientes de produção operando em escala global.

Com o Amazon ECS, como todosAWSOs serviços da, você paga somente por aquilo que usa. Quando arquitetado adequadamente, você pode economizar custos fazendo com que seu aplicativo consuma apenas os recursos necessários no momento em que precisar deles. Este guia de práticas recomendadas mostra como executar suas cargas de trabalho do Amazon ECS de uma maneira que atenda aos seus objetivos de nível de serviço enquanto ainda opera de forma econômica.

Tópicos

Determinar tamanho da tarefa

Uma das escolhas mais importantes a serem feitas ao implantar contêineres no Amazon ECS é o tamanho do contêiner e das tarefas. Os tamanhos de contêiner e de tarefas são essenciais para o dimensionamento e o planejamento da capacidade. No Amazon ECS, há duas métricas de recursos usadas para capacidade: CPU e memória. A CPU é medida em unidades de 1/1024 de uma vCPU completa (onde 1024 unidades é igual a 1 vCPU inteira). A memória é medida em megabytes. Na definição da tarefa, você pode declarar reservas e limites de recursos.

Ao declarar uma reserva, você declara a quantidade mínima de recursos que uma tarefa requer. Sua tarefa recebe pelo menos a quantidade de recursos solicitados. Seu aplicativo pode ser capaz

de usar mais CPU ou memória do que a reserva que você declara. No entanto, isso está sujeito a quaisquer limites que você também declarou. Usar mais do que o valor da reserva é conhecido como estourar. No Amazon ECS, as reservas são garantidas. Por exemplo, se você usar instâncias do Amazon EC2 para fornecer capacidade, o Amazon ECS não colocará uma tarefa em uma instância em que a reserva não possa ser atendida.

Um limite é a quantidade máxima de unidades de CPU ou memória que seu contêiner ou tarefa pode usar. Qualquer tentativa de usar mais CPU do que isso imit resulta em limitação. Qualquer tentativa de usar mais memória resulta em seu contêiner ser interrompido.

Escolher esses valores pode ser um desafio. Isso ocorre porque os valores que são os mais adequados para o seu aplicativo dependem muito dos requisitos de recursos do seu aplicativo. O teste de carga de seu aplicativo é a chave para o planejamento bem-sucedido de requisitos de recursos e uma melhor compreensão dos requisitos de seu aplicativo.

Aplicativos stateless

Para aplicativos sem estado que são dimensionados horizontalmente, como um aplicativo atrás de um balanceador de carga, recomendamos que você primeiro determine a quantidade de memória que seu aplicativo consome quando ele atende solicitações. Para fazer isso, você pode usar ferramentas tradicionais como`psoutop`ou soluções de monitoramento, como o CloudWatch Container Insights.

Ao determinar uma reserva de CPU, considere como deseja dimensionar seu aplicativo para atender aos requisitos de negócios. Você pode usar reservas de CPU menores, como 256 unidades de CPU (ou 1/4 vCPU), para dimensionar horizontalmente de forma refinada que minimiza o custo. Mas, eles podem não escalar rápido o suficiente para atender picos significativos na demanda. Você pode usar reservas de CPU maiores para aumentar e reduzir a escala mais rapidamente e, portanto, corresponder picos de demanda mais rapidamente. No entanto, reservas de CPU maiores são mais dispendiosas.

Outros aplicativos

Para aplicativos que não são dimensionados horizontalmente, como trabalhadores singleton ou servidores de banco de dados, a capacidade e o custo disponíveis representam suas considerações mais importantes. Você deve escolher a quantidade de memória e CPU com base em quais testes de carga indicam que você precisa servir o tráfego para atender ao seu objetivo de nível de serviço. O Amazon ECS garante que o aplicativo seja colocado em um host com capacidade adequada.

Configurando auto scaling de serviço

Um serviço do Amazon ECS é uma coleção gerenciada de tarefas. Cada serviço tem uma definição de tarefa associada, uma contagem de tarefas desejada e uma estratégia de posicionamento opcional. O dimensionamento automático de serviço do Amazon ECS é implementado por meio do serviço Application Auto Scaling. O Application Auto Scaling usa métricas do CloudWatch como a origem para dimensionar métricas. Ele também usa alarmes do CloudWatch para definir limites sobre quando dimensionar seu serviço para dentro ou para fora. Você fornece os limites para escalabilidade, definindo um alvo de métrica, referido comoDimensionamento de rastreamento de destino, ou especificando limiares, referidos comoescalabilidade em etapas. Depois que o Application Auto Scaling é configurado, ele calcula continuamente a contagem de tarefas desejada apropriada para o serviço. Ele também notifica o Amazon ECS quando a contagem de tarefas desejada deve ser alterada, seja dimensionando-a horizontalmente ou dimensionando-a.

Para usar o dimensionamento automático de serviço de forma eficaz, você deve escolher uma métrica de dimensionamento apropriada. Discutimos como escolher uma métrica nas seções a seguir.

Caracterizando seu aplicativo

Dimensionar corretamente um aplicativo requer conhecer as condições em que o aplicativo deve ser dimensionado e quando deve ser dimensionado horizontalmente. Em essência, um aplicativo deve ser dimensionado se a demanda for prevista para ultrapassar a capacidade. Por outro lado, um aplicativo pode ser dimensionado para economizar custos quando os recursos excedem a demanda.

Identificando uma métrica de utilização

Para dimensionar de forma eficaz, é fundamental identificar uma métrica que indique utilização ou saturação. Esta métrica deve apresentar as seguintes propriedades para ser útil para dimensionamento.

- A métrica deve ser correlacionada com a demanda. Quando os recursos são mantidos estáveis, mas a demanda muda, o valor da métrica também deve ser alterado. A métrica deve aumentar ou diminuir quando a demanda aumenta ou diminui.
- O valor métrico deve ser dimensionado proporcionalmente à capacidade. Quando a demanda é constante, a adição de mais recursos deve resultar em uma alteração proporcional no valor da métrica. Assim, duplicar o número de tarefas deve fazer com que a métrica diminua em 50%.

A melhor maneira de identificar uma métrica de utilização é por meio de testes de carga em um ambiente de pré-produção, como um ambiente de preparação. Soluções de teste de carga comerciais e de código aberto estão amplamente disponíveis. Essas soluções normalmente podem gerar carga sintética ou simular tráfego real do usuário.

Para iniciar o processo de teste de carga, você deve começar criando painéis para as métricas de utilização do seu aplicativo. Essas métricas incluem utilização da CPU, utilização da memória, operações de E/S, profundidade da fila de E/S e taxa de transferência da rede. Você pode coletar essas métricas com um serviço como o CloudWatch Container Insights. Ou faça isso usando o Amazon Managed Service for Prometheus juntamente com o Amazon Managed Service for Grafana. Durante esse processo, certifique-se de coletar e plotar métricas para os tempos de resposta do aplicativo ou as taxas de conclusão de trabalho.

Ao testar a carga, comece com uma pequena solicitação ou taxa de inserção de trabalho. Mantenha essa taxa estável por vários minutos para permitir que sua aplicação aqueça. Em seguida, aumente lentamente a taxa e mantenha-a firme por alguns minutos. Repita esse ciclo, aumentando a taxa de cada vez até que os tempos de resposta ou conclusão do aplicativo sejam muito lentos para atender aos SLOs (Service Level Objectives, objetivos de nível de serviço).

Durante o teste de carga, examine cada uma das métricas de utilização. As métricas que aumentam junto com a carga são os principais candidatos para servir como suas melhores métricas de utilização.

Em seguida, identifique o recurso que atinge a saturação. Ao mesmo tempo, examine também as métricas de utilização para ver qual deles achata-se primeiro em um nível alto. Ou, examine qual deles atinge o pico e, em seguida, trava seu aplicativo primeiro. Por exemplo, se a utilização da CPU aumenta de 0% para 70 -80% à medida que você adiciona carga, então permanece nesse lebel depois que ainda mais carga é adicionada, então é seguro dizer que a CPU está saturada. Dependendo da arquitetura da CPU, ela pode nunca atingir 100%. Por exemplo, suponha que a utilização da memória aumenta à medida que você adiciona carga e, em seguida, seu aplicativo falha repentinamente quando atinge a tarefa ou o limite de memória da instância do Amazon EC2. Nessa situação, é provável que a memória tenha sido totalmente consumida. Vários recursos podem ser consumidos pelo seu aplicativo. Portanto, escolha a métrica que representa o recurso que esgota primeiro.

Por último, tente testar novamente a carga depois de duplicar o número de tarefas ou instâncias do Amazon EC2. Suponha que a métrica-chave aumenta, ou diminui, na metade da taxa como antes. Se esse for o caso, a métrica é proporcional à capacidade. Esta é uma boa métrica de utilização para auto scaling.

Agora considere este cenário hipotético. Suponha que você carregue teste um aplicativo e descubra que a utilização da CPU eventualmente atinge 80% em 100 solicitações por segundo. Quando mais carga é adicionada, ela não faz mais a utilização da CPU aumentar. No entanto, ele faz com que seu aplicativo responda mais lentamente. Em seguida, você executar o teste de carga novamente, duplicando o número de tarefas, mas mantendo a taxa em seu valor de pico anterior. Se você achar que a utilização média da CPU cai para cerca de 40%, a utilização média da CPU é um bom candidato para uma métrica de dimensionamento. Por outro lado, se a utilização da CPU permanecer em 80% depois de aumentar o número de tarefas, a utilização média da CPU não é uma boa métrica de dimensionamento. Nesse caso, mais pesquisas são necessárias para encontrar uma métrica adequada.

Modelos de aplicativos comuns e propriedades de dimensionamento

Software de todos os tipos são executados emAWS. Muitas cargas de trabalho são criadas internamente, enquanto outras são baseadas em softwares populares de código aberto. Independentemente de onde eles se originam, observamos alguns padrões comuns de design para serviços. Como dimensionar eficazmente depende em grande parte do padrão.

O eficiente servidor vinculado à CPU

O eficiente servidor ligado à CPU não utiliza quase nenhum recurso além da CPU e da taxa de transferência de rede. Cada pedido pode ser tratado apenas pelo aplicativo. As solicitações não dependem de outros serviços, como bancos de dados. O aplicativo pode lidar com centenas de milhares de solicitações simultâneas e pode utilizar várias CPUs com eficiência para fazer isso. Cada solicitação é atendida por um thread dedicado com pouca sobrecarga de memória ou há um loop de eventos assíncrono que é executado em cada CPU que atende solicitações. Cada réplica do aplicativo é igualmente capaz de lidar com uma solicitação. O único recurso que pode ser esgotado antes da CPU é a largura de banda da rede. Nos serviços limitados da CPU, a utilização da memória, mesmo com taxa de transferência máxima, é uma fração dos recursos disponíveis.

Este tipo de aplicação é adequado para dimensionamento automático baseado em CPU. O aplicativo goza de máxima flexibilidade em termos de dimensionamento. Ele pode ser dimensionado verticalmente, fornecendo instâncias maiores do Amazon EC2 ou vCPUs Fargate para ele. Além disso, ele também pode ser dimensionado horizontalmente adicionando mais réplicas. Adicionar mais réplicas, ou duplicar o tamanho da instância, reduz a utilização média da CPU em relação à capacidade pela metade.

Se você estiver usando a capacidade do Amazon EC2 para este aplicativo, considere colocá-la em instâncias otimizadas para computação, como `oc5ouc6g`família.

O eficiente servidor vinculado à memória

O eficiente servidor vinculado à memória aloca uma quantidade significativa de memória por solicitação. Na simultaneidade máxima, mas não necessariamente taxa de transferência, a memória é esgotada antes que os recursos da CPU sejam esgotados. A memória associada a uma solicitação é liberada quando a solicitação termina. Solicitações adicionais podem ser aceitas desde que haja memória disponível.

Este tipo de aplicação é adequado para auto scaling baseado em memória. O aplicativo goza de máxima flexibilidade em termos de dimensionamento. Ele pode ser dimensionado verticalmente, fornecendo recursos de memória maiores do Amazon EC2 ou Fargate para ele. Além disso, ele também pode ser dimensionado horizontalmente adicionando mais réplicas. Adicionar mais réplicas, ou duplicar o tamanho da instância, pode reduzir pela metade a utilização média da memória em relação à capacidade.

Se você estiver usando a capacidade do Amazon EC2 para este aplicativo, considere colocá-la em instâncias otimizadas para memória, como `or5our6g`família.

Alguns aplicativos vinculados à memória não liberam a memória associada a uma solicitação quando ela termina, de modo que uma redução na simultaneidade não resulte em uma redução na memória usada. Para isso, não recomendamos usar escalabilidade baseada em memória.

O servidor baseado no trabalhador

O servidor baseado em trabalho processa uma solicitação para cada thread de trabalho individual um após o outro. Os threads de trabalho podem ser threads leves, como threads POSIX. Eles também podem ser segmentos de peso mais pesado, como processos UNIX. Não importa qual thread eles são, há sempre uma simultaneidade máxima que o aplicativo pode suportar. Normalmente, o limite de simultaneidade é definido proporcionalmente aos recursos de memória disponíveis. Se o limite de simultaneidade for atingido, solicitações adicionais serão colocadas em uma fila de backlog. Se a fila de backlog ultrapassar, solicitações de entrada adicionais serão imediatamente rejeitadas. Aplicativos comuns que se encaixam nesse padrão incluem o servidor web Apache e o Gunicorn.

A simultaneidade de solicitações geralmente é a melhor métrica para dimensionar esse aplicativo. Como há um limite de simultaneidade para cada réplica, é importante aumentar a escala antes que o limite médio seja atingido.

A melhor maneira de obter métricas de simultaneidade de solicitação é fazer com que seu aplicativo os reporte ao CloudWatch. Cada réplica do seu aplicativo pode publicar o número de solicitações

simultâneas como uma métrica personalizada em alta freqüência. Recomendamos que a frequência seja definida para ser pelo menos uma vez a cada minuto. Depois que vários relatórios são coletados, você pode usar a simultaneidade média como uma métrica de dimensionamento. Essa métrica é calculada tomando a simultaneidade total e dividindo-a pelo número de réplicas. Por exemplo, se a simultaneidade total for 1000 e o número de réplicas for 10, a simultaneidade média será 100.

Se seu aplicativo estiver por trás de um Application Load Balancer, você também pode usar o`ActiveConnectionCount`para o balanceador de carga como um fator na métrica de dimensionamento. O`ActiveConnectionCount`deve ser dividida pelo número de réplicas para obter um valor médio. O valor médio deve ser usado para escalabilidade, em oposição ao valor bruto da contagem.

Para que esse projeto funcione melhor, o desvio padrão da latência de resposta deve ser pequeno com baixas taxas de solicitação. Recomendamos que, durante períodos de baixa demanda, a maioria das solicitações seja respondida em um curto espaço de tempo, e não há muitas solicitações que demoram significativamente mais do que o tempo médio para responder. O tempo médio de resposta deve estar próximo do tempo de resposta do percentil 95. Caso contrário, os estouros de fila podem ocorrer como resultado. Isso leva a erros. Recomendamos que você forneça réplicas adicionais quando necessário para reduzir o risco de estouro.

O servidor em espera

O servidor em espera faz algum processamento para cada solicitação, mas é altamente dependente de um ou mais serviços downstream para funcionar. Os aplicativos de contêiner geralmente fazem uso pesado de serviços downstream, como bancos de dados e outros serviços de API. Isso ocorre porque esses aplicativos tendem a usar poucos recursos de CPU e sua simultaneidade máxima em termos de memória disponível.

O serviço de espera é adequado no padrão de servidor vinculado à memória ou no padrão de servidor baseado no trabalhador, dependendo de como o aplicativo foi projetado. Se a simultaneidade do aplicativo for limitada apenas pela memória, a utilização média da memória deve ser usada como uma métrica de dimensionamento. Se a simultaneidade do aplicativo for baseada em um limite de trabalhador, a simultaneidade média deve ser usada como uma métrica de dimensionamento.

O servidor baseado em Java

Se o seu servidor baseado em Java for vinculado à CPU e for dimensionado proporcionalmente aos recursos da CPU, ele poderá ser adequado para o padrão eficiente de servidor vinculado à

CPU. Se for esse o caso, a utilização média da CPU pode ser apropriada como uma métrica de dimensionamento. No entanto, muitos aplicativos Java não são vinculados à CPU, tornando-os desafiadores de escalabilidade.

Tendo em vista o melhor desempenho, recomendamos alocar o máximo de memória possível ao heap Java Virtual Machine (JVM — Máquina virtual do Java). Versões recentes da JVM, incluindo o Java 8 atualização 191 ou posterior, definem automaticamente o tamanho do heap o maior possível para caber dentro do contêiner. Isso significa que, em Java, a utilização da memória raramente é proporcional à utilização do aplicativo. À medida que a taxa de solicitação e a simultaneidade aumentam, a utilização da memória permanece constante. Por isso, não recomendamos escalar servidores baseados em Java com base na utilização da memória. Em vez disso, geralmente recomendamos escalar a utilização da CPU.

Em alguns casos, os servidores baseados em Java encontram exaustão de heap antes de esgotar a CPU. Se seu aplicativo é propenso a esgotamento de heap em alta simultaneidade, as conexões médias são a melhor métrica de dimensionamento. Se o seu aplicativo estiver propenso a esgotamento de heap com alta taxa de transferência, a taxa média de solicitação é a melhor métrica de dimensionamento.

Servidores que usam outros tempos de execução coletados por lixo

Muitos aplicativos de servidor são baseados em tempos de execução que executam coleta de lixo, como .NET e Ruby. Esses aplicativos de servidor podem se encaixar em um dos padrões descritos anteriormente. No entanto, como acontece com o Java, não recomendamos escalar esses aplicativos com base na memória, porque sua utilização média de memória observada geralmente não está correlacionada com throughput ou simultaneidade.

Para esses aplicativos, recomendamos que você dimensione a utilização da CPU se o aplicativo estiver vinculado à CPU. Caso contrário, recomendamos que você dimensione a taxa de transferência média ou a simultaneidade média, com base nos resultados dos testes de carga.

Processadores de Job

Muitas cargas de trabalho envolvem processamento de trabalho assíncrono. Eles incluem aplicativos que não recebem solicitações em tempo real, mas, em vez disso, se inscrevem em uma fila de trabalho para receber trabalhos. Para esses tipos de aplicativos, a métrica de dimensionamento adequada é quase sempre a profundidade da fila. O crescimento da fila é uma indicação de que o trabalho pendente supera a capacidade de processamento, enquanto uma fila vazia indica que há mais capacidade do que trabalho a ser feito.

AWSserviços de mensagens, como Amazon SQS e Amazon Kinesis Data Streams, fornecem métricas do CloudWatch que podem ser usadas para escalabilidade. Para o Amazon SQS,`ApproximateNumberOfMessagesVisible`é a melhor métrica. Para Kinesis Data Streams, considere usar o`MillisBehindLatest`, publicada pela Kinesis Client Library (KCL). Essa métrica deve ser calculada em todos os consumidores antes de usá-la para dimensionamento.

Capacidade e disponibilidade

A disponibilidade dos aplicativos é crucial para oferecer uma experiência livre de erros e para minimizar a latência dos aplicativos. A disponibilidade depende de ter recursos acessíveis e capacidade suficiente para atender à demanda.AWSfornece vários mecanismos para gerenciar a disponibilidade. Para aplicativos hospedados no Amazon ECS, esses incluem autoscaling e zonas de disponibilidade (AZs). O dimensionamento automático gerencia o número de tarefas ou instâncias com base nas métricas definidas por você, enquanto as Zonas de disponibilidade permitem que você hospede seu aplicativo em locais isolados, mas geograficamente fechados.

Tal como acontece com tamanhos de tarefas, capacidade e disponibilidade apresentam certas compensações que você deve considerar. Idealmente, a capacidade seria perfeitamente alinhada com a demanda. Haveria sempre capacidade suficiente para atender solicitações e trabalhos de processo para atender aos SLOs (Service Level Objectives, objetivos de nível de serviço), incluindo baixa latência e taxa de erro. A capacidade nunca seria muito alta, levando a custos excessivos; nem nunca seria muito baixa, levando a altas taxas de latência e erro.

O dimensionamento automático é um processo latente. Primeiro, as métricas em tempo real devem ser entregues ao CloudWatch. Então, eles precisam ser agregados para análise, o que pode levar até vários minutos dependendo da granularidade da métrica. O CloudWatch compara as métricas com os limites de alarme para identificar uma falta ou excesso de recursos. Para evitar instabilidade, configure os alarmes para exigir que o limite definido seja cruzado por alguns minutos antes do alarme disparar. Também leva tempo para provisionar novas tarefas e encerrar tarefas que não são mais necessárias.

Devido a esses possíveis atrasos no sistema descrito, é importante que você mantenha algum espaço livre por excesso de provisionamento. Fazer isso pode ajudar a acomodar explosões de curto prazo na demanda. Isso também ajuda seu aplicativo a atender solicitações adicionais sem atingir a saturação. Como uma boa prática, você pode definir sua meta de dimensionamento entre 60 e 80% da utilização. Isso ajuda seu aplicativo a lidar melhor com picos de demanda extra, enquanto a capacidade adicional ainda está em processo de provisionamento.

Outra razão pela qual recomendamos que você faça o provisionamento excessivo é para que você possa responder rapidamente a falhas na Zona de disponibilidade.AWSrecomenda que as cargas de trabalho de produção sejam atendidas de várias zonas de disponibilidade. Isso ocorre porque, se ocorrer uma falha na Zona de disponibilidade, suas tarefas que estão sendo executadas nas Zonas de disponibilidade restantes ainda poderão atender à demanda. Se o aplicativo for executado em duas Zonas de disponibilidade, você precisará dobrar a contagem normal de tarefas. Isso é para que você possa fornecer capacidade imediata durante qualquer falha potencial. Se o aplicativo for executado em três Zonas de disponibilidade, recomendamos que você execute 1,5 vezes a contagem normal de tarefas. Ou seja, execute três tarefas para cada duas que são necessárias para o serviço comum.

Maximização da velocidade de escalabilidade

Autoscaling é um processo reativo que leva tempo para entrar em vigor. No entanto, existem algumas maneiras de ajudar a minimizar o tempo necessário para dimensionar horizontalmente.

Minimize o tamanho da imagem As imagens maiores levam mais tempo para serem baixadas de um repositório de imagens e descompactadas. Portanto, manter os tamanhos de imagem menores reduz o tempo necessário para que um contêiner seja iniciado. Para reduzir o tamanho da imagem, você pode seguir estas recomendações específicas:

- Se você pode construir um binário estático ou usar Golang, construa sua imagem`FROM`zero e incluir apenas a sua aplicação binária na imagem resultante.
- Use imagens básicas minimizadas de fornecedores de distribuição upstream, como Amazon Linux ou Ubuntu.
- Não inclua artefatos de compilação na sua imagem final. Usar compilações de vários estágios pode ajudar com isso.
- CompactoRUNsempre que possível. Cada`RUN`cria uma nova camada de imagem, levando a uma ida e volta adicional para baixar a camada. Um único`RUN`que tem vários comandos unidos por`&&`tem menos camadas do que uma com vários`RUN`Estágios.
- Se você quiser incluir dados, como dados de inferência de ML, na imagem final, inclua apenas os dados necessários para iniciar e começar a servir tráfego. Se você buscar dados sob demanda do Amazon S3 ou de outro armazenamento sem afetar o serviço, armazene seus dados nesses locais.

Mantenha suas imagens por perto. Quanto maior a latência da rede, mais tempo leva para baixar a imagem. Hospede suas imagens em um repositório no mesmoAWSRegião em que sua carga de

trabalho está. O Amazon ECR é um repositório de imagens de alto desempenho que está disponível em todas as regiões em que o Amazon ECS está disponível. Evite atravessar a Internet ou um link VPN para baixar imagens de contêiner. Hospedar suas imagens na mesma região melhora a confiabilidade geral. Ele reduz o risco de problemas de conectividade de rede e problemas de disponibilidade em uma região diferente. Como alternativa, você também pode implementar a replicação entre regiões do Amazon ECR para ajudar com isso.

Reduzir os limites da verificação de integridade do load balancer do. Os balanceadores de carga executam verificações de integridade antes de enviar tráfego para seu aplicativo. A configuração de verificação de integridade padrão para um grupo de destino pode levar 90 segundos ou mais. Durante isso, ele verifica o status de integridade e as solicitações de recebimento. Reduzir o intervalo de verificação de integridade e a contagem de limites pode fazer com que seu aplicativo aceite tráfego mais rapidamente e reduzir a carga em outras tarefas.

Considere o desempenho de arranque a frio. Alguns aplicativos usam tempos de execução, como Java executar compilação Just-In-Time (JIT). O processo de compilação pelo menos como ele é iniciado pode mostrar o desempenho do aplicativo. Uma solução alternativa é reescrever as partes críticas de latência da sua carga de trabalho em idiomas que não impõem uma penalidade de desempenho de arranque a frio.

Use políticas de escalabilidade em etapas e não políticas de escalabilidade de rastreamento de destino. Você tem várias opções de Application Auto Scaling para tarefas do Amazon ECS. O rastreamento de alvos é o modo mais fácil de usar. Com ele, tudo o que você precisa fazer é definir um valor de destino para uma métrica, como a utilização média da CPU. Em seguida, o auto scaler gerencia automaticamente o número de tarefas necessárias para atingir esse valor. No entanto, recomendamos que você use o dimensionamento de etapas em vez disso, para que você possa reagir mais rapidamente às alterações na demanda. Com o dimensionamento de etapas, você define os limites específicos para suas métricas de dimensionamento e quantas tarefas serão adicionadas ou removidas quando os limites forem ultrapassados. E, o mais importante, você pode reagir muito rapidamente às mudanças na demanda minimizando o tempo que um alarme de limite está violando. Para obter mais informações, consulteAuto Scaling de serviçonoAmazon Elastic Container Service Developer.

Se você estiver usando instâncias do Amazon EC2 para fornecer capacidade de cluster, considere as seguintes recomendações:

Use instâncias maiores do Amazon EC2 e volumes do Amazon EBS mais rápidos. Você pode melhorar as velocidades de download e preparação de imagens usando uma instância maior do

Amazon EC2 e um volume mais rápido do Amazon EBS. Dentro de uma determinada família de instâncias do Amazon EC2, a taxa de transferência máxima da rede e do Amazon EBS aumenta à medida que o tamanho da instância aumenta (por exemplo, de`m5.xlarge`para`m5.2xlarge`). Além disso, você também pode personalizar volumes do Amazon EBS para aumentar sua taxa de transferência e IOPS. Por exemplo, se estiver usando o`gp2`, use volumes maiores que ofereçam mais taxa de transferência de linha de base. Se estiver usando o`gp3`Os volumes da, especifique a taxa de transferência e o IOPS ao criar o volume.

Use o modo de rede de ponte para tarefas executadas em instâncias do Amazon EC2. Tarefas que usam`bridge`no Amazon EC2 são iniciados mais rapidamente do que as tarefas que usam o`awsvpc`Modo de rede. Quando`awsvpc`O Amazon ECS anexa uma elastic network interface (ENI) à instância antes de iniciar a tarefa. Isso introduz latência adicional. No entanto, existem várias compensações para o uso de redes de pontes. Essas tarefas não obtêm seu próprio grupo de segurança e há algumas implicações para o balanceamento de carga. Para obter mais informações, consulteGrupos de destino do balanceador de cargaноElastic Load Balancing Guia.

Manipulando choques de demanda

Algumas aplicações experimentam grandes choques repentinos na demanda. Isso acontece por uma variedade de razões: um evento de notícias, grande venda, evento de mídia ou algum outro evento que se torna viral e faz com que o tráfego aumente rápida e significativamente em um curto espaço de tempo. Se não for planejado, isso pode fazer com que a demanda ultrapasse rapidamente os recursos disponíveis.

A melhor maneira de lidar com choques de demanda é antecipá-los e planejar adequadamente. Como o dimensionamento automático pode levar tempo, recomendamos que você dimensione seu aplicativo antes que o choque de demanda comece. Para obter os melhores resultados, recomendamos ter um plano de negócios que envolva uma estreita colaboração entre equipes que usam um calendário compartilhado. A equipe que está planejando o evento deve trabalhar em estreita colaboração com a equipe responsável pela inscrição com antecedência. Isso dá a essa equipe tempo suficiente para ter um plano de agendamento claro. Eles podem programar a capacidade para aumentar a escala antes do evento e para aumentar a escala após o evento. Para obter mais informações, consulteEscalabilidade programadanoGuia do usuário do Application Auto Scaling.

Se você tiver um plano de Support Empresarial, certifique-se também de trabalhar com seu Gerente Técnico de Conta (TAM). Seu TAM pode verificar suas cotas de serviço e garantir que todas as cotas necessárias sejam levantadas antes do início do evento. Desta forma, você não atinge

acidentalmente nenhuma cota de serviço. Eles também podem ajudá-lo pré-aquecimento de serviços, como balanceadores de carga, para garantir que seu evento corra sem problemas.

Lidar com choques de demanda não programados é um problema mais difícil. Choques não programados, se grandes o suficiente em amplitude, podem rapidamente fazer com que a demanda ultrapasse a capacidade. Ele também pode superar a capacidade de autoscaling reagir. A melhor maneira de se preparar para choques não programados é provisionar recursos em excesso. Você deve ter recursos suficientes para lidar com a demanda máxima de tráfego prevista a qualquer momento.

Manter a capacidade máxima em antecipação a choques de demanda não programados pode ser dispendioso. Para atenuar o impacto no custo, encontre uma métrica ou evento indicador líder que preveja que um choque de grande demanda está iminente. Se a métrica ou evento fornecer um aviso prévio significativo de forma confiável, inicie o processo de expansão imediatamente quando o evento ocorrer ou quando a métrica ultrapassar o limite específico definido por você.

Se seu aplicativo estiver propenso a choques repentinos de demanda não programados, considere adicionar um modo de alto desempenho ao aplicativo que sacrifique funcionalidades não críticas, mas mantenha funcionalidades cruciais para um cliente. Por exemplo, suponha que seu aplicativo pode alternar de gerar respostas personalizadas caras para servir uma página de resposta estática. Nesse cenário, você pode aumentar a taxa de transferência significativamente sem dimensionar o aplicativo.

Por último, você pode considerar a separação de serviços monolíticos para melhor lidar com choques de demanda. Se o seu aplicativo for um serviço monolítico que é caro de ser executado e lento para dimensionar, talvez você consiga extrair ou reescrever peças críticas de desempenho e executá-las como serviços separados. Esses novos serviços podem ser dimensionados independentemente de componentes menos críticos. Ter a flexibilidade de expandir a funcionalidade crítica de desempenho separadamente de outras partes do aplicativo pode reduzir o tempo necessário para adicionar capacidade e ajudar a economizar custos.

Capacidade do cluster

Anteriormente neste tópico, discutimos como dimensionar a conta de réplica para o seu usando métricas de dimensionamento. Suas tarefas também precisam ser executadas em recursos, incluindo recursos de CPU e memória. Isto relata novamente o tema da capacidade. No Amazon ECS, a capacidade é fornecida por meio de dois provedores principais:AWS Fargatee Amazon EC2.

Você pode fornecer capacidade para um cluster do Amazon ECS de várias maneiras. Por exemplo, você pode executar instâncias do Amazon EC2 e registrá-las com o cluster na inicialização usando o agente de contêiner do Amazon ECS. No entanto, esse método pode ser um desafio porque você precisa gerenciar o dimensionamento por conta própria. Portanto, recomendamos usar provedores de capacidade do Amazon ECS. Eles gerenciam o dimensionamento de recursos para você. Existem três tipos de provedores de capacidade: Amazon EC2, Fargate e Fargate Spot.

Os provedores de capacidade Fargate e Fargate Spot lidam com o ciclo de vida das tarefas Fargate para você. A Fargate oferece capacidade sob demanda, e a Fargate Spot fornece capacidade spot. Quando uma tarefa é iniciada, o ECS provisiona um recurso Fargate para você. Esse recurso Fargate vem com as unidades de memória e CPU que correspondem diretamente aos limites de nível de tarefa que você declarou em sua definição de tarefa. Cada tarefa recebe seu próprio recurso Fargate, fazendo uma relação 1:1 entre a tarefa e computar iresources.

As tarefas executadas no ponto Fargate estão sujeitas a interrupção. As interrupções vêm depois de um aviso de dois minutos. Estes ocorrem durante períodos de grande demanda. O Fargate Spot funciona melhor para cargas de trabalho tolerantes a interrupções, como trabalhos em lote, ambientes de desenvolvimento ou preparo. Eles também são adequados para qualquer outro cenário onde alta disponibilidade e baixa latência não são uma exigência.

Você pode executar tarefas Spot do Fargate juntamente com tarefas sob demanda do Fargate. Ao usá-los juntos, você recebe a capacidade de “intermitência” de provisão a um custo menor.

O ECS também pode gerenciar a capacidade da instância do Amazon EC2 para suas tarefas. Cada provedor de capacidade do Amazon EC2 está associado a um grupo de Auto Scaling do Amazon EC2 especificado por você. Quando você usa o Amazon EC2 Capacity Provider, o Auto-Scaling de cluster do ECS mantém o tamanho do Grupo de Auto Scaling do Amazon EC2 para garantir que todas as tarefas agendadas possam ser colocadas.

Melhores práticas de capacidade de cluster

Adicione espaço para o seu serviço, não para o Provedor de Capacidade. Os provedores de capacidade do Amazon EC2 oferecem um valor de capacidade de destino. Se você definir o valor menor que 100%, o ECS provisiona mais instâncias do Amazon EC2 do que o necessário para acomodar suas tarefas. Ter várias instâncias do Amazon EC2 prontas para aceitar tarefas pode ser útil. No entanto, quando você usa o Amazon Virtual Private Cloud, a execução de novas tarefas requer tempo adicional para baixar a imagem e anexar uma interface de rede. Essa latência adicional pode ser prejudicial ao seu resultado final.

Portanto, recomendamos que você faça o seguinte. Em vez de reduzir a capacidade de destino do provedor de capacidade, aumente o número de réplicas em seu serviço modificando a métrica de dimensionamento de rastreamento de destino ou os limites de escalabilidade de etapas do serviço. Para obter mais informações sobre políticas de escalabilidade relacionadas, consulte.Políticas de escalabilidade de rastreamento de destinoouPolíticas de escalabilidade em etapasnoAmazon Elastic Container Service Developer. O Amazon EC2 Capacity Provider provisiona a capacidade necessária para tarefas adicionais adicionando instâncias adicionais ao Auto Scaling Group. Isso ajuda a garantir que os recursos de computação e aplicativos estejam disponíveis quando você precisar deles. Por exemplo, ele pode ajudar dobrando o número de tarefas em um Serviço ECS para acomodar uma intermitência imediata de 100% na demanda.

Escolhendo tamanhos de tarefa Fargate

Se você executar suas tarefas noAWS Fargate, você deve declarar a CPU e os limites de memória da tarefa em sua definição. O ECS usa esses limites para determinar o tipo de instância do Fargate no qual executar sua tarefa. Os limites que você determina devem ser maiores que ou iguais a quaisquer reservas que você declarou. Na maioria dos casos, você pode configurá-los para a soma das reservas de cada um dos contêineres declarados na definição da tarefa. Em seguida, arredonde também o número para o tamanho de instância Fargate mais próximo. Para obter mais informações sobre os tamanhos disponíveis, consulte.CPU e memória da tarefanoAmazon Elastic Container Service Developer.

Escolhendo o tipo de instância do Amazon EC2

Se você usar o Amazon EC2 para fornecer capacidade para seu cluster ECS, poderá escolher entre uma grande variedade de tipos de instância. Todos os tipos e famílias de instâncias do Amazon EC2 são compatíveis com o ECS.

Para determinar quais tipos de instância você pode usar, comece eliminando os tipos de instância ou as famílias de instâncias que não atendem aos requisitos específicos do seu aplicativo. Por exemplo, se o aplicativo exigir uma GPU, você pode excluir qualquer tipo de instância que não tenha uma GPU. No entanto, você também deve considerar outros requisitos, também. Por exemplo, considere a arquitetura da CPU, a taxa de transferência da rede e se o armazenamento da instância é um requisito. Em seguida, examine a quantidade de CPU e memória fornecida por cada tipo de instância. Como regra geral, a CPU e a memória devem ser grandes o suficiente para conter pelo menos uma réplica da tarefa que você deseja executar.

Você pode escolher entre os tipos de instância compatíveis com seu aplicativo. Com instâncias maiores, você pode executar mais tarefas ao mesmo tempo. E, com instâncias menores, você pode dimensionar horizontalmente de forma mais refinada para economizar custos. Você não precisa escolher um único tipo de instância do Amazon EC2 que se adapte a todos os aplicativos em seu cluster. Em vez disso, você pode criar vários grupos de Auto Scaling,. Cada grupo de pode ter um tipo de instância diferente. Em seguida, você pode criar um provedor de capacidade do Amazon EC2 para cada um desses grupos. Por último, na estratégia do Provedor de Capacidade de seu serviço e tarefa, você pode selecionar o Provedor de Capacidade que melhor se adapte às suas necessidades.

Usando Amazon EC2 Spot e FARGATE_SPOT

A capacidade spot pode proporcionar uma economia significativa em relação às instâncias sob demanda. A capacidade spot é o excesso de capacidade cujo preço é significativamente menor do que a capacidade sob demanda ou reservada. A capacidade spot é adequada para cargas de trabalho de processamento em lote e aprendizado de máquina, além de ambientes de desenvolvimento e preparo. De um modo mais geral, é adequado para qualquer carga de trabalho que tolere tempo de inatividade temporário.

Entenda que as seguintes consequências porque a capacidade spot pode não estar disponível o tempo todo.

- Primeiro, durante períodos de demanda extremamente alta, a capacidade spot pode estar indisponível. Isso pode fazer com que a tarefa Spot Fargate e as execuções da instância spot do Amazon EC2 sejam atrasadas. Nesses eventos, os serviços do ECS tentam executar tarefas novamente, e os grupos do Amazon EC2 Auto Scaling também tentam executar instâncias novamente, até que a capacidade necessária fique disponível. O Fargate e o Amazon EC2 não substituem a capacidade spot por capacidade sob demanda.
- Em segundo lugar, quando a demanda geral por capacidade aumenta, as instâncias spot e as tarefas podem ser encerradas com apenas um aviso de dois minutos. Depois que o aviso é enviado, as tarefas devem iniciar um desligamento ordenado, se necessário, antes que a instância seja totalmente encerrada. Isso ajuda a minimizar a possibilidade de erros. Para obter mais informações sobre um encerramento normal, consulte.Desligamentos graciosos com ECS.

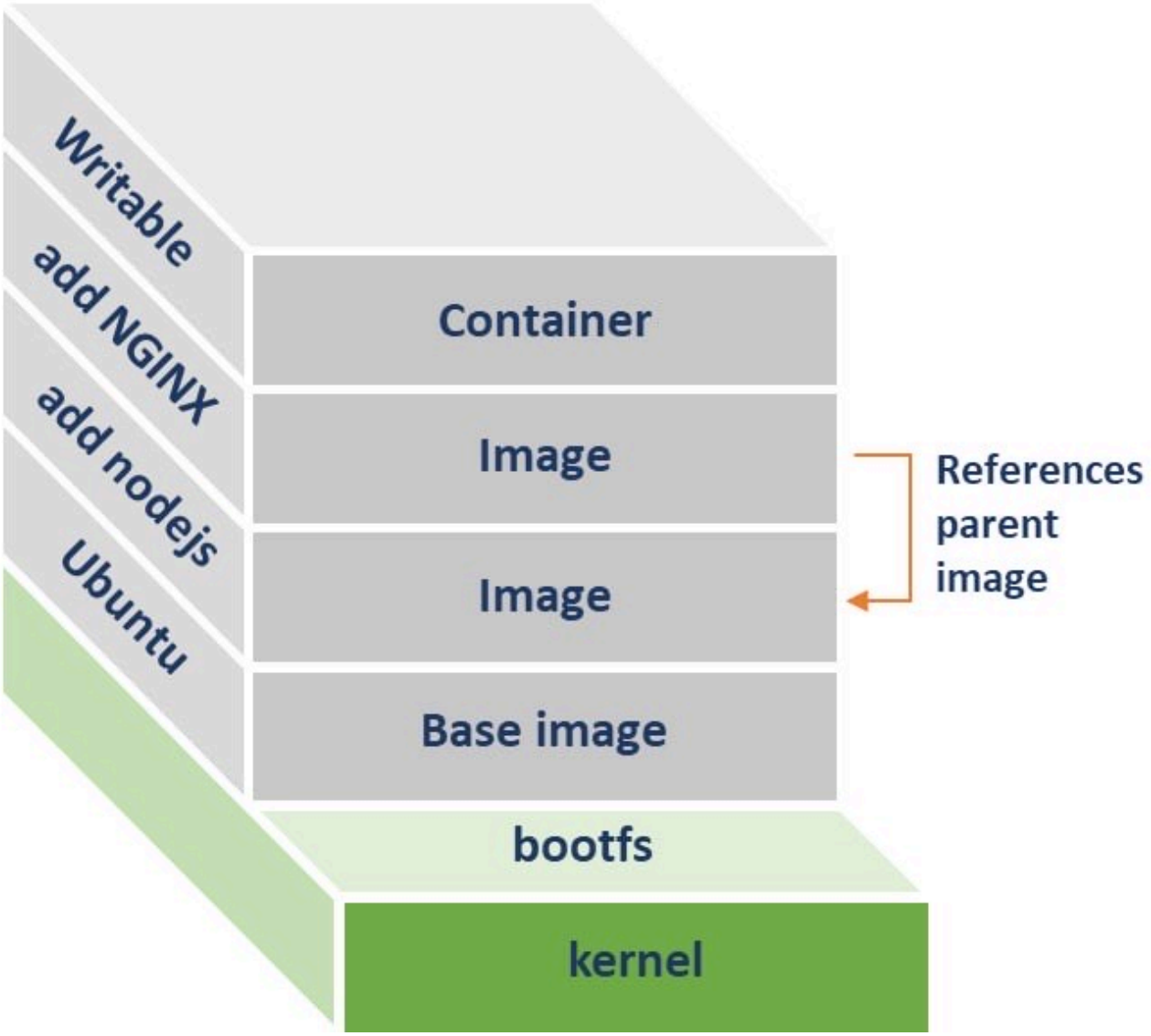
Para ajudar a minimizar a falta de capacidade spot, considere as seguintes recomendações:

- Use várias regiões e zonas de disponibilidade. A capacidade spot varia de acordo com a região e zona de disponibilidade. Você pode melhorar a disponibilidade spot executando suas cargas de trabalho em várias regiões e zonas de disponibilidade. Se possível, especifique sub-redes em todas as Zonas de Disponibilidade nas Regiões onde você executa suas tarefas e instâncias.
- Use vários tipos de instância do Amazon EC2. Quando você usa políticas de instância mista com o Amazon EC2 Auto Scaling, vários tipos de instância são iniciados no seu grupo Auto Scaling. Isso garante que uma solicitação de capacidade spot possa ser atendida quando necessário. Para maximizar a confiabilidade e minimizar a complexidade, use tipos de instância com aproximadamente a mesma quantidade de CPU e memória em sua Política de Instâncias Mistas. Essas instâncias podem ser de uma geração diferente ou variantes do mesmo tipo de instância base. Observe que eles podem vir com recursos adicionais que você pode não precisar. Um exemplo dessa lista pode incluir m4.large, m5.large, m5a.large, m5d.large, m5n.large, m5dn.large e m5ad.large. Para obter mais informações, consulte Grupos de Auto Scaling com vários tipos de instância e opções de compra no Guia do usuário do Auto Scaling do Amazon EC2.
- Use a estratégia de alocação spot otimizada para capacidade. Com o Amazon EC2 Spot, você pode escolher entre as estratégias de alocação otimizadas para capacidade e custo. Se você escolher a estratégia otimizada para capacidade ao iniciar uma nova instância, o Amazon EC2 Spot selecionará o tipo de instância com maior disponibilidade na Zona de disponibilidade selecionada. Isso ajuda a reduzir a possibilidade de que a instância seja encerrada logo após ser iniciada.

Práticas recomendadas - Armazenamento persistente

Você pode usar o Amazon ECS para executar aplicativos em contêineres com estado em escala usandoAWSserviços de armazenamento, como Amazon EFS, Amazon EBS ou Amazon FSx for Windows File Server, que fornecem persistência de dados para contêineres inerentemente efêmeros. O termoPersistência de dadossignifica que os dados em si duram mais do que o processo que os criou. Persistência de dados noAWSé alcançado através do acoplamento de serviços de computação e armazenamento. Semelhante ao Amazon EC2, você também pode usar o Amazon ECS para dissociar o ciclo de vida de seus aplicativos em contêineres dos dados que eles consomem e produzem. O uso doAWS, as tarefas do Amazon ECS podem persistir dados mesmo após o término das tarefas.

Por padrão, os contêineres não persistem os dados que produzem. Quando um contêiner é encerrado, os dados que ele gravou em sua camada gravável são destruídos com o contêiner. Isso torna os contêineres adequados para aplicativos sem estado que não precisam armazenar dados localmente. Os aplicativos em contêineres que exigem persistência de dados precisam de um back-end de armazenamento que não seja destruído quando o contêiner do aplicativo é encerrado.



Uma imagem de contêiner é construída a partir de uma série de camadas. Cada camada representa uma instrução no Dockerfile a partir da qual a imagem foi criada. Cada camada é somente leitura, exceto para o contêiner. Ou seja, quando você cria um contêiner, uma camada gravável é adicionada sobre as camadas subjacentes. Todos os arquivos que o contêiner cria, exclui ou modifica são gravados na camada gravável. Quando o contêiner termina, a camada gravável também é excluída simultaneamente. Um novo contêiner que usa a mesma imagem tem sua própria camada gravável. Esta camada não inclui alterações. Portanto, os dados de um contêiner devem sempre ser armazenados fora da camada gravável do contêiner.

Com o Amazon ECS, você pode executar contêineres com estado usando volumes. O Amazon ECS é integrado ao Amazon EFS nativamente e usa volumes integrados ao Amazon EBS. Para

contêineres do Windows, o Amazon ECS se integra ao Amazon FSx for Windows File Server para fornecer armazenamento persistente.

Tópicos

Escolher o tipo de armazenamento certo para seus contêineres

Os aplicativos que estão sendo executados em um cluster do Amazon ECS podem usar uma variedade deAWSServiços de armazenamento e produtos de terceiros para fornecer armazenamento persistente para cargas de trabalho com estado. Você deve escolher o back-end de armazenamento para o aplicativo em contêineres com base nos requisitos de arquitetura e armazenamento do aplicativo. Para obter mais informações sobreAWSserviços de armazenamento, consulteArmazenamento na nuvem noAWS.

Para clusters do Amazon ECS que contêm instâncias Linux ou são usados com o Fargate, o Amazon ECS se integra ao Amazon EFS e ao Amazon EBS para fornecer armazenamento de contêiner. A diferença mais distinta entre o Amazon EFS e o Amazon EBS é que você pode montar simultaneamente um sistema de arquivos do Amazon EFS em milhares de tarefas do Amazon ECS. Por outro lado, os volumes do Amazon EBS não suportam acesso simultâneo. Diante disso, o Amazon EFS é a opção de armazenamento recomendada para aplicativos em contêineres que são dimensionados horizontalmente. Isso ocorre porque ele suporta simultaneidade. O Amazon EFS armazena seus dados de forma redundante em várias zonas de disponibilidade e oferece acesso de baixa latência a partir de tarefas do Amazon ECS, independentemente da zona de disponibilidade. O Amazon EFS oferece suporte a tarefas executadas no Amazon EC2 e no Fargate.

Suponha que você tenha um aplicativo como um banco de dados transacional que requer latência abaixo de milissegundos e não precisa de um sistema de arquivos compartilhado quando ele é dimensionado horizontalmente. Para esse aplicativo, recomendamos o uso de volumes do Amazon EBS para armazenamento persistente. Atualmente, o Amazon ECS oferece suporte a volumes do Amazon EBS para tarefas hospedadas somente no Amazon EC2. O Support para volumes do Amazon EBS não está disponível para tarefas no Fargate. Antes de usar volumes do Amazon EBS com tarefas do Amazon ECS, você deve primeiro anexar volumes do Amazon EBS a instâncias de contêiner e gerenciar volumes separadamente do ciclo de vida da tarefa.

Para clusters que contêm instâncias do Windows, o Amazon FSx for Windows File Server fornece armazenamento persistente para contêineres. Os sistemas de arquivos do Amazon FSx for Windows File Server são compatíveis com implantações Multi-AZ. Por meio dessas implantações, você pode compartilhar um sistema de arquivos com tarefas do Amazon ECS executadas em várias zonas de disponibilidade.

Você também pode usar o armazenamento de instâncias do Amazon EC2 para persistência de dados para tarefas do Amazon ECS hospedadas no Amazon EC2 usando montagens de ligação ou volumes do Docker. Ao usar montagens de vinculação ou volumes do Docker, os contêineres armazenam dados no sistema de arquivos de instância de contêiner. Uma limitação do uso de um sistema de arquivos host para armazenamento de contêiner é que os dados só estão disponíveis em uma única instância de contêiner por vez. Isso significa que os contêineres só podem ser executados no host onde os dados residem. Portanto, o uso de armazenamento de host é recomendado somente em cenários em que a replicação de dados é tratada no nível do aplicativo.

Volume do Amazon EFS

O Amazon Elastic File System (Amazon EFS) fornece um sistema de arquivos NFS elástico, simples, escalável e totalmente gerenciado. Ele foi criado para ser capaz de dimensionar sob demanda para petabytes sem interromper os aplicativos. Ele pode aumentar ou reduzir conforme você adiciona e remove arquivos.

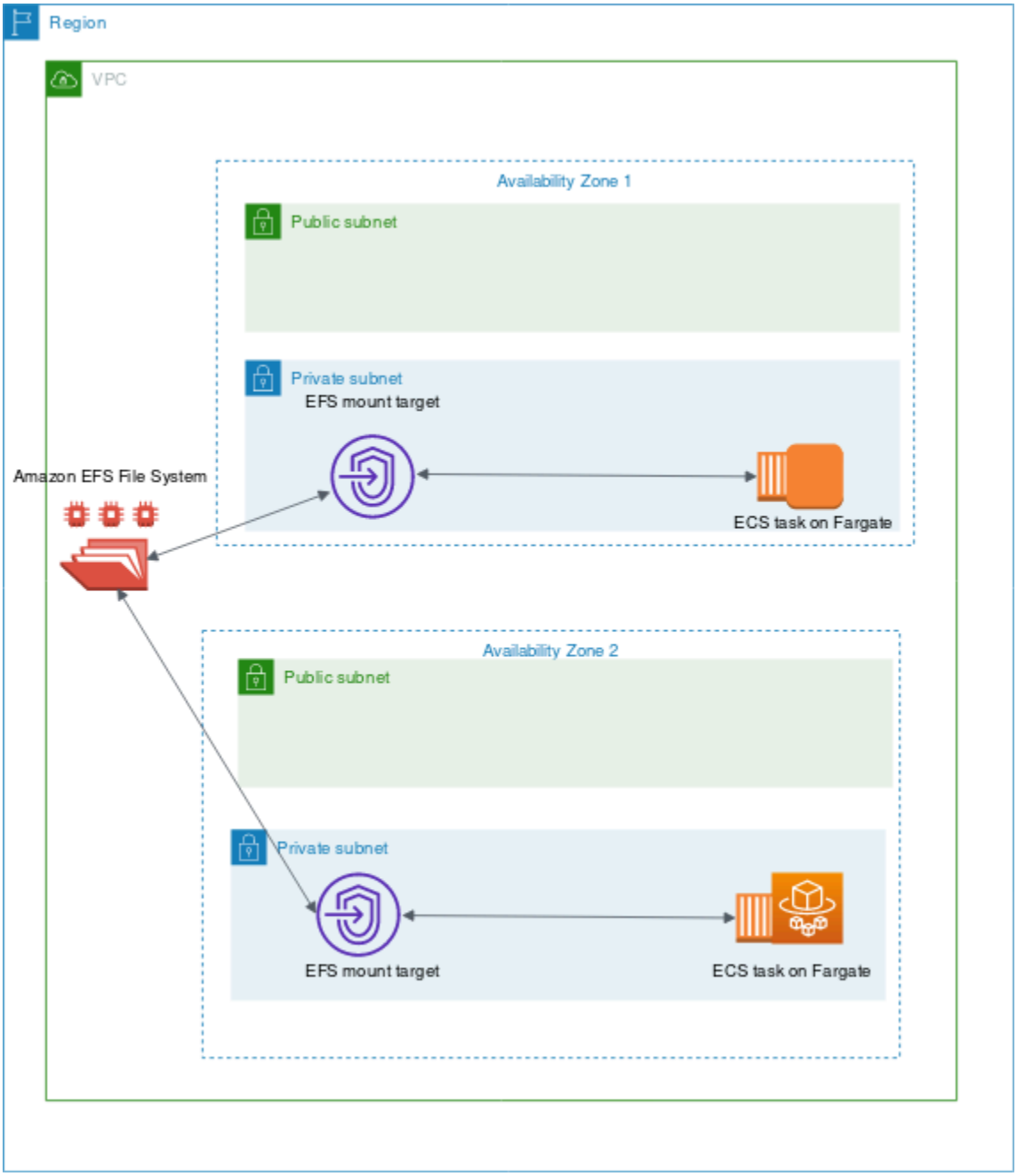
Você pode executar seus aplicativos stateful no Amazon ECS usando volumes do Amazon EFS para fornecer armazenamento persistente. Tarefas do Amazon ECS executadas em instâncias do Amazon EC2 ou no Fargate usando a versão da plataforma`1.4.0`e posterior podem montar um sistema de arquivos do Amazon EFS existente. Dado que vários contêineres podem montar e acessar um sistema de arquivos do Amazon EFS simultaneamente, suas tarefas têm acesso ao mesmo conjunto de dados, independentemente de onde estejam hospedados.

Para montar um sistema de arquivos do Amazon EFS no contêiner, você pode fazer referência ao sistema de arquivos do Amazon EFS e ao ponto de montagem do contêiner na definição de tarefa do Amazon ECS. Veja a seguir um trecho de uma definição de tarefa que usa o Amazon EFS para armazenamento de contêiner.

```
...
"containerDefinitions":[
    {
        "mountPoints": [
```

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

O Amazon EFS armazena dados de forma redundante em várias zonas de disponibilidade em uma única região. Uma tarefa do Amazon ECS monta o sistema de arquivos do Amazon EFS usando um destino de montagem do Amazon EFS em sua zona de disponibilidade. Uma tarefa do Amazon ECS só pode montar um sistema de arquivos do Amazon EFS se o sistema de arquivos do Amazon EFS tiver um destino de montagem na Zona de disponibilidade na qual a tarefa é executada. Portanto, uma prática recomendada é criar destinos de montagem do Amazon EFS em todas as zonas de disponibilidade nas quais você planeja hospedar tarefas do Amazon ECS.



Para obter mais informações, consulteVolume do Amazon EFSnoAmazon Elastic Container Service Developer.

Controles de segurança e acesso

O Amazon EFS oferece recursos de controle de acesso que você pode usar para garantir que os dados armazenados em um sistema de arquivos do Amazon EFS sejam seguros e acessíveis somente a partir de aplicativos que precisam deles. Você pode proteger dados habilitando a

criptografia em repouso e em trânsito. Para obter mais informações, consulteCriptografia de dados no Amazon EFSnoAmazon Elastic File System.

Além da criptografia de dados, você também pode usar o Amazon EFS para restringir o acesso a um sistema de arquivos. Há três maneiras de implementar o controle de acesso no EFS.

- Grupos de segurança—Com os destinos de montagem do Amazon EFS, você pode configurar um grupo de segurança usado para permitir e negar tráfego de rede. Você pode configurar o grupo de segurança anexado ao Amazon EFS para permitir o tráfego NFS (porta 2049) do security group anexado às suas instâncias do Amazon ECS ou, ao usar o`awsvpc`, a tarefa do Amazon ECS.
- IAM— Você pode restringir o acesso a um sistema de arquivos do Amazon EFS usando o IAM. Quando configuradas, as tarefas do Amazon ECS exigem uma função do IAM para o acesso ao sistema de arquivos para montar um sistema de arquivos EFS. Para obter mais informações, consulteUso do IAM para controlar o acesso a dados do sistema denoAmazon Elastic File System.

  As políticas do IAM também podem impor condições predefinidas, como exigir que um cliente use TLS ao se conectar a um sistema de arquivos do Amazon EFS. Para obter mais informações, consulteChaves de condição do Amazon EFS para clientesnoAmazon Elastic File System.
- Pontos de acesso do Amazon EFS—Os pontos de acesso do Amazon EFS são pontos de entrada específicos do aplicativo em um sistema de arquivos do Amazon EFS. Você pode usar pontos de acesso para impor uma identidade de usuário, incluindo os grupos POSIX do usuário, para todas as solicitações do sistema de arquivos feitas por meio do ponto de acesso. Os pontos de acesso também podem impor um diretório raiz diferente para o sistema de arquivos. Isto é para que os clientes só possam acessar dados no diretório especificado ou seus subdiretórios.

Considere implementar todos os três controles de acesso em um sistema de arquivos do Amazon EFS para obter segurança máxima. Por exemplo, você pode configurar o grupo de segurança anexado a um ponto de montagem do Amazon EFS para permitir apenas o tráfego NFS de entrada de um grupo de segurança associado à instância de contêiner ou à tarefa do Amazon ECS. Além disso, você pode configurar o Amazon EFS para exigir uma função do IAM para acessar o sistema de arquivos, mesmo que a conexão seja originada de um grupo de segurança permitido. Por último, você pode usar pontos de acesso do Amazon EFS para impor permissões de usuário POSIX e especificar diretórios raiz para aplicativos.

O trecho de definição de tarefa a seguir mostra como montar um sistema de arquivos do Amazon EFS usando um ponto de acesso.

```
"volumes": [
```

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

Performance

O Amazon EFS oferece dois modos de desempenho: Uso geral e E/S máx. O propósito geral é adequado para aplicativos sensíveis à latência, como sistemas de gerenciamento de conteúdo e ferramentas de CI/CD. Em contraste, os sistemas de arquivos Max I/O são adequados para cargas de trabalho, como análise de dados, processamento de mídia e aprendizado de máquina. Essas cargas de trabalho precisam executar operações paralelas de centenas ou mesmo milhares de contêineres e exigem a maior taxa de transferência agregada possível e IOPS. Para obter mais informações, consulteModo de desempenho do Amazon EFSnoAmazon Elastic File System.

Algumas cargas de trabalho sensíveis à latência exigem os níveis de E/S mais altos fornecidos pelo modo de desempenho máximo de E/S e a latência mais baixa fornecida pelo modo de desempenho de uso geral. Para esse tipo de carga de trabalho, recomendamos a criação de vários sistemas de arquivos do modo de desempenho de uso geral. Dessa forma, você pode distribuir a carga de trabalho do aplicativo entre todos esses sistemas de arquivos, desde que a carga de trabalho e os aplicativos possam oferecer suporte a ela.

Throughput

Todos os sistemas de arquivos do Amazon EFS têm uma taxa de transferência limitada associada que é determinada pela quantidade de throughput provisionada para sistemas de arquivos usandoTaxa de transferência provisionadaou a quantidade de dados armazenados na classe de armazenamento EFS Standard ou One Zone para sistemas de arquivos usandoThroughput de intermitência. Para obter mais informações, consulteNoções básicas do throughput monitoradonoAmazon Elastic File System.

O modo de taxa de transferência padrão para sistemas de arquivos do Amazon EFS é o modo de intermitência. Com o modo de intermitência, a taxa de transferência disponível para um sistema de arquivos aumenta ou diminui à medida que um sistema de arquivos cresce. Como normalmente as cargas de trabalho baseadas em arquivos são de natureza variável com picos, exigindo altos níveis de taxa de transferência por certos períodos de tempo e níveis de taxa de transferência no restante do tempo, o Amazon EFS foi criado para intermitência, permitindo altos níveis de taxa de transferência durante certos períodos de tempo. Além disso, como muitas cargas de trabalho são pesadas na leitura, as operações de leitura são monitoradas em uma proporção de 1:3 para outras operações NFS (como gravação).

Todos os sistemas de arquivos do Amazon EFS oferecem um desempenho de linha de base consistente de 50 MB/s para cada TB de armazenamento Amazon EFS Standard ou Amazon EFS One Zone. Todos os sistemas de arquivos (independentemente do tamanho) podem explodir para 100 MB/s. Os sistemas de arquivos com mais de 1 TB de armazenamento EFS Standard ou EFS One Zone podem atingir 100 MB/s para cada TB. Como as operações de leitura são monitoradas em uma proporção de 1:3, você pode direcionar até 300 MIBS/s para cada TiB de throughput de leitura. À medida que você adiciona dados ao sistema de arquivos, a taxa de transferência máxima disponível para o sistema de arquivos é dimensionada de forma linear e automática com seu armazenamento na classe de armazenamento padrão do Amazon EFS. Se você precisar de mais throughput do que pode obter com a quantidade de dados armazenada, poderá configurar o Throughput Provisionado para a quantidade específica que sua carga de trabalho requer.

A taxa de transferência do sistema de arquivos é compartilhada entre todas as instâncias do Amazon EC2 conectadas a um sistema de arquivos. Por exemplo, um sistema de arquivos de 1 TB que pode atingir 100 MB/s de taxa de transferência pode acionar 100 MB/s de uma única instância do Amazon EC2 pode cada unidade de 10 MB/s. Para obter mais informações, consulteDesempenho do Amazon EFSnoAmazon Elastic File System.

Otimização de custos

O Amazon EFS simplifica o dimensionamento do armazenamento para você. Os sistemas de arquivos do Amazon EFS crescem automaticamente à medida que você adiciona mais dados. Especialmente com o Amazon EFSThroughput de intermitênciaO taxa de transferência no Amazon EFS é dimensionada à medida que aumenta o tamanho do sistema de arquivos na classe de armazenamento padrão. Para melhorar a taxa de transferência sem pagar um custo adicional para a taxa de transferência provisionada em um sistema de arquivos EFS, você pode compartilhar um sistema de arquivos do Amazon EFS com vários aplicativos. Usando pontos de acesso do Amazon EFS, você pode implementar o isolamento de armazenamento em sistemas de arquivos

compartilhados do Amazon EFS. Ao fazer isso, mesmo que os aplicativos ainda compartilhem o mesmo sistema de arquivos, eles não podem acessar dados a menos que você os autorize.

À medida que seus dados crescem, o Amazon EFS ajuda você a mover automaticamente arquivos acessados com pouca frequência para uma classe de armazenamento mais baixa. A classe de armazenamento IA (Standard-Infreqüent Access () do Amazon EFS reduz os custos de armazenamento de arquivos que não são acessados todos os dias. Ele faz isso sem sacrificar a alta disponibilidade, a alta durabilidade, a elasticidade e o acesso ao sistema de arquivos POSIX oferecidos pelo Amazon EFS. Para obter mais informações, consulteClasse de armazenamento do Amazon EFSnoAmazon Elastic File System.

Considere usar as políticas de ciclo de vida do Amazon EFS para economizar dinheiro automaticamente movendo arquivos acessados com pouca frequência para o armazenamento IA do Amazon EF Para obter mais informações, consulteGerenciamento do ciclo EFS vida do AmazonnoAmazon Elastic File System.

Ao criar um sistema de arquivos do Amazon EFS, você pode escolher se o Amazon EFS replica seus dados em várias zonas de disponibilidade (padrão) ou armazena seus dados de forma redundante em uma única zona de disponibilidade. A classe de armazenamento do Amazon EFS One Zone pode reduzir os custos de armazenamento em uma margem significativa em comparação com as classes de armazenamento padrão do Amazon EFS. Considere usar a classe de armazenamento do Amazon EFS One Zone para cargas de trabalho que não exigem resiliência Multi-AZ. Você pode reduzir ainda mais o custo do armazenamento do Amazon EFS One Zone movendo arquivos acessados com pouca frequência para o Amazon EFS One Zone-Unfrequent Access Access. Para obter mais informações, consulteAmazon EFS.

Proteção de dados

O Amazon EFS armazena seus dados de forma redundante em várias zonas de disponibilidade para sistemas de arquivos usando classes de armazenamento padrão. Se você selecionar classes de armazenamento do Amazon EFS One Zone, seus dados serão armazenados redundantemente em uma única Zona de disponibilidade. Além disso, o Amazon EFS foi projetado para fornecer 99,999999999% (11 9) de durabilidade em um determinado ano.

Como em qualquer ambiente, é uma prática recomendada ter um backup e criar proteções contra exclusão acidental. Para dados do Amazon EFS, essa prática recomendada inclui um backup funcional e testado regularmente usandoAWS Backup. Os sistemas de arquivos que usam classes de armazenamento do Amazon EFS One Zone são configurados para fazer backup automático de arquivos por padrão na criação do sistema de arquivos, a menos que você opte por desativar

essa funcionalidade. Para obter mais informações, consulteProteção de dados para o Amazon EFSnoAmazon Elastic File System.

Casos de uso

O Amazon EFS fornece acesso compartilhado paralelo que aumenta e diminui automaticamente à medida que arquivos são adicionados e removidos. Como resultado, o Amazon EFS é adequado para qualquer aplicativo que exija um armazenamento com funcionalidades como baixa latência, alto throughput e consistência de leitura após gravação. O Amazon EFS é um backend de armazenamento ideal para aplicativos que escalam horizontalmente e exigem um sistema de arquivos compartilhado. Cargas de trabalho como análise de dados, processamento de mídia, gerenciamento de conteúdo e serviço na web são alguns dos casos de uso comuns do Amazon EFS.

Um caso de uso em que o Amazon EFS pode não ser adequado é para aplicativos que exigem latência inferior a milissegundos. Isso geralmente é um requisito para sistemas de banco de dados transacionais. Recomendamos a execução de testes de desempenho de armazenamento para determinar o impacto do uso do Amazon EFS para aplicativos sensíveis à latência. Se o desempenho do aplicativo se degradar ao usar o Amazon EFS, considere o Amazon EBS io2 Block Express, que fornece latência de E/S de baixa variação de submilissegundos em instâncias Nitro. Para obter mais informações, consulte Tipos de volumes do Amazon EBS no Guia do usuário do Amazon EC2 para instâncias do Linux.

Alguns aplicativos falham se seu armazenamento subjacente for alterado inesperadamente. Portanto, o Amazon EFS não é a melhor escolha para esses aplicativos. Em vez disso, você pode preferir usar um sistema de armazenamento que não permita acesso simultâneo de vários locais.

Volumes do Docker

Os volumes do Docker são um recurso do tempo de execução do contêiner do Docker que permite que os contêineres persistam dados montando um diretório do sistema de arquivos do host. Drivers de volume do Docker (também conhecidas como plug-ins) são usadas para integrar volumes de contêiner com sistemas de armazenamento externos, como Amazon EBS. Os volumes do Docker só são compatíveis com a hospedagem de tarefas do Amazon ECS em instâncias do Amazon EC2.

As tarefas do Amazon ECS podem usar volumes do Docker para persistir dados usando volumes do Amazon EBS. Isso é feito anexando um volume do Amazon EBS a uma instância do Amazon EC2 e, em seguida, montando o volume em uma tarefa usando volumes do Docker. Um volume Docker pode ser compartilhado entre várias tarefas do Amazon ECS no host.

A limitação dos volumes do Docker é que o sistema de arquivos que a tarefa usa está vinculado à instância específica do Amazon EC2. Se a instância for interrompida por qualquer motivo e a tarefa for colocada em outra instância, os dados serão perdidos. Você pode atribuir tarefas a instâncias para garantir que os volumes do EBS associados estejam sempre disponíveis para tarefas.

Para obter mais informações, consulteVolumes do DockernoAmazon Elastic Container Service Developer.

Volume de vida do Amazon EBS

Há dois padrões de uso principais com armazenamento de contêineres e Amazon EBS. A primeira é quando um aplicativo precisa persistir dados e evitar a perda de dados quando seu contêiner é encerrado. Um exemplo desse tipo de aplicativo seria um banco de dados transacional como MySQL. Quando uma tarefa MySQL termina, espera-se que outra tarefa a substitua. Nesse cenário, o ciclo de vida do volume é separado do ciclo de vida da tarefa. Ao usar o EBS para persistir dados de contêiner, é uma prática recomendada usar restrições de posicionamento de tarefas para limitar o posicionamento da tarefa a um único host com o volume do EBS anexado.

A segunda é quando o ciclo de vida do volume é independente do ciclo de vida da tarefa. Isso é especialmente útil para aplicativos que exigem armazenamento de alto desempenho e baixa latência, mas não precisam persistir dados após a tarefa ser encerrada. Por exemplo, uma carga de trabalho ETL que processa grandes volumes de dados pode exigir um armazenamento de alta taxa de transferência. O Amazon EBS é adequado para esse tipo de carga de trabalho, pois fornece volumes de alto desempenho que fornecem até 256.000 IOPS. Quando a tarefa é encerrada, a réplica de substituição pode ser colocada com segurança em qualquer host do Amazon EC2 no cluster. Desde que a tarefa tenha acesso a um back-end de armazenamento que possa atender aos requisitos de desempenho, a tarefa poderá desempenhar sua função. Portanto, nenhuma restrição de posicionamento de tarefa é necessária neste caso.

Se as instâncias do Amazon EC2 em seu cluster tiverem vários tipos de volumes do Amazon EBS anexados a elas, você poderá usar restrições de posicionamento de tarefas para garantir que as tarefas sejam colocadas em instâncias com um volume apropriado do Amazon EBS anexado. Por exemplo, suponha que um cluster tenha algumas instâncias com um`gp2`volume, enquanto outros usam`io1`Volumes. Você pode anexar atributos personalizados a instâncias com`io1`e, em seguida, usar restrições de posicionamento de tarefas para garantir que suas tarefas de E/S intensas sejam sempre colocadas em instâncias de contêiner com`io1`Volumes.

Os seguintes exemplos deAWS CLIO comando é usado para colocar atributos em uma instância de contêiner do Amazon ECS.

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

Disponibilidade de dados do Amazon EBS

Normalmente, os contêineres são de curta duração, criados com frequência e encerrados à medida que os aplicativos são dimensionados horizontalmente. Como prática recomendada, você pode executar cargas de trabalho em várias zonas de disponibilidade para melhorar a disponibilidade de seus aplicativos. O Amazon ECS fornece uma maneira de controlar o posicionamento de tarefas usando estratégias de posicionamento de tarefas e restrições de posicionamento de tarefas. Quando uma carga de trabalho persiste seus dados usando volumes do Amazon EBS, suas tarefas precisam ser colocadas na mesma zona de disponibilidade que o volume do Amazon EBS. Também recomendamos que você defina uma restrição de posicionamento que limite a Zona de disponibilidade na qual uma tarefa pode ser colocada. Isso garante que suas tarefas e seus volumes correspondentes estejam localizados na mesma zona de disponibilidade.

Ao executar tarefas independentes, você pode controlar qual Zona de disponibilidade a tarefa é colocada definindo restrições de posicionamento usando o atributo zona de disponibilidade.

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

Ao executar aplicativos que se beneficiariam da execução em várias zonas de disponibilidade, considere criar um serviço diferente do Amazon ECS para cada zona de disponibilidade. Isso garante que as tarefas que precisam de um volume do Amazon EBS sejam colocadas na mesma zona de disponibilidade do volume associado.

Recomendamos a criação de instâncias de contêiner em cada zona de disponibilidade, anexando volumes do Amazon EBS usandoModelos de execuçãoe adicionandoAtributos personalizadospara as instâncias para diferenciá-las de outras instâncias de contêiner no cluster do Amazon ECS. Ao criar serviços, configure restrições de posicionamento de tarefas para garantir que o Amazon ECS coloque tarefas na zona de disponibilidade e na instância corretas. Para obter mais informações, consulteExemplos de restrição de posicionamento de tarefanoAmazon Elastic Container Service Developer.

Plug-ins de volume do Docker

Os plug-ins do Docker, como o Portworx, fornecem uma abstração entre o volume do Docker e o volume do Amazon EBS. Esses plugins podem criar dinamicamente um volume do Amazon EBS

quando a tarefa que precisa de um volume é iniciada. O Portworx também pode anexar um volume a um novo host quando um contêiner é encerrado, e sua réplica subsequente é colocada em uma instância de contêiner diferente. Ele também replica os dados de volume de cada contêiner entre nós do Amazon ECS e entre zonas de disponibilidade. Para obter mais informações, consulte Portworx.

Amazon FSx for Windows File Server

O Amazon FSx for Windows File Server oferece armazenamento de arquivos totalmente gerenciado, altamente confiável e escalável, acessível pelo protocolo SMB (Server Message Block) padrão do setor. Ele é criado no Windows Server, oferecendo uma ampla variedade de recursos administrativos, como cotas de usuário, restauração de arquivos de usuário final e integração com o Microsoft Active Directory (AD). Ele oferece opções de implantação Single-AZ e Multi-AZ, backups totalmente gerenciados e criptografia de dados em repouso e em trânsito.

O Amazon ECS oferece suporte ao uso do Amazon FSx for Windows File Server nas definições de tarefas do Windows do Amazon ECS, permitindo o armazenamento persistente como um ponto de montagem por meio do protocolo SMBv3 usando um recurso SMB chamado GlobalMappings.

Para configurar a integração do Amazon FSX para Windows File Server e do Amazon ECS, a instância de contêiner do Windows deve ser um membro de domínio em um Active Directory Domain Service (AD DS), hospedado por umAWS Directory Service for Microsoft Active Directory, Active Directory local ou Active Directory auto-hospedado no Amazon EC2.AWS Secrets Manageré usado para armazenar dados confidenciais, como o nome de usuário e senha de uma credencial do Active Directory que é usada para mapear o compartilhamento na instância de contêiner do Windows.

Para usar volumes do Amazon FSx for Windows File Server para seus contêineres, você deve especificar as configurações de volume e ponto de montagem na definição de tarefa. Veja a seguir um trecho de uma definição de tarefa que usa o Amazon FSx for Windows File Server para armazenamento de contêiner.

```
{
 "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 obter mais informações, consulteVolumes do Amazon FSx for Windows File ServernoAmazon Elastic Container Service Developer.

Controles de segurança e acesso

O Amazon FSx for Windows File Server oferece os recursos de controle de acesso a seguir que você pode usar para garantir que os dados armazenados em um sistema de arquivos do Amazon FSx for Windows File Server sejam seguros e acessíveis somente a partir de aplicativos que precisam dele.

Criptografia de dados

O Amazon FSx for Windows File Server é compatível com duas formas de criptografia para sistemas de arquivos. Eles são a criptografia de dados em trânsito e a criptografia em repouso. A criptografia de dados em trânsito é suportada em compartilhamentos de arquivos mapeados em uma instância de contêiner que oferece suporte ao protocolo SMB 3.0 ou mais recente. A criptografia de dados em repouso é ativada automaticamente ao criar um sistema de arquivos do Amazon FSX. O Amazon FSX criptografa automaticamente os dados em trânsito usando a criptografia SMB à medida que você acessa seu sistema de arquivos sem a necessidade de modificar seus aplicativos. Para obter mais informações, consulteCriptografia de dados no Amazon FSxnoAmazon FSx for Windows File Server.

Controle de acesso em nível de pasta usando ACLs do Windows

A instância do Windows Amazon EC2 acessa compartilhamentos de arquivos do Amazon FSX usando credenciais do Active Directory. Usa listas de controle de acesso (ACLs) do Windows para controle de acesso ao nível de pasta e arquivo de granulação fina. Você pode criar várias credenciais, cada uma para uma pasta específica dentro do compartilhamento que mapeia para uma tarefa específica.

No exemplo a seguir, a tarefa tem acesso à pasta`App01`usando uma credencial salva no Secrets Manager. Seu nome de recurso da Amazon (ARN) é`1234`.

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

Em outro exemplo, uma tarefa tem acesso à pasta`App02`usando uma credencial salva no Secrets Manager. Seu ARN é `6789`.

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

Casos de uso

Os contêineres não são projetados para persistir dados. No entanto, alguns aplicativos .NET em contêineres podem exigir pastas locais como armazenamento persistente para salvar as saídas do aplicativo. O Amazon FSx for Windows File Server oferece uma pasta local no contêiner. Isso permite que vários contêineres leitura-gravação no mesmo sistema de arquivos que é apoiado por um compartilhamento SMB.

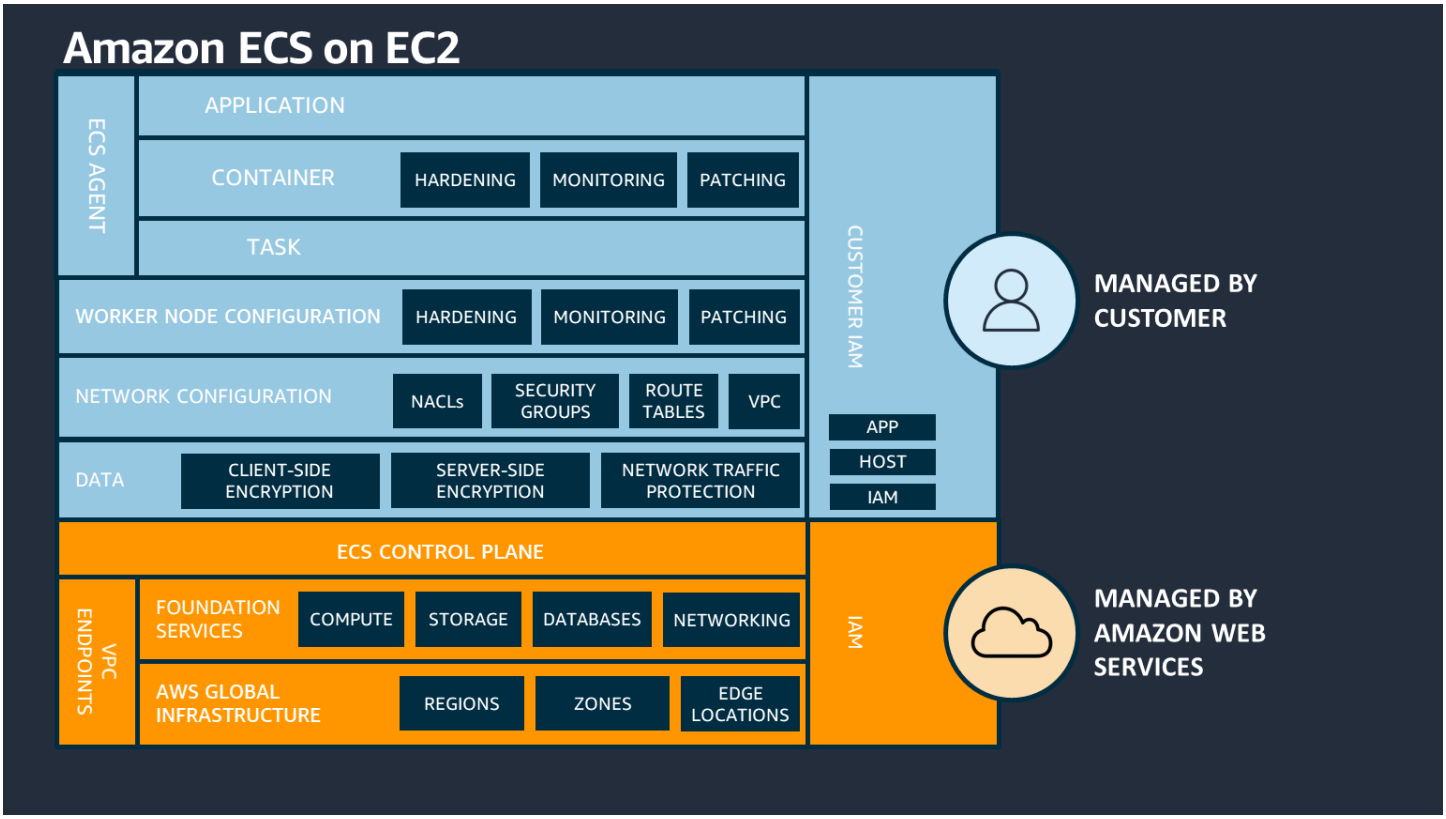
Práticas recomendadas - Segurança

Este guia fornece recomendações de segurança e conformidade para proteger suas informações, sistemas e outros ativos que dependem do Amazon ECS. Ele também apresenta algumas avaliações de risco e estratégias de mitigação que você pode usar para ter um melhor controle sobre os controles de segurança criados para clusters do Amazon ECS e as cargas de trabalho que eles suportam. Cada tópico neste guia começa com uma breve visão geral, seguida de uma lista de recomendações e práticas recomendadas que você pode usar para proteger seus clusters do Amazon ECS.

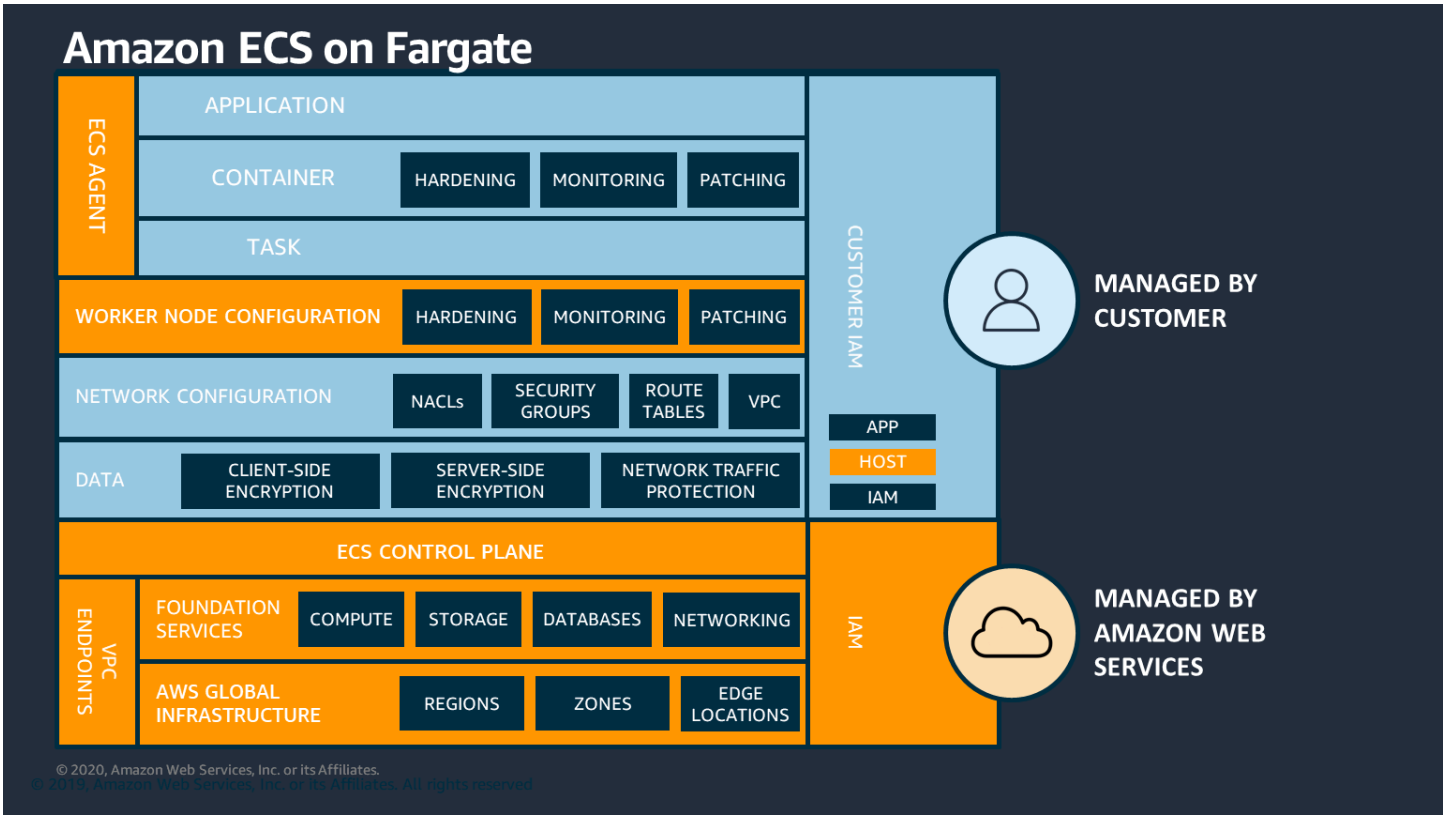
Tópicos

Modelo de responsabilidade compartilhada

A segurança e a conformidade de um serviço gerenciado como o Amazon ECS são uma responsabilidade compartilhada de você eAWS. Em geral,AWSé responsável pela segurança “da” nuvem, enquanto você, o cliente, é responsável pela segurança “na” nuvem.AWSé responsável pelo gerenciamento do plano de controle do Amazon ECS, incluindo a infraestrutura necessária para oferecer um serviço seguro e confiável. E, você é em grande parte responsável pelos tópicos deste guia. Isso inclui segurança de dados, rede e tempo de execução, bem como registro e monitoramento.



No que diz respeito à segurança da infraestrutura,AWSassume mais responsabilidade porAWS Fargatedo que para outras instâncias autogerenciadas. Com Fargate,AWSgerencia a segurança da instância subjacente na nuvem e o tempo de execução usado para executar suas tarefas. A Fargate também dimensiona automaticamente sua infraestrutura em seu nome.



Antes de estender seus serviços para a nuvem, você deve entender quais aspectos de segurança e conformidade são os quais você é responsável.

Para obter mais informações sobre o modelo de responsabilidade compartilhada, consulteModelo de responsabilidade compartilhada.

AWS Identity and Access Management

Você pode usarAWS Identity and Access Management(IAM) para gerenciar e controlar o acesso aoAWSserviços e recursos por meio de políticas baseadas em regras para fins de autenticação e autorização. Mais especificamente, através deste serviço, você controla o acesso ao seuAWSUsando políticas que são aplicadas a usuários, grupos ou funções do IAM. Entre esses três, os usuários do IAM são contas que podem ter acesso aos seus recursos. E, uma função do IAM é um conjunto de permissões que podem ser assumidas por uma identidade autenticada, que não está associada a uma identidade específica fora do IAM. Para obter mais informações, consultePolíticas e permissões no IAM?.

Gerenciando o acesso ao Amazon ECS

Você pode controlar o acesso ao Amazon ECS criando e aplicando políticas do IAM. Essas políticas são compostas por um conjunto de ações que se aplicam a um conjunto específico de recursos. A ação de uma política define a lista de operações (como APIs do Amazon ECS) que são permitidas ou negadas, enquanto o recurso controla quais são os objetos do Amazon ECS aos quais a ação se aplica. Condições podem ser adicionadas a uma política para restringir seu escopo. Por exemplo, uma política pode ser gravada apenas para permitir que uma ação seja executada em tarefas com um determinado conjunto de tags. Para obter mais informações, consulteComo o Amazon ECS funciona com o IAMnoAmazon Elastic Container Service Developer.

Recommendations

Recomendamos fazer o seguinte ao configurar suas funções e políticas do IAM.

Siga a política de acesso menos privilegiado

Crie políticas com escopo para permitir que os usuários executem seus trabalhos prescritos. Por exemplo, se um desenvolvedor precisar interromper periodicamente uma tarefa, crie uma política que permita somente essa ação específica. O exemplo a seguir só permite que um usuário interrompa uma tarefa que pertence a um determinado`task_family`Em um cluster com um nome de recurso da Amazon (ARN) específico. Fazer referência a um ARN em uma condição também é um exemplo de uso de permissões de nível de recurso. Você pode usar permissões de nível de recurso para especificar o recurso ao qual deseja que uma ação seja aplicada ao.

> **Note**
>
> Ao fazer referência a um ARN em uma política, use o novo formato ARN mais longo. Para obter mais informações, consulteNomes de recursos da Amazon (ARNs) e IDsnoAmazon Elastic Container Service Developer.

```
{
  "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>:*"
      ]
    }
  ]
}
```

Permitir que o recurso de cluster sirva como limite administrativo

Políticas com escopo muito restrito podem causar uma proliferação de funções e aumentar a sobrecarga administrativa. Em vez de criar funções com escopo apenas para tarefas ou serviços específicos, crie funções com escopo para clusters e use o cluster como limite administrativo principal.

Isole usuários finais da API do Amazon ECS criando pipelines automatizados

Você pode limitar as ações que os usuários podem usar criando pipelines que empacotam e implantam aplicativos automaticamente em clusters do Amazon ECS. Isso efetivamente delega o trabalho de criação, atualização e exclusão de tarefas para o pipeline. Para obter mais informações, consulteTutorial: Implantação padrão do Amazon ECS com CodePipelinenoAWS CodePipelineGuia do usuário.

Usar condições de política para uma camada adicional de segurança

Quando você precisar de uma camada adicional de segurança, adicione uma condição à sua política. Isso pode ser útil se você estiver executando uma operação privilegiada ou quando precisar restringir o conjunto de ações que podem ser executadas em relação a recursos específicos. A política de exemplo a seguir requer autorização multifator ao excluir um cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

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

As tags aplicadas aos serviços são propagadas para todas as tarefas que fazem parte desse serviço. Por isso, você pode criar funções com escopo para recursos do Amazon ECS com tags específicas. Na política a seguir, um principal do IAM inicia e interrompe todas as tarefas com uma chave de tag de`Departmente` um valor de tag 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"}
            }
        }
    ]
}
```

Auditoria periódica do acesso às APIs do Amazon ECS

Um usuário pode alterar funções. Depois que eles alteram funções, as permissões que foram concedidas anteriormente a eles podem não se aplicar mais. Certifique-se de auditar quem tem acesso às APIs do Amazon ECS e se esse acesso ainda é garantido. Considere integrar o IAM a uma solução de gerenciamento do ciclo de vida do usuário que revoga automaticamente o acesso

quando um usuário sai da organização. Para obter mais informações, consulteDiretrizes de auditoria de segurança do Amazon ECSnoReferência geral da Amazon Web Services.

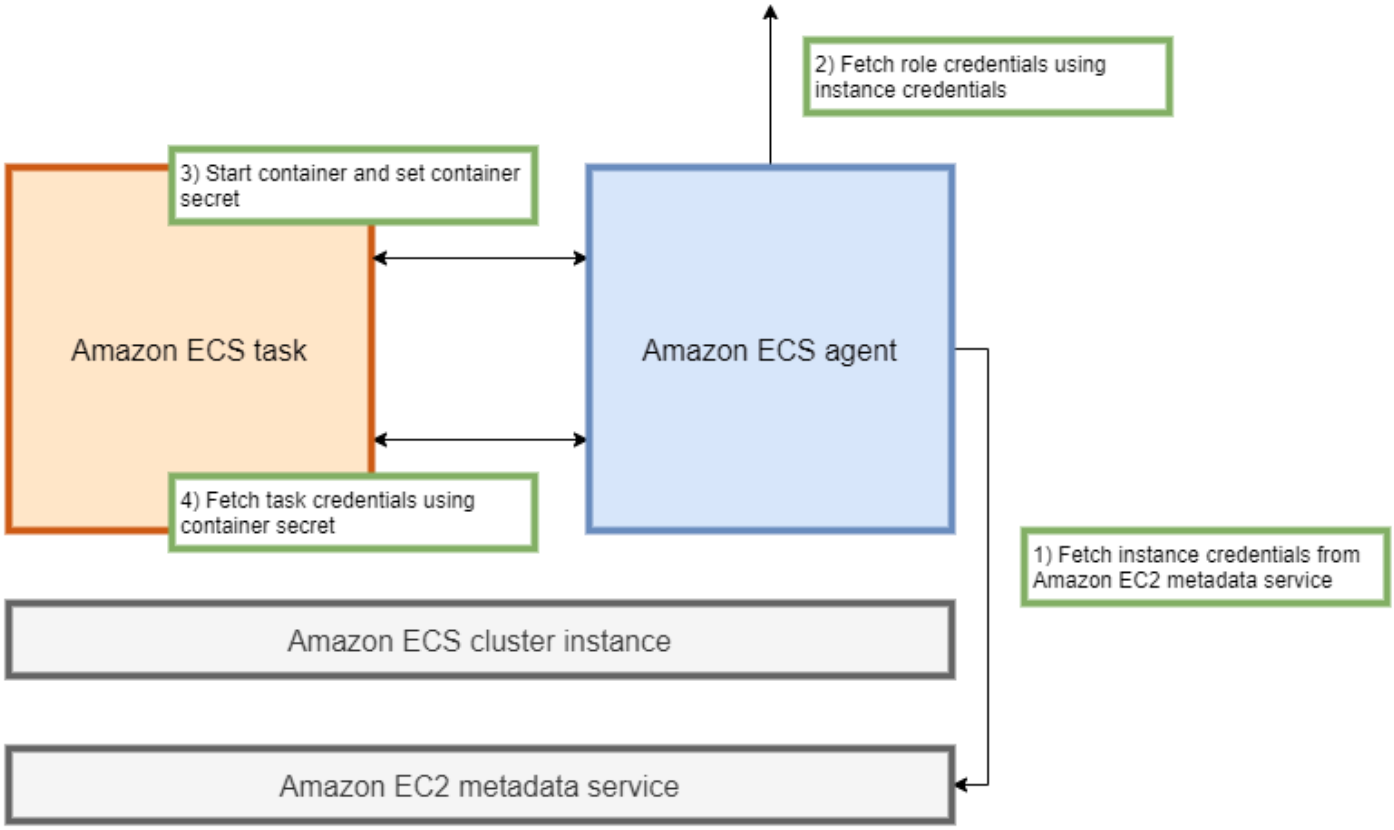
Uso de funções do IAM com tarefas do Amazon ECS

Recomendamos que você atribua uma tarefa uma função do IAM. Sua função pode ser distinguida da função da instância do Amazon EC2 na qual está sendo executada. Atribuir a cada tarefa uma função se alinha ao princípio do acesso menos privilegiado e permite maior controle granular sobre ações e recursos.

Ao atribuir funções do IAM a uma tarefa, você deve usar a política de confiança posterior para que cada uma de suas tarefas possa assumir uma função do IAM diferente daquela que sua instância do EC2 usa. Dessa forma, sua tarefa não herda a função de sua instância do EC2.

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

Quando você adiciona uma função de tarefa a uma definição de tarefa, o agente de contêiner do Amazon ECS cria automaticamente um token com um ID de credencial exclusivo (por exemplo,`12345678-90ab-cdef-1234-567890abcdef`) para a tarefa. Esse token e as credenciais de função são adicionados ao cache interno do agente. O agente preenche a variável de ambiente`AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`no contêiner com o URI do ID de credencial (por exemplo,`/v2/credentials/12345678-90ab-cdef-1234-567890abcdef`).



Você pode recuperar manualmente as credenciais de função temporária de dentro de um contêiner, anexando a variável de ambiente ao endereço IP do agente de contêiner do Amazon ECS e executando `ocurl`na string resultante.

```
curl 192.0.2.0$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

A saída esperada é a seguinte:

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

Versões mais recentes doAWSOs SDKs buscam automaticamente essas credenciais doAWS_CONTAINER_CREDENTIALS_RELATIVE_URIambiente variável ao fazerAWSChamadas de API.

A saída inclui um par de chaves de acesso que consiste em um ID de chave de acesso secreto e uma chave secreta que seu aplicativo usa para acessarAWSrecursos da AWS. Ela também inclui um token queAWSO usa para verificar se as credenciais são válidas. Por padrão, as credenciais atribuídas a tarefas usando funções de tarefa são válidas por seis horas. Depois disso, eles são rotacionados automaticamente pelo agente de contêiner do Amazon ECS.

Função de execução de tarefas

A função de execução da tarefa é usada para conceder ao agente de contêiner do Amazon ECS permissão para chamarAWSAções de API em seu nome. Por exemplo, quando você usaAWS Fargate, o Fargate precisa de uma função do IAM que permita extrair imagens do Amazon ECR e gravar logs no CloudWatch Logs. Uma função do IAM também é necessária quando uma tarefa faz referência a um segredo armazenado noAWS Secrets Manager, como um segredo de extração de imagem.

> **Note**
>
> Se você estiver extraindo imagens como um usuário autenticado, é menos provável que seja afetado pelas alterações que ocorreram noLimites de taxa de pull do Docker Hub. Para obter mais informações, consulteAutenticação de registro privado para instâncias de.
> Ao usar o Amazon ECR e o Amazon ECR Public, você pode evitar os limites impostos pelo Docker. Se você extrair imagens do Amazon ECR, isso também ajuda a reduzir os tempos de pull da rede e reduz as alterações de transferência de dados quando o tráfego sai da VPC.

> **Important**
>
> Quando você usa Fargate, você deve autenticar em um registro de imagem privada usandorepositoryCredentials. Não é possível definir as variáveis de ambiente do agente de contêiner do Amazon ECSECS_ENGINE_AUTH_TYPEouECS_ENGINE_AUTH_DATAou modifique

`aecs.config`para tarefas hospedadas no Fargate. Para obter mais informações, consulte[Autenticação de registro privado para](#).

Função de instância de contêiner do Amazon EC2

O agente de contêiner do Amazon ECS é um contêiner que é executado em cada instância do Amazon EC2 em um cluster do Amazon ECS. Ele é inicializado fora do Amazon ECS usando `o``init`O comando disponível no sistema operacional. Consequentemente, não podem ser concedidas permissões por meio de uma função de tarefa. Em vez disso, as permissões devem ser atribuídas às instâncias do Amazon EC2 nas quais os agentes são executados. A lista de ações no exemplo`AmazonEC2ContainerServiceforEC2Role`precisam ser concedidas a`oecsInstanceRole`. Se você não fizer isso, suas instâncias não poderão ingressar no cluster.

```
{
    "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": "*"
        }
    ]
}
```

Nesta política, `oecrelogs`permitem que os contêineres que estão sendo executados em suas instâncias extraiam imagens do Amazon ECR e gravem logs no Amazon CloudWatch. `Oecs`permitem que o agente registre e cancele instâncias e se comunique com o plano de controle do Amazon ECS. Destes, `oecs:CreateCluster`ação é opcional.

Funções vinculadas ao serviço

Você pode usar a função vinculada a serviço do Amazon ECS para conceder permissão ao serviço do Amazon ECS para chamar outras APIs de serviço em seu nome. O Amazon ECS precisa das permissões para criar e excluir interfaces de rede, registrar e cancelar o registro de destinos com um grupo de destino. Ela também precisa das permissões necessárias para criar e excluir políticas de dimensionamento. Essas permissões são concedidas por meio da função vinculada ao serviço. Essa função é criada em seu nome na primeira vez que você usar o serviço.

> **Note**
>
> Se você excluir inadvertidamente a função vinculada ao serviço, poderá recriá-la. Para obter instruções, consulteCriar a função vinculada ao serviço.

Recommendations

Recomendamos fazer o seguinte ao configurar as funções e políticas do IAM da tarefa.

Bloquear o acesso aos metadados do Amazon EC2

Quando você executa suas tarefas em instâncias do Amazon EC2, recomendamos que você bloqueie o acesso aos metadados do Amazon EC2 para evitar que seus contêineres herdem a função atribuída a essas instâncias. Se seus aplicativos tiverem que chamar umAWSação da API, use funções do IAM para tarefas.

Para impedir a execução de tarefas nopontede acessar metadados do Amazon EC2, execute o seguinte comando ou atualize os dados do usuário da instância. Para obter mais instruções sobre como atualizar os dados do usuário de uma instância, consulte esteAWSArtigo de Support do. Para obter mais informações sobre o modo de ponte de definição de tarefa, consultemodo de rede de definição de tarefa.

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

Para que essa alteração persista após uma reinicialização, execute o seguinte comando específico para sua Imagem de máquina da 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 tarefas que usam`awsvpc`modo de rede, defina a variável de ambiente`ECS_AWSVPC_BLOCK_IMDS`para`true`no`/etc/ecs/ecs.config`file.

Você deve definir a propriedade`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST`Variável para`false`no arquivo de configuração do ecs-agent para impedir que os contêineres que estão sendo executados no`host`acessar os metadados do Amazon EC2.

Usar o**`awsvpc`**Modo de rede

Usar a rede`awsvpc`para restringir o fluxo de tráfego entre diferentes tarefas ou entre suas tarefas e outros serviços executados na Amazon VPC. Isso adiciona uma camada adicional de segurança. O`awsvpc`fornece isolamento de rede em nível de tarefa para tarefas executadas no Amazon EC2. É o modo padrão emAWS Fargate. é o único modo de rede que você pode usar para atribuir um grupo de segurança a tarefas.

Use o IAM Access Advisor para refinar funções

Recomendamos que você remova todas as ações que nunca foram usadas ou que não foram usadas há algum tempo. Isso impede que o acesso indesejado aconteça. Para fazer isso, revise os resultados produzidos pelo IAM Access Advisor e remova ações que nunca foram usadas ou que não foram usadas recentemente. Para isso, você pode seguir as seguintes etapas.

Execute o seguinte comando para gerar um relatório mostrando as últimas informações de acesso para a política referenciada:

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

Use a`JobId`que estava na saída para executar o seguinte comando. Depois disso, você pode visualizar os resultados do relatório.

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

Para obter mais informações, consulteIAM Access Advisor.

MonitorAWS CloudTrailPara atividades suspeitas

Você pode monitorarAWS CloudTrailpara qualquer atividade suspeita. MostAWSChamadas de API doAWS CloudTrailComo eventos. Eles são analisados porAWS CloudTrailInsights, e você será alertado sobre qualquer comportamento suspeito associado ao`write`Chamadas de API. Isso pode incluir um pico no volume da chamada. Esses alertas incluem informações como o horário em que a atividade incomum ocorreu e o ARN de identidade superior que contribuiu para as APIs.

Você pode identificar ações que são executadas por tarefas com uma função do IAM noAWS CloudTrailolhando para o evento`userIdentity`Propriedade. No exemplo a seguir, `aarn`inclui o nome da função assumida,`s3-write-go-bucket-role`, seguido pelo nome da tarefa,`7e9894e088ad416eb5cab92afExample`.

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

> **Note**
>
> Quando tarefas que assumem uma função são executadas em instâncias de contêiner do Amazon EC2, uma solicitação é registrada pelo agente de contêiner do Amazon ECS para o log de auditoria do agente localizado em um endereço no`/var/log/ecs/audit.log.YYYY-MM-DD-HH`format. Para obter mais informações, consulteLog de funções do IAMeRegistrar em log eventos do Insights para trilhas.

Segurança de rede

Segurança de rede é um tópico amplo que abrange vários subtópicos. Estes incluem criptografia em trânsito, segmentação e isolamento de rede, firewall, roteamento de tráfego e observabilidade.

Criptografia em trânsito

A criptografia do tráfego de rede impede que usuários não autorizados interceptem e leiam dados quando esses dados são transmitidos através de uma rede. Com o Amazon ECS, a criptografia de rede pode ser implementada de qualquer uma das seguintes formas.

- Com uma malha de serviço (TLS):

  comAWS App Mesh, você pode configurar conexões TLS entre os proxies Envoy implantados com endpoints de malha. Dois exemplos são nós virtuais e gateways virtuais. Os certificados TLS podem vir deAWS Certificate Manager(ACM). Ou, ele pode vir de sua própria autoridade de certificação privada.

  - Ativar Transport Layer Security (TLS)
  - Ativar criptografia de tráfego entre serviços noAWS App Meshusando certificados ACM ou certificados fornecidos pelo cliente
  - Demonstra a passo do TLS ACM
  - Demonstra passo a passo do arquivo TLS
  - Envoy
- Usar instâncias do Nitro:

  Por padrão, o tráfego é criptografado automaticamente entre os seguintes tipos de instância Nitro: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn e R5n. O tráfego não é criptografado quando é roteado por um gateway de trânsito, balanceador de carga ou intermediário similar.

  - Criptografia em trânsito
  - Novidades em contas de 2019
  - Esta conversa de Re:Inforce 2019
- Usando a SNI (Server Name Indication) com um Application Load Balancer:

  O Application Load Balancer (ALB) e o Network Load Balancer (NLB) suportam o Server Name Indication (SNI). Usando o SNI, você pode colocar vários aplicativos seguros atrás de um único ouvinte. Para isso, cada um tem seu próprio certificado TLS. Recomendamos que você provisione

certificados para o balanceador de carga usandoAWS Certificate Manager(ACM) e adicione-os à lista de certificados do ouvinte. OAWSO load balancer usa um algoritmo inteligente de seleção de certificado com SNI. Se o nome de host fornecido por um cliente corresponder a um único certificado na lista, o load balancer escolherá esse certificado. Se um nome de host fornecido por um cliente corresponder a vários certificados na lista, o load balancer selecionará um certificado que o cliente puder suportar. Exemplos incluem certificado autoassinado ou um certificado gerado por meio do ACM.

  - SNI com Application Load Balancer
  - SNI com Network Load Balancer
- Criptografia de ponta a ponta com certificados TLS:

  Isso envolve a implantação de um certificado TLS com a tarefa. Pode ser um certificado autoassinado ou um certificado de uma autoridade de certificação confiável. Você pode obter o certificado fazendo referência a um segredo para o certificado. Caso contrário, você pode optar por executar um contêiner que emite uma Solicitação de Assinatura de Certificado (CSR) para o ACM e, em seguida, monta o segredo resultante em um volume compartilhado.

  - Mantendo a segurança da camada de transporte até seus contêineres usando o Network Load Balancer com o Amazon ECS parte 1
  - Mantendo Transport Layer Security (TLS) até a parte 2 do seu contêiner: Como usar o AWS Private Certificate Authority

Redes de tarefas

As recomendações a seguir são consideradas como o Amazon ECS funciona. O Amazon ECS não usa uma rede de sobreposição. Em vez disso, as tarefas são configuradas para operar em diferentes modos de rede. Por exemplo, tarefas configuradas para usar`bridge`adquirem um endereço IP não roteável de uma rede Docker que é executada em cada host. Tarefas que estão configuradas para usar o`awsvpc`Adquira um endereço IP da sub-rede do host. Tarefas configuradas com`host`usam a interface de rede do host.`awsvpc`é o modo de rede preferido. Isso ocorre porque ele é o único modo que você pode usar para atribuir grupos de segurança a tarefas. É também o único modo disponível paraAWS Fargatetarefas no Amazon ECS.

Grupos de segurança para tarefas

Recomendamos que você configure suas tarefas para usar o`awsvpc`Modo de rede. Depois de configurar sua tarefa para usar esse modo, o agente do Amazon ECS automaticamente provisiona e

anexa uma interface de rede elástica (ENI) à tarefa. Quando o ENI é provisionado, a tarefa é inscrita em umAWSO grupo de segurança. O security group atua como um firewall virtual que você pode usar para controlar o tráfego de entrada e saída.

Malha de serviço e segurança de camada de transporte mútuo (MTLs)

Você pode usar uma malha de serviço, comoAWS App Meshpara controlar o tráfego de rede. Por padrão, um nó virtual só pode se comunicar com seus back-ends de serviço configurados, como os serviços virtuais com os quais o nó virtual se comunicará. Se um nó virtual precisar se comunicar com um serviço fora da malha, você pode usar o`ALLOW_ALL`filtro de saída ou criando um nó virtual dentro da malha para o serviço externo. Para obter mais informações, consulteInstruções de saída do Kubernetes Passo a passo.

O App Mesh também oferece a capacidade de usar Mutual Transport Layer Security (MTLs) onde o cliente e o servidor são autenticados mutuamente usando certificados. A comunicação subsequente entre cliente e servidor é então criptografada usando TLS. Ao exigir MTLs entre serviços em uma malha, você pode verificar se o tráfego vem de uma fonte confiável. Para obter mais informações, consulte os tópicos a seguir:

- Autenticação de
- Passo a passo do Serviço de Descoberta Secreta (SDS) MTLs
- Demonstra a passo do arquivo MTLS

AWS PrivateLink

AWS PrivateLinké uma tecnologia de rede que permite criar endpoints privados para diferentesAWS, incluindo o Amazon ECS. Os endpoints são necessários em ambientes restritos onde não há Internet Gateway (IGW) conectado à Amazon VPC e nenhuma rota alternativa para a Internet. O uso doAWS PrivateLinkgarante que as chamadas para o serviço Amazon ECS permaneçam dentro da Amazon VPC e não atravessem a Internet. Para obter instruções sobre como criarAWS PrivateLinkendpoints para o Amazon ECS e outros serviços relacionados, consulteInterface do Amazon ECS endpoints Amazon VPC.

> **Important**
>
> AWS Fargatetarefas não exigem umAWS PrivateLinkpara o Amazon ECS.

O Amazon ECR e o Amazon ECS suportam políticas de endpoint. Essas políticas permitem refinar o acesso às APIs de um serviço. Por exemplo, você pode criar uma política de endpoint para o Amazon ECR que só permite que as imagens sejam enviadas para registros em particularAWSContas. Uma política como essa pode ser usada para evitar que os dados sejam extraídos por imagens de contêiner e, ao mesmo tempo, permitir que os usuários enviem para registros autorizados do Amazon ECR. Para obter mais informações, consulteUsar políticas de VPC endpoint.

A política a seguir permite que todas asAWSdiretores em sua conta para executar todas as ações somente em seus repositórios do Amazon ECR:

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

Você pode aprimorar isso ainda mais definindo uma condição que usa o novo`PrincipalOrgID`Propriedade. Isso evita o envio e a extração de imagens por um principal do IAM que não faz parte doAWS Organizations. Para obter mais informações, consulteaws:PrincipalOrgID.

Recomendamos aplicar a mesma política para ambos os`com.amazonaws.region.ecr.dkr`O e a`com.amazonaws.region.ecr.api`Endpoints do .

Configurações de agente de contêiner do Amazon ECS

O arquivo de configuração do agente de contêiner do Amazon ECS inclui várias variáveis de ambiente relacionadas à segurança da rede.`ECS_AWSVPC_BLOCK_IMDS`e`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST`são usados para bloquear o acesso de uma tarefa aos metadados do Amazon EC2.`HTTP_PROXY`é usado para configurar o agente para rotear através de um proxy HTTP para se conectar à internet. Para obter instruções sobre como configurar o agente e o tempo de execução do Docker para rotear por meio de um proxy, consulteConfiguração de proxy HTTP.

> **Important**
>
> Essas configurações não estão disponíveis quando você usaAWS Fargate.

Recommendations

Recomendamos que você faça o seguinte ao configurar sua Amazon VPC, load balancers e rede.

Usar criptografia de rede, quando aplicável

Você deve usar a criptografia de rede, quando aplicável. Determinados programas de conformidade, como PCI DSS, exigem que você criptografar dados em trânsito se os dados contiverem dados do titular do cartão. Se sua carga de trabalho tiver requisitos semelhantes, configure a criptografia de rede.

Os navegadores modernos alertam os usuários quando se conectam a sites inseguros. Se o serviço for fornecido por um balanceador de carga voltado para o público, use TLS/SSL para criptografar o tráfego do navegador do cliente para o balanceador de carga e criptografar novamente para o back-end, se necessário.

Usar o`awsvpc`modo de rede e grupos de segurança quando você precisa controlar o tráfego entre tarefas ou entre tarefas e outros recursos de rede

Você deve usar`awsvpc` grupos de segurança quando você precisa controlar o tráfego entre tarefas e entre tarefas e outros recursos de rede. Se seu serviço por trás de um ALB, use grupos de segurança para permitir somente tráfego de entrada de outros recursos de rede usando o mesmo grupo de segurança que seu ALB. Se seu aplicativo estiver atrás de um NLB, configure o grupo de segurança da tarefa para permitir somente o tráfego de entrada do intervalo CIDR do Amazon VPC e os endereços IP estáticos atribuídos ao NLB.

Os grupos de segurança também devem ser usados para controlar o tráfego entre tarefas e outros recursos dentro da Amazon VPC, como bancos de dados do Amazon RDS.

Crie clusters em Amazon VPCs separadas quando o tráfego de rede precisar ser estritamente isolado

Você deve criar clusters em Amazon VPCs separadas quando o tráfego de rede precisar ser estritamente isolado. Evite executar cargas de trabalho que tenham requisitos de segurança

rigorosos em clusters com cargas de trabalho que não tenham de aderir a esses requisitos. Quando o isolamento rigoroso da rede for obrigatório, crie clusters em Amazon VPCs separadas e exponha serviços seletivamente a outras Amazon VPCs usando endpoints da Amazon VPC. Para obter mais informações, consulteVPC endpoints da Amazon endpoints.

ConfigureAWS PrivateLinkendpoints quando garantidos

Você deve configurar oAWS PrivateLinkendpoints quando justificado. Se sua política de segurança impedir que você anexe um Internet Gateway (IGW) às suas Amazon VPCs, configureAWS PrivateLinkendpoints para o Amazon ECS e outros serviços, como o Amazon ECR,AWS Secrets Managere o Amazon CloudWatch.

Use o Amazon VPC Flow Logs para analisar o tráfego de e para tarefas de execução longa

Você deve usar o Amazon VPC Flow Logs para analisar o tráfego de e para tarefas de execução longa. Tarefas que usam`awsvpc`modo de rede obter seu próprio ENI. Ao fazer isso, você pode monitorar o tráfego que vai de e para tarefas individuais usando o Amazon VPC Flow Logs. Uma atualização recente dos logs de fluxo do Amazon VPC (v3) enriquece os logs com metadados de tráfego, incluindo o ID da vpc, o ID da sub-rede e o ID da instância. Esses metadados podem ser usados para ajudar a restringir uma investigação. Para obter mais informações, consulteRegistros de fluxo da Amazon VPC.

> **Note**
>
> Devido à natureza temporária dos contêineres, os logs de fluxo nem sempre podem ser uma maneira eficaz de analisar padrões de tráfego entre diferentes contêineres ou contêineres e outros recursos de rede.

Gerenciamento de segredos do

Segredos, como chaves de API e credenciais de banco de dados, são freqüentemente usados por aplicativos para obter acesso a outros sistemas. Eles geralmente consistem em um nome de usuário e senha, um certificado ou uma chave de API. O acesso a esses segredos deve ser restrito a entidades específicas do IAM que estão usando o IAM e injetadas em contêineres em tempo de execução.

Segredos podem ser perfeitamente injetados em recipientes deAWS Secrets ManagerO Armazenamento de parâmetros do Amazon EC2 Systems Manager. Esses segredos podem ser referenciados em sua tarefa como qualquer um dos seguintes.

1. Eles são referenciados como variáveis de ambiente que usam `osecrets`parâmetro de definição de contêiner.
2. Eles são referenciados como`secretOptions`se sua plataforma de registro exigir autenticação. Para obter mais informações, consulteOpções de configuração de log.
3. Eles são referenciados como segredos puxados por imagens que usam `orepositoryCredentials`parâmetro de definição de contêiner se o registro do qual o contêiner está sendo extraído requer autenticação. Use esse método ao extrair imagens do Docker Hub. Para obter mais informações, consulteAutenticação de registro privado para.

Recommendations

Recomendamos que você faça o seguinte ao configurar o gerenciamento de segredos.

Usar oAWS Secrets ManagerO Armazenamento de parâmetros do Amazon EC2 Systems Manager para armazenar materiais secretos

Você deve armazenar com segurança chaves de API, credenciais de banco de dados e outros materiais secretos noAWS Secrets ManagerOu como um parâmetro criptografado no Armazenamento de parâmetros do Amazon EC2 Systems Manager. Esses serviços são semelhantes porque ambos são armazenamentos de chave-valor gerenciados que usamAWS KMSPara criptografar dados confidenciais.AWS Secrets Manager, no entanto, também inclui a capacidade de girar automaticamente segredos, gerar segredos aleatórios e compartilhar segredos entreAWSContas. Se você considerar esses recursos importantes, useAWS Secrets Managercaso contrário, use parâmetros criptografados.

> **Note**
>
> Tarefas que fazem referência a um segredo doAWS Secrets ManagerO Armazenamento de parâmetros do Amazon EC2 Systems Manager exige umFunção de execução de tarefascom uma política que concede ao Amazon ECS acesso ao segredo desejado e, se aplicável, aoAWS KMSchave usada para criptografar e descriptografar esse segredo.

> **Important**
>
> Segredos que são referenciados em tarefas não são girados automaticamente. Se o segredo for alterado, você deverá forçar uma nova implantação ou iniciar uma nova tarefa para recuperar o valor secreto mais recente. Para obter mais informações, consulte os tópicos a seguir:
>
> - AWS Secrets Manager: Injetando dados como variáveis de ambiente
> - Armazenamento de parâmetros do Amazon EC2 Systems Manager: Injetando dados como variáveis de ambiente

Recuperação de dados de um bucket criptografado do Amazon S3

Como o valor das variáveis de ambiente pode inadvertidamente vazar em logs e são revelados ao executar`docker inspect`, você deve armazenar segredos em um bucket criptografado do Amazon S3 e usar funções de tarefa para restringir o acesso a esses segredos. Quando você fizer isso, seu aplicativo deve ser gravado para ler o segredo do bucket do Amazon S3. Para obter instruções, consulteDefinir o comportamento padrão de criptografia do lado do servidor para buckets do Amazon S3.

Montar o segredo em um volume usando um contêiner sidecar

Como há um risco elevado de vazamento de dados com variáveis de ambiente, você deve executar um contêiner sidecar que lê seus segredos doAWS Secrets ManagerE escreva-os em um volume compartilhado. Esse contêiner pode ser executado e sair antes do contêiner do aplicativo usandoPedido de contêiner Amazon ECS. Quando você fizer isso, o contêiner do aplicativo é posteriormente montar o volume onde o segredo foi escrito. Como o método bucket do Amazon S3, seu aplicativo deve ser gravado para ler o segredo do volume compartilhado. Como o volume é escopo para a tarefa, o volume é excluído automaticamente após a tarefa ser interrompida. Para obter um exemplo de um contêiner de sidecar, consulte oaws-secret-sidecar-injetorProjeto do.

> **Note**
>
> No Amazon EC2, o volume no qual o segredo é gravado pode ser criptografado com umAWS KMSChave gerenciada pelo cliente. NoAWS Fargate, o armazenamento de volume é automaticamente criptografado usando uma chave gerenciada pelo serviço.

Recursos adicionais

- Passar segredos a contêineres em uma tarefa do Amazon ECS
- CâmaraO é um wrapper para armazenar segredos no Armazenamento de parâmetros do Amazon EC2 Systems Manager Manager

Compliance

Sua responsabilidade de conformidade ao usar o Amazon ECS é determinada pela confidencialidade dos seus dados e pelos objetivos de conformidade da sua empresa e pelas regulamentações e leis aplicáveis.

AWSFornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e conformidade: Esses guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em conformidade e segurança noAWS.
- Whitepaper sobre a arquitetura da HIPAA Security and Compliance: Este whitepaper descreve como as empresas podem usarAWSPara criar aplicativos compatíveis com HIPAA.
- AWSServiços no escopo pelo programa de conformidade: A lista contém aAWSServiços no escopo de programas de conformidade específicos. Para obter mais informações, consulteAWSProgramas de conformidade.

Padrões de segurança de dados do setor de cartões de pagamento (PCI DSS)

É importante que você entenda o fluxo completo de dados de titulares de cartão (CHD) dentro do ambiente ao aderir ao PCI DSS. O fluxo CHD determina a aplicabilidade do PCI DSS, define os limites e componentes de um ambiente de dados de titulares de cartão (CDE) e, portanto, o escopo de uma avaliação PCI DSS. A determinação precisa do escopo do PCI DSS é fundamental para definir a postura de segurança e, em última análise, uma avaliação bem-sucedida. Os clientes devem ter um procedimento para a determinação do escopo que garanta sua integridade e detecte alterações ou desvios do escopo.

A natureza temporária dos aplicativos em contêineres fornece complexidades adicionais ao auditar configurações. Como resultado, os clientes precisam manter um conhecimento de todos os

parâmetros de configuração de contêiner para garantir que os requisitos de conformidade sejam atendidos em todas as fases do ciclo de vida de um contêiner.

Para obter informações adicionais sobre como alcançar a conformidade com PCI DSS no Amazon ECS, consulte os whitepapers a seguir.

- Architecting on Amazon ECS para compatibilidade com PCI DSS
- Arquitetura para Escopo e Segmentação do PCI DSS em AWS

HIPAA (Lei de Portabilidade e Responsabilidade de Provedores de Health dos EUA).

O uso do Amazon ECS com cargas de trabalho que processam informações de saúde protegidas (PHI) não requer configuração adicional. O Amazon ECS atua como um serviço de orquestração que coordena o lançamento de contêineres no Amazon EC2. Ele não opera com ou sobre dados dentro da carga de trabalho que está sendo orquestrada. Consistente com as regulamentações da HIPAA e osAWSAdendo de associado de negócios, PHI deve ser criptografado em trânsito e em repouso quando acessado por contêineres iniciados com o Amazon ECS.

Vários mecanismos para criptografar em repouso estão disponíveis com cadaAWS, como Amazon S3, Amazon EBS eAWS KMS. Você pode implantar uma rede de sobreposição (como VNS3 ou Weave Net) para garantir a criptografia completa de PHI transferida entre contêineres ou para fornecer uma camada redundante de criptografia. O registro completo também deve ser ativado e todos os logs de contêiner devem ser direcionados para o Amazon CloudWatch. Para obter mais informações, consulteArchitecting for HIPAA Security and Compliance.

Recommendations

Você deve envolver os proprietários do programa de conformidade em sua empresa com antecedência e usar oAWSModelo de responsabilidade compartilhadapara identificar a propriedade do controle de conformidade para o sucesso com os programas de conformidade relevantes.

Registro em log e monitoramento

O registro e o monitoramento são importantes para manter a confiabilidade, a disponibilidade e o desempenho do Amazon ECS e do seuAWSSoluções do.AWSA fornece várias ferramentas para monitorar os recursos do Amazon ECS e responder a possíveis incidentes:

- Alarmes do Amazon CloudWatch
- Amazon CloudWatch Logs
- Amazon CloudWatch Events
- AWS CloudTrail Logs

Você pode configurar os contêineres nas tarefas para enviar informações de log ao Amazon CloudWatch Logs. Se você estiver usando aAWS FargatePara suas tarefas, você pode visualizar os logs de seus contêineres. Se você estiver usando o tipo de inicialização do Amazon EC2, poderá visualizar logs diferentes dos contêineres em um local conveniente. Isso também evita que os logs de contêiner ocupem espaço em disco nas instâncias do seu contêiner.

Para obter mais informações sobre o Amazon CloudWatch Logs, consulteMonitorar registros de instâncias do Amazon EC2noGuia do usuário do Amazon CloudWatch. Para obter instruções sobre como enviar logs de contêiner das suas tarefas para o Amazon CloudWatch Logs, consulteUsar `oawslogs`driver de log.

Registro de contêiner com bit fluente

AWSO fornece uma imagem do Fluent Bit com plug-ins para o Amazon CloudWatch Logs e o Amazon Kinesis Data Firehose. Esta imagem fornece a capacidade de rotear logs para destinos do Amazon CloudWatch e do Amazon Kinesis Data Firehose (que incluem o Amazon S3, o Amazon Elasticsearch Service e o Amazon Redshift). Recomendamos usar o Fluent Bit como seu roteador de log porque ele tem uma taxa de utilização de recursos mais baixa do que o Fluentd. Para obter mais informações, consulteRegistro do Amazon CloudWatch Logs para bit fluenteeAmazon Kinesis Data Firehose para bit fluente.

OAWSA imagem Fluent Bit está disponível em:

- Amazon ECR na Galeria pública do Amazon ECR
- Repositório do Amazon ECR(na maioria das regiões de alta disponibilidade)
- Hub do Docker

Veja a seguir a sintaxe a sintaxe a ser usada na CLI do Docker.

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

Por exemplo, você pode puxar oAWSpara imagem de bit fluente usando este comando da CLI do Docker:

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

Consulte também as seguintes publicações do blog para obter informações sobre o Fluent Bit e os recursos relacionados:

- Bit fluente para Amazon EKS em AWS Fargate
- Registro de contêiner centralizado com bit fluente
- Criando um agregador de soluções de log escalável comAWS Fargate, Fluentd e Amazon Kinesis Data Firehose

Roteamento de log personalizado - FireLens para Amazon ECS

Com o FireLens para Amazon ECS, você pode usar parâmetros de definição de tarefa para rotear logs para umAWSServiço do ouAWSDestino da Rede de Parceiros (APN) para armazenamento de logs e análises. O FireLens funciona com Fluentd e Fluent Bit. Nós fornecemos oAWSpara a imagem de bit fluente. Ou, você pode usar sua própria imagem Fluentd ou Fluent Bit.

Você deve considerar as seguintes condições e considerações ao usar o FireLens para o Amazon ECS:

- O FireLens para Amazon ECS é compatível com tarefas hospedadas emAWS Fargatee o Amazon EC2.
- FireLens para Amazon ECS é compatível com oAWS CloudFormationModelos do. Para obter mais informações, consulteAWS::ECS::TaskDefinition FirelensConfigurationnoAWS CloudFormationGuia do usuário.
- Para tarefas que usam o`bridge`Em modo de rede, os contêineres com a configuração do FireLens devem ser iniciados antes que um contêiner de aplicativo que dependa dele seja iniciado. Para controlar a ordem em que os contêineres começam, use as condições de dependência na definição de tarefa. Para obter mais informações, consulteDependência de contêiner.

Segurança do AWS Fargate

Recomendamos que você leve em consideração as seguintes melhores práticas ao usar oAWS Fargate.

Usar oAWS KMSpara criptografar o armazenamento efêmero

Você deve ter seu armazenamento efêmero criptografado peloAWS KMS. Para tarefas do Amazon ECS hospedadas noAWS FargateUsar a versão da plataforma`1.4.0`Ou posterior, cada tarefa do recebe 20 GB de armazenamento temporário. A quantidade de armazenamento não pode ser ajustada. Em caso de tarefas que foram iniciadas em 28 de maio de 2020 ou mais tarde, o armazenamento temporário é criptografado com um algoritmo de criptografia AES-256 usando uma chave de criptografia gerenciada peloAWS Fargate.

Exemplo: Iniciar uma tarefa do Amazon ECS noAWS FargatePlataforma versão 1.4.0 com criptografia de armazenamento temporário

O comando a seguir executará uma tarefa do Amazon ECS noAWS Fargateversão 1.4 da plataforma. Como essa tarefa é iniciada como parte do cluster do Amazon ECS, ela usa os 20 GB de armazenamento efêmero criptografado automaticamente.

```
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
```

Capacidade SYS_PTRACE para rastreamento de syscall do kernel

A configuração padrão dos recursos do Linux que são adicionados ou removidos do contêiner é fornecida pelo Docker. Para obter mais informações sobre os recursos disponíveis, consultePrivilégio de execução e recursos LinuxnoExecução do Docker dodocumentação.

Tarefas que são iniciadas noAWS Fargatesuportam apenas a adição de`SYS_PTRACE`Capacidade do kernel.

Consulte o vídeo tutorial abaixo que mostra como usar esse recurso através do SysdigFalcoProjeto do.

#ContainersFromTheCouch - Solução de problemas em seuAWS FargateTarefa com recurso SYS_PTRACE

O código discutido no vídeo anterior pode ser encontrado no GitHubAqui está.

Segurança de tarefas e contêineres

Você deve considerar a imagem do contêiner como a primeira linha de defesa contra um ataque. Uma imagem insegura e mal construída pode permitir que um invasor escape dos limites do contêiner e obtenha acesso ao host. Você deve fazer o seguinte para mitigar o risco de isso acontecer.

Recommendations

Recomendamos fazer o seguinte ao configurar suas tarefas e contêineres.

Crie imagens mínimas ou use imagens sem distrosidade

Comece removendo todos os binários estranhos da imagem do contêiner. Se você estiver usando uma imagem desconhecida do Docker Hub, inspecione a imagem para consultar o conteúdo de cada uma das camadas do contêiner. Você pode usar um aplicativo comoMergulhoPara fazer isso.

Como alternativa, você pode usardistrosoque incluem somente seu aplicativo e suas dependências de tempo de execução. Eles não contêm gerenciadores de pacotes ou shells. Imagens distrosas melhoram o “sinal para o ruído dos scanners e reduz a carga de estabelecer a proveniência apenas para o que você precisa”. Para obter mais informações, consulte a Documentação do GitHub nodistroso.

O Docker tem um mecanismo para criar imagens a partir de uma imagem reservada e mínima conhecida comoarranhão. Para obter mais informações, consulteCriando uma imagem pai simples usandoarranhãona documentação do Docker. Com langages como Go, você pode criar um binário vinculado estático e referenciá-lo em seu Dockerfile. O exemplo a seguir mostra como conseguir isso.

```
############################
# 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.
```

O exemplo anterior também é um exemplo de uma compilação de vários estágios. Esses tipos de compilações são atraentes do ponto de vista da segurança, pois você pode usá-los para minimizar o tamanho da imagem final enviada para o registro de contêiner. Imagens de contêiner desprovidas de ferramentas de construção e outros binários estranhos melhoram sua postura de segurança reduzindo a superfície de ataque da imagem. Para obter mais informações sobre compilações de vários estágios, consultecriação de compilações de vários estágios.

Analisar suas imagens em busca de vulnerabilidades

Semelhante aos seus homólogos de máquinas virtuais, as imagens de contêiner podem conter binários e bibliotecas de aplicativos com vulnerabilidades ou desenvolver vulnerabilidades ao longo do tempo. A melhor maneira de proteger contra exploits é digitalizando regularmente suas imagens com um scanner de imagens. As imagens armazenadas no Amazon ECR podem ser digitalizadas por push ou sob demanda (uma vez a cada 24 horas). O Amazon ECR atualmente usaClair, uma solução de varredura de imagens de código aberto. Depois que uma imagem é digitalizada, os resultados são registrados no stream de eventos do Amazon ECR no Amazon EventBridge. Você também pode ver os resultados de uma varredura no console do Amazon ECR ou chamando oDescribeImageScanDAPI. Imagens com um`HIGH`ou`CRITICAL`deve ser excluída ou reconstruída. Se uma imagem implantada desenvolver uma vulnerabilidade, ela deve ser substituída o mais rápido possível.

Docker Desktop Edge versão 2.3.6.0ou posterior podescanimagens locais. As varreduras são alimentadas porSnyk, um serviço de segurança de aplicativos. Quando as vulnerabilidades são descobertas, o Snyk identifica as camadas e dependências com a vulnerabilidade no Dockerfile. Ele também recomenda alternativas seguras, como usar uma imagem base mais fina com menos vulnerabilidades ou atualizar um pacote específico para uma versão mais recente. Usando a verificação do Docker, os desenvolvedores podem resolver possíveis problemas de segurança antes de enviar suas imagens para o registro.

- Automatizar a conformidade de imagens usando o Amazon ECR eAWS Security Hubexplica como exibir informações de vulnerabilidade do Amazon ECR noAWS Security Hube automatize a correção bloqueando o acesso a imagens vulneráveis.

Remover permissões especiais de suas imagens

Os sinalizadores de direitos de acesso`setuid`e`setgid`permitem executar um executável com as permissões do proprietário ou grupo do executável. Remova todos os binários com esses direitos de acesso da sua imagem, pois esses binários podem ser usados para escalar privilégios. Considere remover todos os shells e utilitários com`nc`e`curl`que podem ser usados para fins maliciosos. Você pode encontrar os arquivos com`setuid`e`setgid`Usando o comando a seguir.

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

Para remover essas permissões especiais desses arquivos, adicione a seguinte diretiva à imagem do contêiner.

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

Criar um conjunto de imagens selecionadas

Em vez de permitir que os desenvolvedores criem suas próprias imagens, crie um conjunto de imagens verificadas para as diferentes pilhas de aplicativos em sua organização. Ao fazer isso, os desenvolvedores podem renunciar a aprender a compor Dockerfiles e concentrar-se na escrita de código. À medida que as alterações são mescladas na sua base de código, um pipeline CI/CD pode compilar automaticamente o ativo e, em seguida, armazená-lo em um repositório de artefatos. E, por último, copie o artefato para a imagem apropriada antes de enviá-lo para um registro do

Docker, como o Amazon ECR. No mínimo, você deve criar um conjunto de imagens de base que os desenvolvedores podem criar seus próprios Dockerfiles. Evite extrair imagens do Docker Hub. Você nem sempre sabe o que está na imagem e cerca de um quinto das 1000 imagens principais têm vulnerabilidades. Uma lista dessas imagens e suas vulnerabilidades pode ser encontrada emhttps://vulnerablecontainers.org/.

Analisar vulnerabilidades em pacotes de aplicativos e bibliotecas

O uso de bibliotecas de código aberto agora é comum. Assim como acontece com sistemas operacionais e pacotes de SO, essas bibliotecas podem ter vulnerabilidades. Como parte do ciclo de vida de desenvolvimento, essas bibliotecas devem ser verificadas e atualizadas quando forem encontradas vulnerabilidades críticas.

O Docker Desktop executa verificações locais usando o Snyk. Ele também pode ser usado para encontrar vulnerabilidades e possíveis problemas de licenciamento em bibliotecas de código aberto. Ele pode ser integrado diretamente aos fluxos de trabalho do desenvolvedor, dando-lhe a capacidade de mitigar os riscos colocados pelas bibliotecas de código aberto. Para obter mais informações, consulte os tópicos a seguir:

- Ferramentas de segurança de aplicativos de código abertoinclui uma lista de ferramentas para detectar vulnerabilidades em aplicativos.
- Cheatsheet de verificação do Docker

Realizar análise de código estático

Você deve executar análise de código estático antes de criar uma imagem de contêiner. Ele é executado em seu código-fonte e é usado para identificar erros de codificação e código que podem ser explorados por um ator mal-intencionado, como injeções de falhas. SonarQubeé uma opção popular para testes de segurança de aplicativos estáticos (SAST), com suporte para uma variedade de linguagens de programação diferentes.

Executar contêineres como um usuário não-root

Você deve executar contêineres como um usuário que não seja raiz. Por padrão, os contêineres são executados como`root`, a menos que o`USER`está incluída no seu Dockerfile. Os recursos padrão do Linux atribuídos pelo Docker restringem as ações que podem ser executadas como`root`, mas apenas marginalmente. Por exemplo, um contêiner sendo executado como`root`ainda não tem permissão para acessar dispositivos.

Como parte do pipeline CI/CD, você deve lint Dockerfiles para procurar o`USER`diretiva e falhar na compilação se ele estiver ausente. Para obter mais informações, consulte os tópicos a seguir:

- Dockerfile-linté uma ferramenta de código aberto da RedHat que pode ser usada para verificar se o arquivo está em conformidade com as melhores práticas.
- Hadolinté outra ferramenta para criar imagens do Docker que estejam em conformidade com as práticas recomendadas.

Usar um sistema de arquivos raiz somente leitura

Você deve usar um sistema de arquivos raiz somente leitura. O sistema de arquivos raiz de um contêiner pode ser gravado por padrão. Quando você configura um contêiner com um`R0`(somente leitura) sistema de arquivos raiz que força você a definir explicitamente onde os dados podem ser persistidos. Isso reduz a superfície de ataque porque o sistema de arquivos do contêiner não pode ser gravado a menos que as permissões sejam especificamente concedidas.

> **Note**
>
> Ter um sistema de arquivos raiz somente leitura pode causar problemas com determinados pacotes do sistema operacional que esperam ser capazes de gravar no sistema de arquivos. Se você estiver planejando usar sistemas de arquivos raiz somente leitura, teste cuidadosamente com antecedência.

Configurar tarefas com limites de CPU e memória (Amazon EC2)

Você deve configurar tarefas com limites de CPU e memória para minimizar o risco a seguir. Os limites de recursos de uma tarefa definem um limite superior para a quantidade de CPU e memória que pode ser reservada por todos os contêineres de uma tarefa. Se nenhum limite for definido, as tarefas terão acesso à CPU e à memória do host. Isso pode causar problemas em que as tarefas implantadas em um host compartilhado podem deixar outras tarefas de recursos do sistema.

> **Note**
>
> Amazon ECS emAWS Fargateexigem que você especifique limites de CPU e memória porque ele usa esses valores para fins de cobrança. Uma tarefa monopolizando todos os recursos do sistema não é um problema para o Amazon ECS Fargate porque cada tarefa é executada em sua própria instância dedicada. Se você não especificar um limite de

memória, o Amazon ECS aloca um mínimo de 4 MB para cada contêiner. Da mesma forma, se nenhum limite de CPU for definido para a tarefa, o agente de contêiner do Amazon ECS atribuirá um mínimo de 2 CPUs.

Usar tags imutáveis com o Amazon ECR

Com o Amazon ECR, você pode e deve usar configurar imagens com tags imutáveis. Isso evita enviar uma versão alterada ou atualizada de uma imagem para o repositório de imagens com uma tag idêntica. Isso protege contra um invasor empurrar uma versão comprometida de uma imagem sobre sua imagem com a mesma tag. Ao usar tags imutáveis, você efetivamente se força a empurrar uma nova imagem com uma tag diferente para cada alteração.

Evite executar contêineres como privilegiados (Amazon EC2)

Você deve evitar a execução de contêineres como privilegiados. Em segundo plano, contêineres são executados como`privileged`são executados com privilégios estendidos no host. Isso significa que o contêiner herda todos os recursos do Linux atribuídos a`root`no host. Seu uso deve ser severamente restrito ou proibido. Recomendamos definir a variável de ambiente do agente de contêiner do Amazon ECS`ECS_DISABLE_PRIVILEGED`para`true`para impedir que os contêineres sejam executados como`privileged`em hosts específicos se`privileged`não é necessário. Como alternativa, você pode usarAWS Lambdapara verificar suas definições de tarefa para o uso do`privileged`parâmetro .

> **Note**
>
> Executando um contêiner como`privileged`Não é compatível no Amazon ECS emAWS Fargate.

Remova recursos desnecessários do Linux do contêiner

Veja a seguir uma lista dos recursos padrão do Linux atribuídos aos contêineres do Docker. Para obter mais informações sobre cada recurso, consulteVisão geral dos recursos do 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
```

Se um contêiner não exigir todas as capacidades kernal do Docker listadas acima, considere soltá-las do contêiner. Para obter mais informações sobre cada recurso kernal do Docker, consulteKernalCapabilities. Você pode descobrir quais recursos estão em uso fazendo o seguinte:

- Instale o pacote do SOlibcap-nge execute `apscap`para listar os recursos que cada processo está usando.
- Você também pode usarcapshpara decifrar quais recursos um processo está usando.
- ConsulteRecursos do LinuxPara obter mais informações.

Usar uma chave gerenciada pelo cliente (CMK) para criptografar imagens enviadas para o Amazon ECR

Você deve usar uma chave gerenciada pelo cliente (CMK) para incrupar imagens enviadas para o Amazon ECR. As imagens enviadas para o Amazon ECR são automaticamente criptografadas em repouso com umAWS Key Management Service(AWS KMS) gerenciada pela. Se você preferir usar sua própria chave, o Amazon ECR agora suportaAWS KMScriptografia com chaves gerenciadas pelo cliente (CMK). Antes de habilitar a criptografia do lado do servidor com um CMK, revise as Considerações listadas na documentação sobreCriptografia em repouso.

Segurança de execução do

A segurança em tempo de execução oferece proteção ativa aos contêineres durante a execução. A ideia é detectar e evitar que atividades maliciosas ocorram seus contêineres.

Com computação segura (`seccomp`), você pode impedir que um aplicativo em contêiner faça determinados syscalls para o kernel do sistema operacional host subjacente. Embora o sistema operacional Linux tenha algumas centenas de chamadas de sistema, a maioria delas não é necessária para executar contêineres. Ao restringir quais syscalls podem ser feitos por um contêiner, você pode efetivamente diminuir a superfície de ataque do aplicativo.

Para começar a usar o seccomp, você pode usar`strace`para gerar um rastreamento de pilha para ver quais chamadas de sistema seu aplicativo está fazendo. Você pode usar uma ferramenta como`syscall2seccomp`para criar um perfil seccomp a partir dos dados coletados do rastreamento de pilha. Para obter mais informações, consultestraceesyscall2seccomp.

Ao contrário do módulo de segurança SELinux, o seccomp não pode isolar contêineres uns dos outros. No entanto, ele protege o kernel host de syscalls não autorizados. Ele funciona interceptando syscalls e permitindo apenas aqueles que foram permitidos listados para passar. O Docker tem umpadrãoseccomp que é adequado para a maioria das cargas de trabalho de uso geral.

> **Note**
>
> Também é possível criar perfis próprios para itens que exigem privilégios adicionais.

AppArmor é um módulo de segurança Linux semelhante ao seccomp, mas restringe as capacidades de um contêiner, incluindo o acesso a partes do sistema de arquivos. Ele pode ser executado em`enforcement`ou`complain`Modo. Como criar perfis do AppArmor pode ser um desafio, recomendamos que você use uma ferramenta como obanir. Para obter mais informações sobre o AppArmor, consulte oAppArmor.

> **Important**
>
> AppArmor só está disponível para distribuições Ubuntu e Debian do Linux.

Recommendations

Recomendamos que você execute as ações seguintes ao configurar a segurança do tempo de execução.

Usar uma solução de terceiros para defesa em tempo de execução

Usar uma solução de terceiros para defesa em tempo de execução. Se você estiver familiarizado com o funcionamento da segurança do Linux, crie e gerencie perfis seccomp e AppArmor. Ambos são projetos de código aberto. Caso contrário, considere usar um serviço de terceiros diferente. A maioria usa aprendizado de máquina para bloquear ou alertar sobre atividades suspeitas. Para obter uma lista de soluções de terceiros disponíveis, consulteAWS MarketplaceContêineres.

Adicionar ou remover recursos do Linux usando políticas seccomp

Use seccomp para ter maior controle sobre os recursos do Linux e para evitar erros de verificação do syscall. Seccomp funciona como um filtro syscall que revoga a permissão para executar determinados syscalls ou para usar agruments específicos.

AWSParceiros

Você pode usar qualquer um dosAWSParticipe de produtos para adicionar segurança e recursos adicionais às suas cargas de trabalho do Amazon ECS. Para obter mais informações, consulteParceiros do Amazon ECS.

Aqua Security

Você pode usarAqua Securitypara proteger seus aplicativos nativos da nuvem do desenvolvimento à produção. A plataforma de segurança nativa do Aqua Cloud integra-se aos recursos nativos da nuvem e às ferramentas de orquestração para fornecer segurança transparente e automatizada. Ele pode evitar atividades suspeitas e ataques em tempo real, além de ajudar a impor políticas e simplificar a conformidade normativa.

Palo Alto Networks

Palo Alto Networksfornece segurança e proteção para seus hosts, contêineres e infraestrutura sem servidor na nuvem e durante todo o ciclo de vida de desenvolvimento e software.

O Twistlock é fornecido pela Palo Alto Networks e pode ser integrado ao Amazon ECS FireLens. Com ele, você tem acesso a logs e incidentes de segurança de alta fidelidade que são perfeitamente agregados em váriosAWSServiços da . Eles incluem Amazon CloudWatch, Amazon Athena e Amazon Kinesis. O Twistlock protege cargas de trabalho implantadas noAWSServiços de contêiner.

Sysdig

Você pode usarSysdigpara executar cargas de trabalho nativas de nuvem seguras e compatíveis em cenários de produção. A plataforma Sysdig Secure DevOps possui recursos de segurança e conformidade incorporados para proteger suas cargas de trabalho nativas da nuvem, além de oferecer escalabilidade, desempenho e personalização de nível empresarial.

Histórico do documento do Guia de melhores práticas do Amazon ECS do

A tabela a seguir descreve as versões da documentação do Amazon ECS Best Practices Guide do.

update-history-change	update-history-description	update-history-date
Práticas recomendadas de segurança	Adição de práticas recomendadas para gerenciamento de segurança para cargas de trabalho do Amazon ECS.	26 de maio de 2021
Práticas recomendadas de dimensionamento automático e gerenciamento de capacidade	Adição de práticas recomendadas para auto scaling e gerenciamento de capacidade para cargas de trabalho do Amazon ECS.	14 de maio de 2021
Melhores práticas de armazenamento persistente	Adição de práticas recomendadas para armazenamento persistente para cargas de trabalho do Amazon ECS.	7 de maio de 2021
Melhores práticas de rede	Adição de práticas recomendadas para gerenciamento de rede para cargas de trabalho do Amazon ECS.	6 de abril de 2021
Versão inicial	Versão inicial do Guia de melhores práticas do Amazon ECS do	6 de abril de 2021

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.