Amazon ECS

Guia do usuário do AWS Fargate

Versão da API 2014-11-13

Amazon ECS: Guia do usuário do AWS Fargate

Table of Contents

O que é Amazon ECS? ... 1
- Recursos do Amazon ECS ... 1
  - Contêineres e imagens ... 2
  - Definições de tarefas ... 3
  - Tarefas e programação ... 4
  - Clusters ... 4
- Conceitos básicos do Amazon ECS ... 4
- Serviços relacionados ... 4
- Como acessar a Amazon ECS ... 5

Configurar ... 7
- Cadastre-se na AWS ... 7
- Criar um usuário do IAM ... 7
- Criar uma função do IAM ... 9
- Criar uma nuvem privada virtual ... 9
- Instalar a AWS CLI ... 10

Noções básicas do Docker para Amazon ECS ... 11
- Instalação do Docker ... 11
- Criar uma imagem do Docker ... 12
- (Opcional) Enviar sua imagem por push para o Amazon Elastic Container Registry ... 14
- Limpar (opcional) ... 15
- Próximas etapas ... 15

Conceitos básicos do Amazon ECS ... 16
- Conceitos básicos do Amazon ECS que usa Fargate ... 16
  - Pré-requisitos ... 16
  - Etapa 1: Criar uma definição de tarefa ... 17
  - Etapa 2: Configurar o serviço ... 17
  - Etapa 3: Configurar o cluster ... 18
  - Etapa 4: Revisão ... 18
  - Etapa 5: (Opcional) Exibir o serviço ... 18
  - Etapa 6: limpeza ... 19

Versões da plataforma ... 20
- Considerações da versão da plataforma ... 20
- Versões de plataforma disponíveis ... 20

Clusters ... 22
- Criação de um cluster ... 22
- Atualizar configurações de cluster ... 23
- Exclusão de um cluster ... 23

Definições de Tarefas ... 25
- Considerações sobre definição de tarefas ... 25
  - Modo de rede ... 26
  - CPU e memória da tarefa ... 26
  - Registro ... 27
  - Função do IAM da execução de tarefas do Amazon ECS ... 27
  - Exemplo de definição de tarefa ... 27
  - Armazenamento de tarefas ... 28
- Arquitetura do aplicativo ... 29
  - Uso do tipo de inicialização Fargate ... 29
- Como criar uma definição de tarefa ... 29
  - Modelo de definição de tarefa ... 31
- Parâmetros de definição de tarefa ... 34
  - Família ... 35
  - Função de execução de tarefas ... 35
  - Modo de rede ... 35
  - Definições de contêiner ... 35

Volumes 54
Tipos de inicialização 54
Tamanho da tarefa 55
Configuração do proxy 56
Outros parâmetros de definição de tarefa 58
Tipos de inicialização 59
Tipo de inicialização Fargate 59
Tipo de inicialização EC2 60
Como usar volumes de dados em tarefas 61
Redes de tarefas 62
Considerações sobre redes de tarefas 63
Habilitar a rede de tarefas 63
Como usar o driver de log awslogs 64
Ativação do driver de logs awslogs para seus contêineres 64
Criar um grupo de logs 64
Opções do driver de log awslogs disponíveis 65
Como especificar uma configuração de log na definição da tarefa 66
Como visualizar logs de contêiner awslogs no CloudWatch Logs 68
Rotear logs personalizados 70
Autenticação de registro privado para tarefas 71
Permissões do IAM necessárias para a autenticação de registro privado 71
Habilitar a autenticação de registro privado 72
Especificação de dados confidenciais 73
Considerações para especificar dados confidenciais 74
Injetar dados confidenciais como uma variável de ambiente 74
Injetar dados confidenciais em uma configuração de log 75
Permissões necessárias do IAM para segredos do Amazon ECS 76
Criação de um parâmetro do Parameter Store do AWS Systems Manager 76
Criar uma definição de tarefa que faz referência a um segredo 77
Definições de tarefa de exemplo 79
Exemplo: servidor da Web 79
Exemplo: driver de log do `awslogs` 80
Exemplo: driver de log do `splunk` 80
Exemplo: driver de log `fluentd` 81
Exemplo: driver de log `gelf` 81
Exemplo: dependência de contêiner 82
Como atualizar uma definição de tarefa 83
Como cancelar o registro das definições de tarefa 83
Configurações da conta 85
Nomes de recursos da Amazon (ARNs) e IDs 86
Visualizar configurações da conta 87
Como modificar configurações da conta 88
Como programar tarefas 90
Tarefas em execução 91
Executar uma tarefa usando o tipo de inicialização Fargate 91
Tarefas programadas (`cron`) 93
Desativação da tarefa 96
Como trabalhar com tarefas agendadas para desativação 96
Reciclagem de tarefas Fargate 97
Serviços 99
Conceitos do programador de serviço 99
Daemon 100
Réplica 100
Conceitos de serviços adicionais 100
Parâmetros de definição de serviço 101
Tipos de implantação 107
Atualização contínua 108

Implantação azul/verde com o CodeDeploy ... 108
Implantação externa ... 111
Balanceamento de carga do serviço ... 115
Considerações sobre balanceamento de carga de serviço ... 116
Tipos de load balancer ... 117
Como criar um balanceador de carga ... 119
Registro de vários grupos de destino com um serviço ... 126
Serviço Auto Scaling ... 129
Serviço Auto Scaling Permissões obrigatórias do IAM ... 129
Políticas de escalabilidade de rastreamento de destino ... 130
Políticas de escalabilidade em etapas ... 135
Descoberta de serviço ... 137
Conceitos do Descoberta de serviço ... 137
Considerações sobre o Descoberta de serviço ... 138
Experiência de console do Amazon ECS ... 139
Definição de preço do Descoberta de serviço ... 139
Tutorial: como criar um serviço usando Descoberta de serviço ... 139
Criar um serviço ... 148
Etapa 1: configuração de parâmetros básicos de serviço ... 149
Etapa 2: configuração de uma rede ... 151
Etapa 3: (opcional) configurar o serviço para usar um load balancer ... 151
Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço ... 156
Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling ... 157
Etapa 6: consultar e criar o serviço ... 160
Atualizar um serviço ... 160
Excluir um serviço ... 163
Lógica de controle de serviço ... 164
Recursos e tags ... 165
Marcação dos seus recursos ... 165
Conceitos básicos de tags ... 165
Marcação dos seus recursos ... 166
Restrições de tag ... 167
Marcação dos seus recursos para faturamento ... 167
Trabalho com tags usando o console ... 167
Trabalho com tags usando a CLI ou a API ... 168
Relatórios de uso ... 170
Monitoramento ... 171
Ferramentas de monitoramento ... 171
Ferramentas automatizadas ... 172
Ferramentas manuais ... 172
Métricas do CloudWatch ... 173
Como habilitar as métricas do CloudWatch ... 173
Métricas e dimensões disponíveis ... 173
Utilização de serviço ... 176
Contagem de tarefas `RUNNING` de serviço ... 177
Visualizar métricas do Amazon ECS ... 178
Eventos do CloudWatch ... 179
Eventos do Amazon ECS ... 179
Como processar eventos ... 181
Tutorial: como escutar Eventos do CloudWatch do Amazon ECS ... 183
Tutorial: como enviar alertas do Amazon Simple Notification Service para eventos de tarefa parada ... 185
CloudWatch Container Insights ... 186
Como trabalhar com clusters habilitados para Container Insights ... 187
Registro em log de chamadas à API do Amazon ECS com o AWS CloudTrail ... 188
Informações sobre o Amazon ECS no CloudTrail ... 189
Noções básicas das entradas dos arquivos de log do Amazon ECS ... 189

Segurança 191
Identity and Access Management 191
Público 192
Autenticação com identidades 192
Gerenciamento do acesso usando políticas 194
Como o Amazon Elastic Container Service funciona com o IAM 196
Exemplos de políticas baseadas em identidade 200
Permissões no nível do recurso compatíveis 209
Usar funções vinculadas a serviço 216
Políticas gerenciadas e relacionamentos de confiança 221
Função do IAM da execução de tarefas do Amazon ECS 228
Função do IAM programador de serviço do Amazon ECS 232
Função do IAM CodeDeploy do Amazon ECS 234
Função do IAM Serviço Auto Scaling do Amazon ECS 237
Função do IAM Eventos do CloudWatch 239
Amazon ECS Task Role (Função da tarefa do Amazon ECS) 241
Funções do IAM para tarefas 242
Solução de problemas 245
Usar a CLI do Amazon ECS 248
Como instalar a CLI do Amazon ECS 248
Etapa 1: faça download da CLI do Amazon ECS 248
Etapa 2: (opcional) verificar a CLI do Amazon ECS 248
Etapa 3: aplicar permissões de execução ao binário 253
Etapa 4: concluir a instalação 253
Como configurar a CLI do Amazon ECS 254
Perfis 254
Configurações de cluster 254
Ordem de precedência 255
Migração de arquivos de configuração 255
Migração de arquivos de configuração mais antigos para o formato v1.0.0+ 256
Tutorial: Como criar um cluster com uma tarefa Fargate usando a CLI do Amazon ECS 256
Pré-requisitos 256
Etapa 1: criar a função do IAM para execução de tarefas 257
Etapa 2: configurar a CLI do Amazon ECS 257
Etapa 3: Criar um cluster e configurar o grupo de segurança 258
Etapa 4: Criar um arquivo de composição 258
Etapa 5: Implantar o arquivo de composição em um cluster 259
Etapa 6: Visualizar os contêineres em execução em um cluster 259
Etapa 7: Visualizar os logs do contêiner 260
Etapa 8: Dimensionar as tarefas no cluster 260
Etapa 9: (Opcional) Visualizar seu aplicativo web 260
Etapa 10: Limpeza 261
Tutorial: Criação de um serviço do Amazon ECS que usa o Descoberta de serviço usando a CLI do Amazon ECS 261
Pré-requisitos 261
Configurar a CLI do Amazon ECS 261
Criar um serviço do Amazon ECS configurado para usar o Descoberta de serviço 262
Usar o AWS CLI 265
Tutorial: criar um cluster com uma tarefa do Fargate usando a AWS CLI 265
Pré-requisitos 265
Etapa 1: (Opcional) Criar um cluster 266
Etapa 2: Registrar uma definição de tarefa 266
Etapa 3: Listar definições de tarefa 268
Etapa 4: Criar um serviço 268
Etapa 5: Listar serviços 269
Etapa 6: Descrever o serviço em execução 270
Endpoint de metadados de tarefas 272

Habilitação de metadados de tarefas ... 272
Caminhos do endpoint de metadados de tarefas ... 272
Resposta do JSON de metadados de tarefas ... 273
Exemplo de resposta de metadados de tarefas ... 274
Service Limits ... 277
App Mesh e Amazon ECS ... 279
Pré-requisitos ... 279
Etapa 1: criar sua malha de serviços ... 279
Etapa 2: criar seus nós virtuais ... 279
Etapa 3: criar seus roteadores virtuais ... 281
Etapa 4: criar suas rotas ... 281
Etapa 5: criar seus serviços virtuais ... 282
Update Your Microservice Task Definitions ... 283
Proxy Configuration ... 283
Application Container Envoy Dependency ... 284
Envoy Container Definition ... 284
Credentials ... 285
Update an Existing Task Definition ... 285
Example Task Definitions ... 286
Tutoriais ... 289
Tutorial: Creating a VPC ... 289
Step 1: Create an Elastic IP Address for Your NAT Gateway ... 289
Step 2: Run the VPC Wizard ... 289
Step 3: Create Additional Subnets ... 290
Next Steps ... 290
Tutorial: Especificação de dados confidenciais usando segredos do Secrets Manager ... 291
Pré-requisitos ... 291
Etapa 1: criar um segredo do Secrets Manager ... 291
Etapa 2: atualizar a função do IAM de execução de tarefa ... 292
Etapa 3: criar uma definição de tarefa do Amazon ECS ... 293
Etapa 4: criar um cluster do Amazon ECS ... 294
Etapa 5: executar uma tarefa do Amazon ECS ... 295
Etapa 6: verificar ... 295
Etapa 7: limpeza ... 296
Tutorial: como criar um serviço usando uma implantação azul/verde ... 297
Pré-requisitos ... 297
Etapa 1: criar um Balanceador de carga de aplicações ... 297
Etapa 2: criar um cluster do Amazon ECS ... 298
Etapa 3: registrar uma definição de tarefa ... 298
Etapa 4: criar um serviço do Amazon ECS ... 299
Etapa 5: criar os recursos do AWS CodeDeploy ... 300
Etapa 5: criar e monitorar uma implantação do CodeDeploy ... 301
Etapa 6: limpeza ... 304
Tutorial: implantação contínua com o CodePipeline ... 305
Pré-requisitos ... 305
Etapa 1: Adicionar um arquivo de especificação de compilação ao repositório de origem ... 305
Etapa 2: Criar uma pipeline de implantação contínua ... 307
Etapa 3: Adicionar permissões do Amazon ECR para a função do CodeBuild ... 308
Etapa 4: Testar o pipeline ... 308
Solução de problemas ... 310
Solução de problemas de inicialização do assistente da primeira execução ... 310
Como verificar se há erros em tarefas interrompidas ... 310
Mensagens de evento de serviço ... 312
Mensagens de evento de serviço ... 313
Valor de memória ou CPU inválido especificado ... 314
Erro "Não foi possível obter a imagem do contêiner" ... 314
Como solucionar problemas de load balancers de serviço ... 316

Histórico do documento 318
AWS Glossary 327

O que é Amazon Elastic Container Service?

O Amazon Elastic Container Service (Amazon ECS) é um serviço de gerenciamento de contêineres altamente dimensionável e rápido que facilita a execução, a interrupção e o gerenciamento de contêineres do Docker em um cluster. Você pode hospedar seu cluster em uma infraestrutura sem servidor gerenciada pelo Amazon ECS ao iniciar seus serviços ou tarefas usando o tipo de inicialização Fargate. Para obter mais controle, você pode hospedar suas tarefas em um cluster de instâncias do Amazon Elastic Compute Cloud (Amazon EC2) gerenciado usando o tipo de inicialização EC2. Para obter mais informações sobre tipos de inicialização, consulte Tipos de inicialização Amazon ECS (p. 59).

O Amazon ECS permite iniciar e parar aplicativos baseados em contêiner com simples chamadas à API, permite que você obtenha o estado de seu cluster em um serviço centralizado e fornece acesso a muitos dos recursos do Amazon EC2.

Você pode usar o Amazon ECS para programar a colocação de contêineres no cluster com base nas necessidades de recursos, nas políticas de isolamento e nos requisitos de disponibilidade. O Amazon ECS elimina a necessidade de operar os próprios sistemas de gerenciamento de cluster e de configuração ou de se preocupar com a escalabilidade da infraestrutura de gerenciamento.

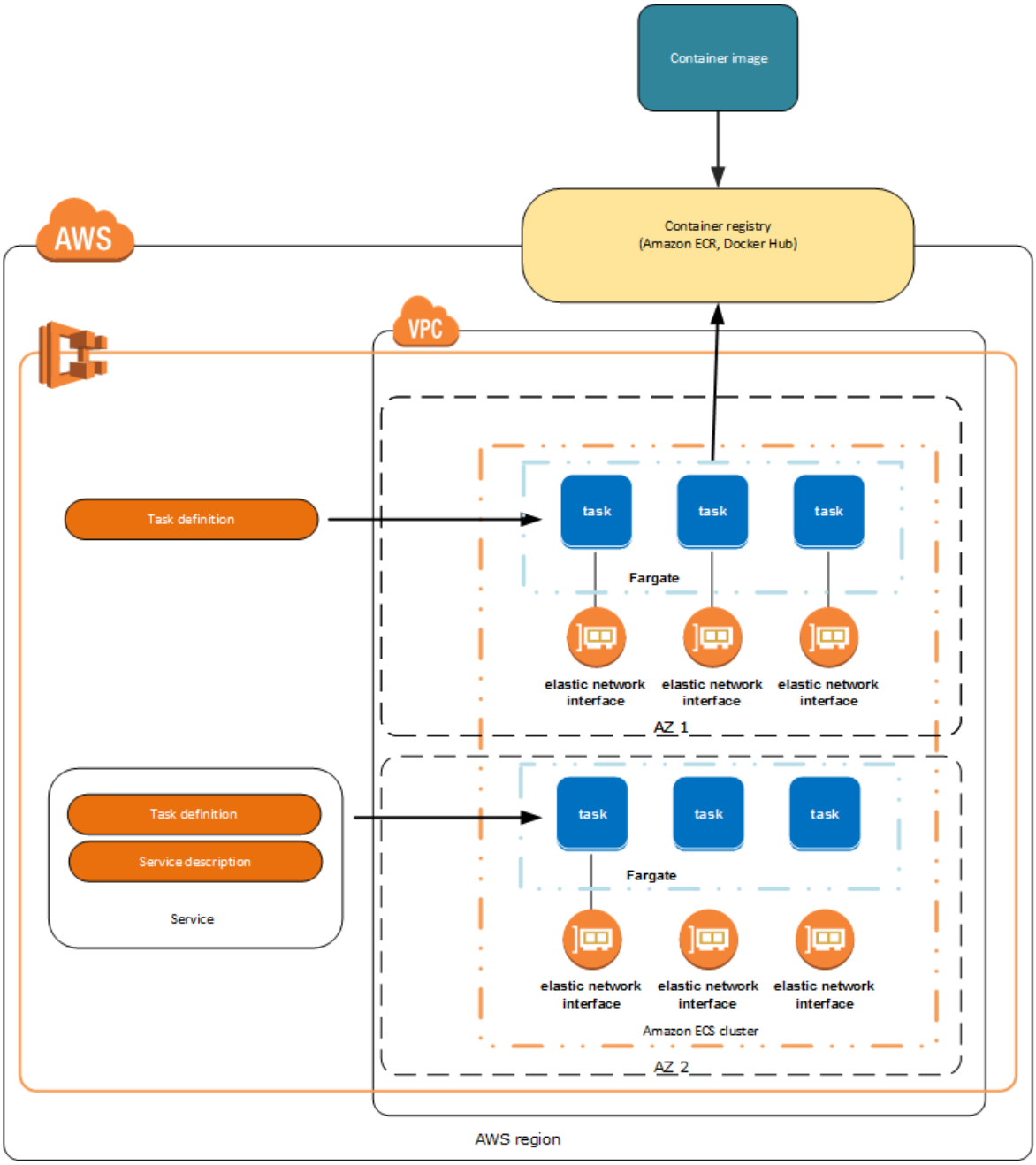
O Amazon ECS pode ser usado para criar uma implantação consistente, criar experiência, gerenciar e escalar workloads ETL (Extract-Transform-Load, Extração-Transformação-Carga) e criar arquiteturas de aplicativo sofisticadas em um modelo de microsserviços. Para obter informações sobre casos de uso e cenários do Amazon ECS, consulte Casos de uso de contêiner.

O AWS Elastic Beanstalk também pode ser usado para desenvolver, testar e implantar rapidamente contêineres do Docker em conjunto com outros componentes de sua infraestrutura de aplicativos. Contudo, usar o Amazon ECS diretamente proporciona controle mais pormenorizado e acesso um conjunto mais amplo de casos de uso. Para obter mais informações, consulte o Guia do desenvolvedor do AWS Elastic Beanstalk.

Recursos do Amazon ECS

O Amazon ECS é um serviço regional que simplifica a execução de contêineres de aplicativos de uma forma altamente disponível em várias zonas de disponibilidade em uma região. Você pode criar clusters do Amazon ECS em uma VPC nova ou existente. Depois que um cluster estiver funcionando, você poderá criar definições e serviços de tarefa que especificam quais imagens de contêiner do Docker executar em seus clusters. As imagens de contêiner são armazenadas em registros de contêiner e extraídas deles, que podem existir dentro ou fora de sua infraestrutura da AWS.

O diagrama a seguir mostra a arquitetura de um ambiente do Amazon ECS usando o tipo de inicialização Fargate:

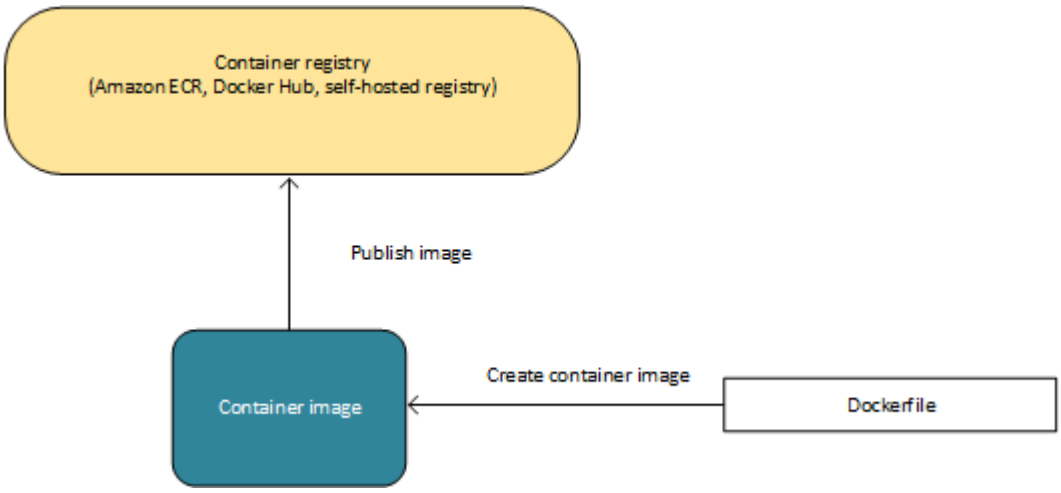


As seções a seguir mergulham nesses elementos individuais da arquitetura do Amazon ECS mais detalhadamente.

Contêineres e imagens

Para implantar aplicativos no Amazon ECS, os componentes de aplicativos devem ser projetados para serem executados em contêineres. Um contêiner do Docker é uma unidade padronizada de desenvolvimento de software, contendo tudo que seu aplicativo de software precisar para executar: código, tempo de execução, ferramentas de sistema, bibliotecas de sistema, etc. Os contêineres são criados a partir de um modelo somente leitura chamado imagem.

Normalmente, as imagens são criadas a partir de um Dockerfile, um arquivo de texto simples que especifica todos os componentes que estão incluídos no contêiner. Essas imagens são armazenadas em um registro, a partir do qual elas podem ser obtidas por download e executadas no seu cluster. Para obter mais informações sobre a tecnologia de contêiner, consulte Noções básicas do Docker para Amazon ECS (p. 11).



Definições de tarefas

Para preparar seu aplicativo para execução no Amazon ECS, crie uma definição de tarefa. A definição de tarefa é um arquivo de texto em formato JSON que descreve um ou mais contêineres, até o máximo de dez, que compõem seu aplicativo. Pode ser considerada como um esquema do seu aplicativo. As definições de tarefas especificam vários parâmetros para seu aplicativo. Exemplos de parâmetros de definição de tarefas são: os contêineres e o tipo de inicialização a serem usados, as portas que devem ser abertas para seu aplicativo e os volumes de dados que devem ser usados com os contêineres na tarefa. Os parâmetros específicos disponíveis para a definição de tarefa dependem de qual tipo de inicialização você está usando. Para obter mais informações sobre como criar definições de tarefa, consulte Definições de tarefa do Amazon ECS (p. 25).

A seguir está um exemplo de uma definição de tarefas que contém um único contêiner que executa um servidor web NGINX usando o tipo de inicialização Fargate. Para um exemplo mais estendido que demonstra uso de vários contêineres em uma definição de tarefa, consulte Definições de tarefa de exemplo (p. 79).

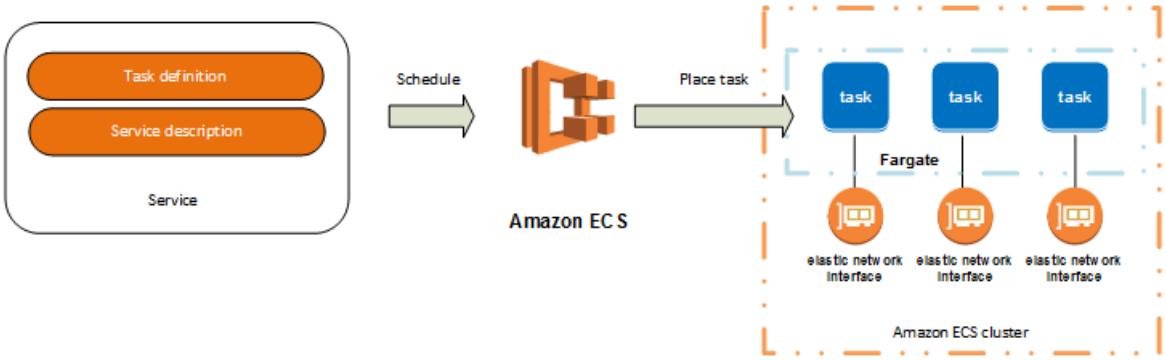
```
{
    "family": "webserver",
    "containerDefinitions": [
        {
            "name": "web",
            "image": "nginx",
            "memory": "100",
            "cpu": "99"
        },
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "networkMode": "awsvpc",
    "memory": "512",
    "cpu": "256",
}
```

Tarefas e programação

Uma tarefa é a instanciação de uma definição de tarefa dentro de um cluster. Depois de criar uma definição de tarefa para seu aplicativo no Amazon ECS você pode especificar o número de tarefas que serão executadas em seu cluster.

Cada tarefa que usa o tipo de inicialização do Fargate tem seu próprio limite de isolamento e não compartilha o kernel, os recursos de CPU, os recursos de memória ou a interface de rede elástica subjacente com outra tarefa.

O programador de tarefas do Amazon ECS é responsável por posicionar tarefas dentro do seu cluster. Há várias opções diferentes de programação disponíveis. Por exemplo, você pode definir um serviço que executa e mantém um número de tarefas especificado simultaneamente. Para obter mais informações sobre as diferentes opções de programação disponíveis, consulte Como programar tarefas do Amazon ECS (p. 90).



Clusters

Ao executar tarefas usando o Amazon ECS, você as coloca em um cluster, que é um agrupamento lógico de recursos. Ao usar o tipo de lançamento Fargate com tarefas no cluster, o Amazon ECS gerencia seus recursos de cluster.

Para obter mais informações sobre como criar clusters, consulte Clusters do Amazon ECS (p. 22).

Conceitos básicos do Amazon ECS

Se você está usando o Amazon ECS pela primeira vez, o Console de gerenciamento da AWS para o Amazon ECS fornece um assistente de primeira execução que conduz você na criação de uma definição de tarefa para um servidor web, na configuração de um serviço e na execução de sua primeira tarefa Fargate. O assistente de primeira execução é altamente recomendado para usuários que não têm nenhuma experiência anterior com o Amazon ECS. Para obter mais informações, consulte o tutorial Conceitos básicos do Amazon ECS (p. 16).

Como alternativa, você pode instalar o AWS Command Line Interface (AWS CLI) para usar o Amazon ECS. Para obter mais informações, consulte Configuração com o Amazon ECS (p. 7).

Serviços relacionados

O Amazon ECS pode ser usado junto com os seguintes serviços da AWS:

AWS Identity and Access Management

O IAM é um serviço da Web que ajuda você a controlar seguramente o acesso de seus usuários aos recursos da AWS. Use o IAM para controlar quem pode usar os recursos da AWS (autenticação) e quais recursos os usuários podem usar e de que maneira (autorização). No Amazon ECS, o IAM pode ser usado para controlar o acesso em nível de instância de contêiner usando funções do IAM, e em nível de tarefa usando funções de tarefa do IAM. Para obter mais informações, consulte Gerenciamento de identidade e acesso do Amazon Elastic Container Service (p. 191).

Amazon EC2 Auto Scaling

O Auto Scaling é um serviço web que permite aumentar ou diminuir a escala das suas tarefas com base em políticas definidas pelo usuário, verificações de status de integridade e cronogramas. Você pode usar o Auto Scaling com uma tarefa Fargate em um serviço para escalar em resposta a uma série de métricas Para obter mais informações, consulte Serviço Auto Scaling (p. 129).

Elastic Load Balancing

O Elastic Load Balancing distribui automaticamente o tráfego de entrada do aplicativo entre as tarefas do seu serviço do Amazon ECS. Ele permite que você obtenha maiores níveis de tolerância a falhas em seus aplicativos, fornecendo continuamente a quantidade necessária de capacidade de balanceamento de carga para distribuir o tráfego de aplicativos. Você pode usar o Elastic Load Balancing para criar um endpoint que balanceie o tráfego entre os serviços em um cluster. Para obter mais informações, consulte Balanceamento de carga do serviço (p. 115).

Amazon Elastic Container Registry

O Amazon ECR é um serviço gerenciado de registro seguro, dimensionável e confiável do Docker da AWS. O Amazon ECR oferece suporte a repositórios do Docker privados com permissões baseadas em recursos usando o IAM, de maneira que usuários específicos ou tarefas possam acessar repositórios e imagens. Os desenvolvedores podem usar a CLI do Docker para enviar, extrair e gerenciar imagens. Para obter mais informações, consulte o Guia do usuário do Amazon Elastic Container Registry.

AWS CloudFormation

O AWS CloudFormation oferece aos desenvolvedores e administradores de sistemas uma maneira fácil de criar e gerenciar um conjunto de recursos relacionados na AWS, fornecendo provisionamento e atualização de uma forma organizada e previsível. Você pode definir clusters, definições de tarefa e serviços como entidades em um script do AWS CloudFormation. Para obter mais informações, consulte Referência do modelo AWS CloudFormation.

Como acessar a Amazon ECS

Você pode trabalhar com o Amazon ECS das seguintes formas:

Console de gerenciamento da AWS

O console é uma interface baseada em navegador para gerenciar recursos do Amazon ECS. Para obter um tutorial que guie você pelo console, consulte Conceitos básicos do Amazon ECS (p. 16).

Ferramentas da linha de comando da AWS

Você pode usar as ferramentas da linha de comando da AWS para executar comandos na linha de comando de seu sistema a fim de realizar tarefas do Amazon ECS e da AWS. Isso pode ser mais rápido e conveniente do que usar o console. As ferramentas da linha de comando também são úteis para criar scripts que executam tarefas da AWS.

A AWS fornece dois conjuntos de ferramentas de linha de comando: a AWS Command Line Interface (AWS CLI) e o AWS Tools para Windows PowerShell. Para obter mais informações, consulte o Guia

do usuário do AWS Command Line Interface e o Guia do usuário do AWS Tools para Windows PowerShell.

CLI do Amazon ECS

Além de usar a AWS CLI para acessar os recursos do Amazon ECS, você pode usar a CLI do Amazon ECS, que fornece comandos de alto nível para simplificar a criação, a atualização e o monitoramento de clusters e tarefas a partir de um ambiente de desenvolvimento local usando o Docker Compose. Para obter mais informações, consulte Como usar a interface de linha de comando do Amazon ECS (p. 248).

SDKs do AWS

Também fornecemos SDKs que permitem que você acesse Amazon ECS usando diferentes linguagens de programação. Os SDKs cuidarão automaticamente de tarefas tais como:

- Assinar criptograficamente suas solicitações de serviço
- Recuperar solicitações
- Lidar com respostas de erro

Para obter mais informações sobre os SDKs disponíveis, consulte Tools for Amazon Web Services.

Configuração com o Amazon ECS

Caso já tenha se cadastrado na Amazon Web Services (AWS) e usado o Amazon Elastic Compute Cloud (Amazon EC2), você está próximo de poder usar o Amazon ECS. O processo de configuração para os dois serviços é semelhante. O guia a seguir prepara você para a ativação do primeiro cluster usando o assistente da primeira execução do Amazon ECS ou a Command Line Interface (CLI – Interface de linha de comando) do Amazon ECS.

Conclua as tarefas a seguir para configuração do Amazon ECS. Caso já tenha concluído qualquer uma dessas etapas, você pode ignorá-las e passar à instalação da AWS CLI personalizada.

Cadastre-se na AWS

Ao se cadastrar na AWS, sua conta da AWS é automaticamente cadastrada em todos os serviços, inclusive Amazon EC2 e Amazon ECS. Você será cobrado apenas pelos serviços que usar.

Se já tiver uma conta da AWS, passe para a próxima tarefa. Se você ainda não possuir uma conta da AWS, use o procedimento a seguir para criar uma.

Para criar uma conta da AWS

1. Abra https://portal.aws.amazon.com/billing/signup.
2. Siga as instruções online.

   Parte do procedimento de cadastro envolve uma chamada telefônica e a digitação de um código de verificação usando o teclado do telefone.

Anote o número da conta da AWS, pois você precisará dele na próxima tarefa.

Criar um usuário do IAM

Os serviços na AWS, como Amazon EC2 e Amazon ECS, exigem que você forneça credenciais ao acessá-los, de maneira que o serviço possa determinar se você tem permissão para acessar os recursos. O console requer sua senha. Você pode criar chaves de acesso para sua conta da AWS para acessar a interface de linha de comando ou a API. Contudo, não recomendamos que você acesse a AWS usando as credenciais de sua conta da AWS; em vez disso, recomendamos que você use o AWS Identity and Access Management (IAM). Crie um usuário do IAM e adicione o usuário a um grupo do IAM com permissões administrativas ou conceda permissões administrativas a esse usuário. Em seguida, você poderá acessar a AWS usando uma URL especial e as credenciais do usuário do IAM.

Se você tiver se cadastrado na AWS, mas não criou um usuário do IAM para você mesmo, poderá criar um usando o console do IAM.

Para criar um usuário administrador para você mesmo e adicionar o usuário a um grupo de administradores (console)

1. Use seu endereço de e-mail e senha da conta da AWS para fazer login como Usuário raiz da conta da AWS no console do IAM em https://console.aws.amazon.com/iam/.

> **Note**
>
> Recomendamos que você siga as melhores práticas para utilizar o usuário do **`Administrator`** IAM abaixo e armazene as credenciais do usuário raiz com segurança. Cadastre-se como usuário raiz para executar somente algumas tarefas de gerenciamento de serviços e contas.

2. No painel de navegação, escolha Usuários e depois Adicionar usuário.
3. Em User name (Nome do usuário), digite **`Administrator`**.
4. Marque a caixa de seleção ao lado de Console de gerenciamento da AWS access (Acesso ao &console;). Então, selecione Custom password (Senha personalizada), e insira sua nova senha na caixa de texto.
5. (Opcional) Por padrão, a AWS exige que o novo usuário crie uma senha ao fazer login pela primeira vez. Você pode desmarcar a caixa de seleção próxima de User must create a new password at next sign-in (O usuário deve criar uma senha no próximo login) para permitir que o novo usuário redefina a senha depois de fazer login.
6. Escolha Próximo: Permissões.
7. Em Set permissions (Conceder permissões), escolha Add user to group (Adicionar usuário ao grupo).
8. Escolha Create group (Criar grupo).
9. Na caixa de diálogo Create group (Criar grupo), em Group name (Nome do grupo), digite **`Administrators`**.
10. Escolha Filter policies (Filtrar políticas) e, depois, selecione AWS managed -job function (Função de trabalho gerenciado pela &AWS;) para filtrar o conteúdo de tabelas.
11. Na lista de políticas, marque a caixa de seleção AdministratorAccess. A seguir escolha Criar grupo.

    > **Note**
    >
    > Você deve ativar o acesso de usuário e função do IAM ao faturamento para poder usar as permissões do `AdministratorAccess` a fim de acessar o console do AWS Billing and Cost Management. Para fazer isso, siga as instruções na etapa 1 do tutorial sobre como delegar acesso ao console de faturamento.

12. Suporte a lista de grupos, selecione a caixa de seleção para seu novo grupo. Escolha Refresh (Atualizar) caso necessário, para ver o grupo na lista.
13. Escolha Next: Tags (Próximo: tags).
14. (Opcional) Adicione metadados ao usuário anexando tags como pares de chave-valor. Para obter mais informações sobre como usar tags no IAM, consulte Marcar entidades do IAM no Guia do usuário do IAM.
15. Escolha Next: Review (Próximo: Análise) para ver uma lista de associações de grupos a serem adicionadas ao novo usuário. Quando você estiver pronto para continuar, selecione Criar usuário.

Você pode usar esse mesmo processo para criar mais grupos e usuários e conceder aos seus usuários acesso aos recursos de sua conta da AWS. Para saber como usar políticas para restringir as permissões de usuário a recursos específicos da AWS, acesse Gerenciamento de acesso e Políticas de exemplo.

Para fazer login como esse novo usuário do IAM, faça logout no console da AWS e use a seguinte URL, em que your_aws_account_id é o número de sua conta da AWS sem hifens (por exemplo, se o número da conta da AWS é `1234-5678-9012`, o ID da conta da AWS é `123456789012`):

```
https://your_aws_account_id.signin.aws.amazon.com/console/
```

Insira o nome e a senha de usuário do IAM que você acabou de criar. Quando você está conectado, a barra de navegação exibe "your_user_name @ your_aws_account_id".

Se você não quiser que a URL da página de cadastro contenha o ID da sua conta da AWS, crie um alias da conta. No painel do IAM, escolha Create Account Alias (Criar alias de conta) e insira um alias, como o nome de sua empresa. Para fazer o login depois de criar o alias de uma conta, use o seguinte URL:

```
https://your_account_alias.signin.aws.amazon.com/console/
```

Para verificar o link de login de usuários do IAM para sua conta, abra o console do IAM e verifique no link de login de usuários do IAM no painel.

Para obter mais informações sobre o IAM, consulte o Guia do usuário do AWS Identity and Access Management.

Criar uma função do IAM

Antes que o agente de contêiner do Amazon ECS possa fazer chamadas para as ações de API do Amazon ECS em seu nome, ele exige uma política e uma função do IAM para o serviço saber que o agente pertence a você.

Para tarefas que usam o tipo de inicialização Fargate, você pode criar uma função do IAM que permite que o agente extraia imagens de contêiner do Amazon ECR ou use o driver de log awslogs, que atualmente é a única opção de registro com suporte para esse tipo de inicialização. Essa função é chamada de função do IAM para execução de tarefa do Amazon ECS. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

> **Note**
>
> Essas funções do IAM são criadas automaticamente para você na primeira execução do console do Amazon ECS. Assim, caso pretenda usar o console, você pode avançar para a próxima seção. Se você não pretende usar o console e planeja usar a AWS CLI, essas funções do IAM precisarão ser criadas manualmente.

Criar uma nuvem privada virtual

A Amazon Virtual Private Cloud (Amazon VPC) permite executar os recursos da AWS em uma rede virtual definida por você.

> **Note**
>
> A primeira execução do console do Amazon ECS cria uma VPC para o cluster. Assim, caso pretenda usar o console do Amazon ECS, você pode avançar para a próxima seção.

Caso tenha uma VPC padrão, você também pode ignorar esta seção e avançar à próxima tarefa, Instalar a AWS CLI (p. 10). Para determinar se você tem uma VPC padrão, consulte Plataformas compatíveis com o console do Amazon EC2 no Guia do usuário do Amazon EC2 para instâncias do Linux. Do contrário, você pode criar uma VPC não padrão na conta usando as etapas abaixo.

> **Important**
>
> Caso a conta dê suporte ao Amazon EC2 Classic em uma região, você não tem uma VPC padrão nessa região.

Para criar um VPC não padrão

1. Abra o console de Amazon VPC em https://console.aws.amazon.com/vpc/.

2. Na barra de navegação, selecione uma região para a VPC. Os VPCs são específicos para uma região, portanto, você deve selecionar a mesma região em que criou o par de chaves.
3. No painel da VPC, selecione Launch VPC Wizard (Iniciar assistente da VPC).
4. Na página Etapa 1: selecione uma configuração de VPC, verifique se VPC com uma única sub-rede pública está selecionado e escolha Selecionar.
5. Na página Step 2: VPC with a Single Public Subnet, insira um nome amigável para a VPC no campo VPC name. Deixe as outras definições de configuração padrão e escolha Create VPC. Na página de confirmação, escolha OK.

Para obter mais informações sobre a Amazon VPC, consulte O que é a Amazon VPC? no Guia do usuário da Amazon VPC.

Instalar a AWS CLI

O Console de gerenciamento da AWS pode ser usado para gerenciar manualmente todas as operações com o Amazon ECS. Porém, a instalação da AWS CLI na área de trabalho local ou na caixa de um desenvolvedor permite criar scripts que podem automatizar tarefas de gerenciamento comuns no Amazon ECS.

Para usar a AWS CLI com o Amazon ECS, instale a versão mais recente da AWS CLI. Para obter informações sobre a instalação da AWS CLI ou a atualização para a versão mais recente, consulte Instalar a interface da linha de comando da AWS no Guia do usuário do AWS Command Line Interface.

Noções básicas do Docker para Amazon ECS

Docker é uma tecnologia que permite criar, executar, testar e implantar aplicativos distribuídos que são baseados em contêineres Linux. Amazon ECS usa imagens do Docker em definições de tarefas para executar contêineres nas instâncias do Amazon EC2 em seus clusters. Para ver detalhes do produto, estudos de caso de clientes em destaque e perguntas frequentes sobre o Amazon ECS, consulte as páginas de detalhes do produto Amazon Elastic Container Service.

A documentação neste guia supõe que os leitores possuem uma compreensão básica do Docker e de como ele funciona. Para obter mais informações sobre o Docker, consulte O que é o Docker? e Visão geral do Docker.

Tópicos

- Instalação do Docker (p. 11)
- Criar uma imagem do Docker (p. 12)
- (Opcional) Enviar sua imagem por push para o Amazon Elastic Container Registry (p. 14)
- Limpar (opcional) (p. 15)
- Próximas etapas (p. 15)

Instalação do Docker

> **Note**
>
> Se você já tiver um Docker instalado, vá para Criar uma imagem do Docker (p. 12).

O Docker está disponível em muitos sistemas operacionais diferentes, incluindo a maioria das distribuições modernas do Linux, como Ubuntu, e até o Mac OSX e o Windows. Para obter mais informações sobre como instalar o Docker no seu sistema operacional, consulte o Guia de instalação do Docker.

Não é necessário nem mesmo um sistema de desenvolvimento local para usar o Docker. Se você já usa o Amazon EC2, poderá iniciar uma instância e instalar o Docker para começar.

Para instalar o Docker em uma instância do Amazon EC2

1. Execute uma instância com a AMI do Amazon Linux 2. Para obter mais informações, consulte Executar de uma instância no Guia do usuário do Amazon EC2 para instâncias do Linux.
2. Conecte-se à sua instância. Para obter mais informações, consulte Conectar-se à instância do Linux no Guia do usuário do Amazon EC2 para instâncias do Linux.
3. Atualize os pacotes instalados e o cache de pacotes em sua instância.

```
sudo yum update -y
```

4. Instale o pacote do Docker Community Edition mais recente.

```
sudo amazon-linux-extras install docker
```

5. Inicie o serviço Docker.

```
sudo service docker start
```

6. Adicione o `ec2-user` ao grupo `docker`, de modo que você possa executar comandos do Docker sem usar o `sudo`.

```
sudo usermod -a -G docker ec2-user
```

7. Faça logout e login novamente para selecionar as novas permissões do grupo `docker`. Você pode fazer isso ao fechar a janela de terminal SSH atual e se reconectar à sua instância em outra janela. Sua nova sessão SSH terá as permissões de grupo `docker` apropriadas.
8. Verifique se o `ec2-user` pode executar comandos do Docker sem `sudo`.

```
docker info
```

> **Note**
>
> Em alguns casos, pode ser necessário reinicializar sua instância para fornecer permissões para o `ec2-user` acessar o daemon do Docker. Tente reinicializar sua instância se você vir o seguinte erro:
>
> ```
> Cannot connect to the Docker daemon. Is the docker daemon running on this host?
> ```

Criar uma imagem do Docker

As definições de tarefa do Amazon ECS usam imagens de docker para executar contêineres nas instâncias de contêiner em seus clusters. Nesta seção, crie uma imagem de docker de um aplicativo web simples e teste-a no seu sistema local ou instância do EC2. Em seguida, envie a imagem ao registro de contêiner (como o Amazon ECR ou o Docker Hub) para poder usá-la em uma definição de tarefa do ECS.

Para criar uma imagem do Docker de um aplicativo web simples

1. Crie um arquivo chamado `Dockerfile`. Dockerfile é um manifesto que descreve a imagem básica a ser usada para a sua imagem do Docker e o que você deseja instalar e executar nela. Para obter mais informações sobre a Dockerfiles, visite Referência de Dockerfiles.

```
touch Dockerfile
```

2. Edite o `Dockerfile` que você acabou de criar e adicione o conteúdo a seguir.

```
FROM ubuntu:16.04

# Install dependencies
RUN apt-get update
RUN apt-get -y install apache2

# Install apache and write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure apache
RUN echo '. /etc/apache2/envvars' > /root/run_apache.sh
RUN echo 'mkdir -p /var/run/apache2' >> /root/run_apache.sh
RUN echo 'mkdir -p /var/lock/apache2' >> /root/run_apache.sh
RUN echo '/usr/sbin/apache2 -D FOREGROUND' >> /root/run_apache.sh
```

```
RUN chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

Esse Dockerfile usa a imagem do Ubuntu 16.04. As instruções de `RUN` atualizam os caches do pacote, instalam alguns pacotes de software para o servidor web e, em seguida, gravam o conteúdo "Hello World!" na raiz do documento do servidor web. A instrução `EXPOSE` expõe a porta 80 no contêiner, e a instrução `CMD` inicia o servidor web.

3. Crie a imagem do Docker do seu Dockerfile.

   Note

   Algumas versões do Docker podem exigir o caminho completo para o seu Dockerfile no seguinte comando, em vez de o caminho relativo mostrado abaixo.

```
docker build -t hello-world .
```

4. Execute docker images para verificar se a imagem foi criada corretamente.

```
docker images --filter reference=hello-world
```

   Resultado:

```
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world         latest              e9ffedc8c286        4 minutes ago       241MB
```

5. Execute a imagem recém-criada. A opção `-p 80:80` mapeia a porta 80 exposta no contêiner para a porta 80 no sistema de host. Para obter mais informações sobre o docker run, acesse a Referência de execução do Docker.

```
docker run -t -i -p 80:80 hello-world
```

   Note

   A saída do servidor Web Apache é exibida na janela do terminal. Você pode ignorar a mensagem "`Could not reliably determine the server's fully qualified domain name`".

6. Abra um navegador e aponte para o servidor que está executando o Docker e hospedando seu contêiner.

   - Se você estiver usando uma instância do EC2, esse é o valor Public DNS para o servidor, que é o mesmo endereço usado para se conectar à instância com o SSH. Certifique-se de que o security group para sua instância permita o tráfego de entrada na porta 80.
   - Se você estiver executando o Docker localmente, aponte seu navegador para http://localhost/.
   - Se você estiver usando docker-machine em um computador Windows ou Mac, localize o endereço IP da VM VirtualBox que está hospedando o Docker com o comando docker-machine ip, substituindo *`machine-name`* pelo nome da máquina de docker que você está usando.

```
docker-machine ip machine-name
```

   Você deverá ver uma página da web com a sua instrução "Hello World!".

7. Interrompa o contêiner do Docker, digitando Ctrl+c.

(Opcional) Enviar sua imagem por push para o Amazon Elastic Container Registry

O Amazon ECR é um serviço de registro gerenciado do Docker da AWS. Os clientes podem usar a CLI do Docker que já conhecem para enviar por push, extrair e gerenciar imagens. Para ver detalhes do produto, estudos de caso de clientes em destaque e perguntas frequentes sobre o Amazon ECR, consulte as páginas de detalhes do produto Amazon Elastic Container Registry.

Esta seção requer o seguinte:

- Tenha a AWS CLI instalada e configurada. Se você não tiver a AWS CLI instalada em seu sistema, consulte Instalar a AWS Command Line Interface no Guia do usuário do AWS Command Line Interface.
- Seu usuário deve ter as permissões necessárias do IAM para acessar o serviço do Amazon ECR. Para obter mais informações, consulte Políticas gerenciadas do Amazon ECR.

Para marcar a imagem e enviá-la por push para o Amazon ECR

1. Crie um repositório do Amazon ECR para armazenar sua imagem `hello-world`. Observe `repositoryUri` na saída.

```
aws ecr create-repository --repository-name hello-repository --region region
```

   Resultado:

```
{
    "repository": {
        "registryId": "aws_account_id",
        "repositoryName": "hello-repository",
        "repositoryArn": "arn:aws:ecr:region:aws_account_id:repository/hello-
repository",
        "createdAt": 1505337806.0,
        "repositoryUri": "aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository"
    }
}
```

2. Marque a imagem `hello-world` com o valor `repositoryUri` da etapa anterior.

```
docker tag hello-world aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

3. Execute o comando aws ecr get-login --no-include-email para obter a string de comando de autenticação docker login para seu registro.

   Note

   O comando get-login está disponível na AWS CLI desde a versão 1.9.15, mas recomendamos a versão 1.11.91 ou posterior para versões recentes do Docker (17.06 ou posterior). Você pode verificar a versão da AWS CLI com o comando aws --version. Se você estiver usando a versão 17.06 do Docker ou posterior, inclua a opção `--no-include-email` após `get-login`. Se você receber um erro `Unknown options: --no-include-email`, instale a versão mais recente da CLI da AWS. Para obter mais informações, consulte Instalar a interface de linha de comando da AWS no Guia do usuário do AWS Command Line Interface.

```
aws ecr get-login --no-include-email --region region
```

4. Execute o comando docker login retornado na etapa anterior. Esse comando fornece um token de autorização válido por 12 horas.

   Important

   Durante a execução desse comando docker login, a string de comando pode ser visível a outros usuários no sistema em uma exibição da lista de processos (ps -e). Como o comando docker login contém credenciais de autenticação, há risco de que outros usuários no sistema possam visualizá-las. Eles podem usar as credenciais para conseguir acesso de envio aos repositórios. Se você não estiver em um sistema seguro, considere esses riscos e efetue login interativamente ao omitir a opção `-p` *`password`* e forneça a senha quando solicitado.

5. Envie a imagem por push para o Amazon ECR com o valor `repositoryUri` da etapa anterior.

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

Limpar (opcional)

Quando acabar de testar sua imagem do Amazon ECR, você poderá excluir o repositório para não ser cobrado pelo armazenamento de imagens.

```
aws ecr delete-repository --repository-name hello-repository --region region --force
```

Próximas etapas

Agora que você criou uma imagem do Docker e a enviou por push para um repositório do Amazon ECR, você pode começar a criar seus recursos do Amazon ECS para iniciar um contêiner. Use os tópicos a seguir para continuar:

- Conclua os pré-requisitos. Para obter mais informações, consulte Configuração com o Amazon ECS (p. 7).
- Para demonstrações da AWS CLI, consulte Usar a AWS CLI com o Amazon ECS (p. 265).
- Para demonstrações do Console de gerenciamento da AWS, consulte Conceitos básicos do Amazon ECS (p. 16).

Conceitos básicos do Amazon ECS

Comece a usar o Amazon Elastic Container Service (Amazon ECS), criando os recursos necessários doAmazon ECS para executar sua primeira tarefa. O console do Amazon ECS fornece uma experiência de primeira execução que facilita essa configuração.

Nas regiões que dão suporte ao AWS Fargate, o assistente executado pela primeira vez pelo Amazon ECS o orienta no processo de conceitos básicos do Amazon ECS usando o Fargate. Para obter mais informações, consulte Tipo de inicialização Fargate (p. 59). O assistente dá a opção de criar um cluster e executar um aplicativo web de exemplo. Se já tiver uma imagem do Docker a ser iniciada no Amazon ECS, você poderá criar uma definição de tarefa com essa imagem e usá-la no cluster.

Nas regiões que não oferecem suporte ao AWS Fargate, o assistente executado pela primeira vez pelo Amazon ECS o orienta no processo de conceitos básicos das tarefas que usam o tipo de inicialização do EC2. O assistente dá a opção de criar um cluster e executar um aplicativo web de exemplo. Se já tiver uma imagem do Docker a ser iniciada no Amazon ECS, você poderá criar uma definição de tarefa com essa imagem e usá-la no cluster.

Tópicos

- Conceitos básicos do Amazon ECS que usa Fargate (p. 16)

Conceitos básicos do Amazon ECS que usa Fargate

Vamos começar com o Amazon Elastic Container Service (Amazon ECS) usando o tipo de inicialização Fargate ao criar uma definição de tarefa, programar tarefas e configurar um cluster no console do Amazon ECS.

Nas regiões que dão suporte ao AWS Fargate, o assistente executado pela primeira vez pelo Amazon ECS o orienta no processo de conceitos básicos do Amazon ECS usando o Fargate. Para obter mais informações, consulte Tipo de inicialização Fargate (p. 59). O assistente dá a opção de criar um cluster e executar um aplicativo web de exemplo. Se já tiver uma imagem do Docker a ser iniciada no Amazon ECS, você poderá criar uma definição de tarefa com essa imagem e usá-la no cluster.

> Important
>
> Para obter mais informações sobre o assistente executado pela primeira vez pelo Amazon ECS para tarefas EC2, consulte Conceitos básicos do Amazon ECS.

Conclua as seguintes tarefas e conceitos básicos no Amazon ECS usando o Fargate:

Pré-requisitos

Antes de começar, certifique-se de que você tenha concluído as etapas em Configuração com o Amazon ECS (p. 7) e que o usuário da AWS tenha as permissões especificadas no exemplo de política do IAM `AdministratorAccess` ou Permissões do assistente de primeira execução do Amazon ECS (p. 202).

O assistente executado pela primeira vez tenta criar automaticamente a função do IAM de execução da tarefa, o que é obrigatório para tarefas Fargate. Para garantir que a experiência da primeira execução seja capaz de criar essa função do IAM, uma das seguintes opções devem ser verdadeiras:

- O usuário tem acesso de administrador. Para obter mais informações, consulte Configuração com o Amazon ECS (p. 7).
- O usuário tem as permissões do IAM para criar uma função do serviço. Para obter mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS .
- Um usuário com acesso de administrador criou manualmente a função de execução da tarefa, de maneira que ela esteja disponível na conta a ser usada. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Etapa 1: Criar uma definição de tarefa

Uma definição de tarefa é como um guia para seu aplicativo. Sempre que iniciar uma tarefa no Amazon ECS, você especificará uma definição de tarefa. Dessa maneira, o serviço sabe qual imagem do Docker usar para os contêineres, quantos contêineres usar na tarefa e a alocação de recursos para cada contêiner.

1. Abra o assistente de primeira execução do console do Amazon ECS em https://console.aws.amazon.com/ecs/home#/firstRun.
2. Na barra de navegação, selecione a região US East (N. Virginia) (Leste dos EUA [Norte da Virgínia]).

   Note

   Você pode concluir esse assistente executado pela primeira vez usando essas etapas para qualquer região que suporte o Amazon ECS usando Fargate. Para obter mais informações, consulte Tipo de inicialização Fargate (p. 59).
3. Configure seus parâmetros de definição de tarefas do contêiner.

   Para Container definition (Definição de contêiner), o assistente de primeira execução vem pré-carregado nas definições de contêiner `sample-app`, `nginx` e `tomcat-webserver` no console. Opcionalmente, você pode renomear o contêiner ou revisar e editar os recursos usados pelo contêiner (como unidades de CPU e limites de memória) escolhendo Edit e editando os valores mostrados. Para obter mais informações, consulte Definições de contêiner (p. 35).

   Note

   Se você estiver usando uma imagem do Amazon ECR na sua definição de contêiner, use a nomenclatura `registry/repository:tag` completa para suas imagens do Amazon ECR. Por exemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*.
4. Em Task definition (Definição de tarefa), o assistente de primeira execução define uma definição de tarefa para usar com as definições de contêiner pré-carregadas. Opcionalmente, você pode renomear a definição de tarefa e editar os recursos usados por ela (como os valores Task memory e Task CPU), escolhendo Edit e editando os valores mostrados. Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34).

   As definições de tarefas criadas no assistente de primeira execução estão limitadas a um único contêiner para fins de simplificação. Você pode criar várias definições de tarefas de contêiner posteriormente no console do Amazon ECS.
5. Escolha Next.

Etapa 2: Configurar o serviço

Nessa seção do assistente, selecione como configurar o serviço Amazon ECS que é criado a partir de sua definição de tarefa. Um serviço é executado e mantém um número específico de cópias da definição da tarefa no seu cluster. O aplicativo Amazon ECS sample (Amostra do ECS) é um aplicativo em estilo –

Hello World baseado na web a ser executado indefinidamente. Executando-o como um serviço, ele será reiniciado se a tarefa se tornar não íntegra ou parar inesperadamente.

O assistente de primeira execução vem pré-carregado com uma definição de serviço, e você pode ver o serviço `sample-app-service` definido no console. Você também pode renomear o serviço ou revisar e editar os detalhes escolhendo Edit e fazendo o seguinte:

1. No campo Service name, selecione um nome para seu serviço.
2. No campo Number of desired tasks (Número de tarefas desejadas), insira o número de tarefas a serem executadas com a sua definição de tarefa especificada.
3. No campo Security group (Grupo de segurança), especifique um intervalo de endereços IPv4 para permitir tráfego de entrada, em notação do bloco CIDR. Por exemplo, `203.0.113.0/24`.
4. (Opcional) Você pode escolher usar um Balanceador de carga de aplicações para seu serviço. Quando é executada a partir de um serviço configurado para usar um load balancer, uma tarefa é registrada com o load balancer. O tráfego do load balancer é distribuído entre as instâncias no load balancer. Para obter mais informações, consulte Introdução a Application Load Balancers.

   Important

   Os Balanceador de carga de aplicaçõess não geram custo enquanto existem em seus recursos da AWS. Para obter mais informações, consulte Definição de preço Balanceador de carga de aplicações.

   Execute as etapas a seguir para usar um load balancer com seu serviço.

   - Na seção Container to load balance (Contêiner para balanceamento de carga), escolha a Load balancer listener port (Porta de escuta do load balancer). O valor padrão aqui é configurado para o aplicativo de amostra, mas você pode configurar diferentes opções de listener para o load balancer. Para obter mais informações, consulte Balanceamento de carga do serviço (p. 115).
5. Verifique suas configurações de serviço e clique em Save, Next.

Etapa 3: Configurar o cluster

Nesta seção do assistente, você dá um nome ao cluster e o Amazon ECS cuida das redes e configuração do IAM para você.

1. No campo Cluster name, escolha um nome para o seu cluster.
2. Clique em Next para continuar.

Etapa 4: Revisão

1. Analise a sua definição de tarefa, a configuração da tarefa e a configuração de cluster e clique em Create (Criar) para terminar. Você é direcionado para uma página Launch Status (Status da execução) que mostra o status da execução. Ele descreve cada etapa do processo (pode demorar alguns minutos para ser concluída enquanto o grupo do Auto Scaling é criado e preenchido).
2. Depois que a execução for concluída, escolha View service (Exibir serviço).

Etapa 5: (Opcional) Exibir o serviço

Se o serviço for um aplicativo baseado na Web, como o aplicativo Amazon ECS sample (Amostra do ECS), você pode visualizar seus contêineres com um navegador da web.

1. Na página Service: ***`service-name`***, selecione a guia Tasks.

2. Escolha uma tarefa na lista de tarefas em seu serviço.
3. Na seção Network, escolha o ENI Id da sua tarefa. Isso leva você para o console do Amazon EC2, onde você pode ver os detalhes da interface de rede associados à sua tarefa, incluindo o endereço IPv4 Public IP (IP público IPv4).
4. Insira o endereço IPv4 Public IP (IP público IPv4) no navegador da web e veja uma página da web que exibe o aplicativo Amazon ECS sample (Amostra do ECS).

Etapa 6: limpeza

Ao terminar de usar um cluster do Amazon ECS, é necessário limpar os recursos associados a ele para evitar cobranças por recursos que você não está usando.

Alguns recursos do Amazon ECS, como tarefas, serviços, clusters e instâncias de contêiner, são eliminados utilizando o console do Amazon ECS. Outros recursos, como instâncias do Amazon EC2, load balancers do Elastic Load Balancing e grupos do Auto Scaling, devem ser limpos manualmente no console do Amazon EC2 ou com a exclusão da pilha do AWS CloudFormation que os criou.

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, escolha Clusters.
3. Na página Clusters, selecione o cluster a ser excluído.
4. Escolha Delete Cluster. No prompt de confirmação, insira delete me e escolha Delete (Excluir). A exclusão do cluster limpa os recursos associados que foram criados com o cluster, incluindo VPCs, load balancers ou grupos do Auto Scaling.

Versões de plataforma do AWS Fargate

As versões da plataforma AWS Fargate são usadas para fazer referência a um ambiente de tempo de execução para infraestrutura de tarefas do Fargate. Trata-se de uma combinação da versão do kernel e do tempo de execução do contêiner.

As novas versões da plataforma são lançadas à medida que o ambiente de tempo de execução se desenvolve, por exemplo, em caso de atualizações no kernel ou no sistema operacional, novos recursos, correções de erros ou atualizações de segurança. As atualizações de segurança e as correções são implantadas automaticamente em suas tarefas do Fargate. Se for encontrado um problema de segurança que afeta uma versão da plataforma, a AWS corrigirá a versão da plataforma. Em alguns casos, você poderá ser notificado de que as tarefas Fargate foram programadas para desativação. Para obter mais informações, consulte Desativação da tarefa (p. 96).

Tópicos

- Considerações da versão da plataforma (p. 20)
- Versões disponíveis da plataforma do AWS Fargate (p. 20)

Considerações da versão da plataforma

Considere as seguintes informações ao especificar a versão da plataforma:

- Ao especificar a versão da plataforma, é possível usar o número dela (por exemplo, `1.3.0`) ou `LATEST`.
- Para usar uma versão específica da plataforma, especifique o número dela ao criar ou atualizar seu serviço. Se você especificar `LATEST`, suas tarefas usarão a versão mais atual disponível da plataforma, que pode não ser a versão mais recente.
- Nas regiões China (Pequim) e China (Ningxia), a única versão da plataforma compatível é `1.3.0`. O Console de gerenciamento da AWS exibe versões da plataforma mais antigas, mas um erro será retornado se elas forem escolhidas. A versão da plataforma `LATEST` é compatível porque usa a versão da plataforma `1.3.0`.
- Se você tiver um serviço com tarefas em execução e quiser atualizar a versão da plataforma, atualize o serviço, especifique a nova versão e escolha Force new deployment (Forçar nova implantação). Suas tarefas serão reimplantadas com a versão da plataforma mais recente. Para obter mais informações, consulte Atualizar um serviço (p. 160).
- Caso seu serviço seja expandido sem atualizar a versão da plataforma, essas tarefas receberão a versão especificada na implantação atual dele.

Versões disponíveis da plataforma do AWS Fargate

Esta é uma lista das versões da plataforma disponíveis no momento:

Plataforma do Fargate versão-1.3.0

- Adicionada a reciclagem de tarefas para as tarefas Fargate, que é o processo de atualização das tarefas que compõem um serviço do Amazon ECS. Para obter mais informações, consulte Reciclagem de tarefas Fargate (p. 97).

- A partir de 27 de março de 2019, todas as novas tarefas do Fargate que forem iniciadas poderão usar parâmetros adicionais de definição de tarefas que permitem que você defina uma configuração de proxy, dependências para inicialização e desligamento de contêiner, além de um valor de tempo limite de início e interrupção por contêiner. Para obter mais informações, consulte Configuração do proxy (p. 56), Dependência de contêiner (p. 51) e Tempos limite de contêiner (p. 52).
- A partir de 2 de abril de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte à injeção de dados confidenciais nos seus contêineres. Isso é feito armazenando seus dados confidenciais em segredos do AWS Secrets Manager ou em parâmetros do Parameter Store do AWS Systems Manager e fazendo referência a eles na definição de contêiner. Para mais informações, consulte Especificação de dados confidenciais (p. 73).
- A partir de 1º de maio de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte à referência a dados confidenciais na configuração de log de um contêiner usando o parâmetro de definição de contêiner `secretOptions`. Para mais informações, consulte Especificação de dados confidenciais (p. 73).
- A partir de 1º de maio de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte ao driver de log `splunk`, além do driver de log `awslogs`. Para obter mais informações, consulte Armazenamento e registro em log (p. 44).
- A partir de 9 de julho de 2019, todas as tarefas novas do Fargate que for iniciada oferecerá suporte ao Container Insights do CloudWatch. Para obter mais informações, consulte CloudWatch Container Insights do Amazon ECS (p. 186).

Plataforma do Fargate versão-1.2.0

- Adicionado suporte para autenticação de registro privado usando AWS Secrets Manager. Para obter mais informações, consulte Autenticação de registro privado para tarefas (p. 71).

Plataforma do Fargate versão-1.1.0

- Adicionado suporte para o endpoint de metadados de tarefa do Amazon ECS. Para obter mais informações, consulte Endpoint de metadados de tarefas do Amazon ECS (p. 272).
- Inclusão de suporte às verificações de integridade do Docker nas definições de contêiner. Para obter mais informações, consulte Verificação de integridade (p. 40).
- Suporte adicionado para a descoberta de serviço do Amazon ECS. Para obter mais informações, consulte Descoberta de serviço (p. 137).

Plataforma do Fargate versão-1.0.0

- Com base no Amazon Linux de setembro de 2017.
- Versão inicial.

Clusters do Amazon ECS

Um cluster do Amazon ECS é um agrupamento lógico de tarefas ou serviços. Na primeira vez que você usar o Amazon ECS, um cluster padrão será criado, mas você poderá criar vários clusters em uma conta para manter os recursos separados.

Veja a seguir os conceitos gerais sobre clusters do Amazon ECS.

- Os clusters são específicos da região.
- Os clusters podem conter tarefas usando os tipos de inicialização Fargate e EC2. Para obter mais informações sobre tipos de inicialização, consulte Tipos de inicialização Amazon ECS (p. 59).
- Você pode criar políticas do IAM personalizadas para seus clusters para permitir ou restringir o acesso do usuário a clusters específicos. Para obter mais informações, consulte a seção Exemplos de cluster (p. 205) em Exemplos de políticas baseadas em identidade do Amazon Elastic Container Service (p. 200).
- Os clusters com tarefas do Fargate podem ser dimensionados usando o Serviço Auto Scaling. Para obter mais informações, consulte Serviço Auto Scaling (p. 129).

Tópicos

- Criação de um cluster (p. 22)
- Atualizar configurações de cluster (p. 23)
- Exclusão de um cluster (p. 23)

Criação de um cluster

Você pode criar um cluster do Amazon ECS usando o Console de gerenciamento da AWS, como descrito neste tópico. Antes de começar, você deve concluir as etapas em Configuração com o Amazon ECS (p. 7).

> **Note**
>
> Esse assistente de criação de cluster permite criar facilmente os recursos necessários a um cluster do Amazon ECS. Ele também permite que você personalize várias opções comuns de configuração de cluster. No entanto, o assistente não permite que você personalize toda opção de recurso. Se os requisitos exigidos forem além dos compatíveis com esse assistente, considere o uso de nossa arquitetura de referência em https://github.com/awslabs/ecs-refarch-cloudformation. Não tente modificar diretamente os recursos subjacentes após eles terem sido criados pelo assistente.

Para criar um cluster

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região a ser usada.
3. No painel de navegação, escolha Clusters.
4. Na página Clusters, escolha Create Cluster (Criar cluster).
5. Em Select cluster compatibility (Selecionar a compatibilidade do cluster), escolha Networking only (Apenas em rede) e Next Step (Próxima etapa).
6. Na página Configure cluster (Configurar cluster), insira um Cluster name (Nome do cluster). São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.

7. Na seção Networking (Rede), configure a VPC para seu cluster. Você pode manter as configurações padrão ou pode modificar essas configurações com as etapas a seguir.

   a. (Opcional) Se você optar por criar uma nova VPC, em CIDR Block (Bloco CIDR), selecione um bloco CIDR para sua VPC. Para obter mais informações, consulte Sua VPC e suas sub-redes em Guia do usuário da Amazon VPC.

   b. Em Subnets, selecione as sub-redes a serem usadas para sua VPC. Você pode manter as configurações padrão ou modificá-las para atender às suas necessidades.
8. Na seção CloudWatch Container Insights, escolha se o Container Insights deve ser habilitado para o cluster. Para obter mais informações, consulte CloudWatch Container Insights do Amazon ECS (p. 186).
9. Escolha Criar.

Atualizar configurações de cluster

As configurações de cluster permitem que você defina parâmetros para seus clusters do Amazon ECS existentes. É possível atualizar as configurações de cluster usando a API do Amazon ECS, a AWS CLI ou os SDKs. Atualmente, a única configuração de cluster compatível é `containerInsights`, que permite habilitar ou desabilitar o CloudWatch Container Insights para um cluster existente. Para habilitar o CloudWatch Container Insights para um novo cluster, isso pode ser feito no Console de gerenciamento da AWS durante a criação do cluster. Para obter mais informações, consulte Criação de um cluster (p. 22).

> **Important**
>
> Atualmente, se você excluir um cluster existente que não tenha o Container Insights habilitado e criar um novo cluster com o mesmo nome com o Container Insights habilitado, o Container Insights não será realmente habilitado. Se quiser preservar o mesmo nome para o cluster existente e habilitar o Container Insights, será necessário aguardar 7 dias para que você possa recriá-lo.

Como atualizar as configurações de um cluster usando a linha de comando

Use um dos seguintes comandos para atualizar a configuração de um cluster.

- update-cluster-settings (AWS CLI)

```
aws ecs update-cluster-settings --cluster cluster_name_or_arn --settings
 name=containerInsights,value=enabled|disabled --region us-east-1
```

Exclusão de um cluster

Se você terminou de usar um cluster, poderá excluí-lo. Ao excluir um cluster no console do Amazon ECS, os recursos associados excluídos com ele variarão dependendo de como o cluster foi criado. Step 5 (p. 24) do procedimento a seguir muda com base nessa condição.

Se o cluster foi criado com a experiência de primeira execução do console descrita em Conceitos básicos do Amazon ECS (p. 16) após 24 de novembro de 2015 ou com o assistente de criação de cluster descrito em Criação de um cluster (p. 22), a pilha do AWS CloudFormation criada para o cluster também será excluída quando você excluir o cluster.

Para excluir um cluster

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.

2. Na barra de navegação, selecione a região a ser usada.
3. No painel de navegação, escolha Clusters.
4. Na página Clusters, selecione o cluster a ser excluído.
5. Escolha Delete Cluster. Você verá um prompt de confirmação.

Definições de tarefa do Amazon ECS

Uma definição de tarefa é necessária para executar contêineres do Docker no Amazon ECS. Entre alguns dos parâmetros que você pode especificar em uma definição de tarefa estão:

- A imagem do Docker a ser usada com cada contêiner em sua tarefa
- A CPU e a memória a serem usadas com cada tarefa
- O tipo de inicialização a ser usado, que determina a infraestrutura na qual as tarefas são hospedadas
- O modo de rede do Docker a ser usado para os contêineres na tarefa
- A configuração de registro em log a ser usada para suas tarefas
- Se a tarefa deverá continuar sendo executada se o contêiner for concluído ou falhar
- O comando que o contêiner deve executar quando iniciado
- Eventuais volumes de dados que devem ser usados com os contêineres na tarefa
- A função do IAM que suas tarefas devem usar

A pilha inteira de aplicativos não precisa existir em uma única definição de tarefa, e na maioria dos casos não deve. O aplicativo pode abranger várias definições de tarefa integrando contêineres relacionados às próprias definições de tarefa, cada uma representando um único componente. Para obter mais informações, consulte Arquitetura do aplicativo (p. 29).

Tópicos

- Considerações sobre definição de tarefas (p. 25)
- Arquitetura do aplicativo (p. 29)
- Como criar uma definição de tarefa (p. 29)
- Parâmetros de definição de tarefa (p. 34)
- Tipos de inicialização Amazon ECS (p. 59)
- Como usar volumes de dados em tarefas (p. 61)
- Redes de tarefas com o modo de rede do awsvpc (p. 62)
- Como usar o driver de log awslogs (p. 64)
- Rotear logs personalizados (p. 70)
- Autenticação de registro privado para tarefas (p. 71)
- Especificação de dados confidenciais (p. 73)
- Definições de tarefa de exemplo (p. 79)
- Como atualizar uma definição de tarefa (p. 83)
- Como cancelar o registro das definições de tarefa (p. 83)

Considerações sobre definição de tarefas

As tarefas que usam o tipo de inicialização Fargate não são compatíveis com todos os parâmetros de definição de tarefas do Amazon ECS disponíveis. Alguns parâmetros são totalmente incompatíveis e outros se comportam de maneira diferente nas tarefas do Fargate.

Os seguintes parâmetros de tarefa não são válidos nas tarefas do Fargate:

- `devices`
- `disableNetworking`

- `dnsSearchDomains`
- `dnsServers`
- `dockerSecurityOptions`
- `dockerVolumeConfiguration`
- `extraHosts`
- `host`
- `hostname`
- `links`
- `placementConstraints` — Por padrão, as tarefas Fargate são distribuídas entre as zonas de disponibilidade.
- `privileged`
- `sharedMemorySize`
- `tmpfs`

> **Important**
>
> Quando qualquer parâmetro de definição de tarefa não for compatível, presume-se que quaisquer subsinalizadores para esse parâmetro não sejam compatíveis.

Os seguintes parâmetros de definição de tarefa se comportam de maneira diferente para as tarefas do Fargate:

- Ao usar a `logConfiguration`, o único `logDriver` compatível com as tarefas Fargate é o driver de logs `awslogs`.
- Ao usar `linuxParameters`, para `capabilities`, o parâmetro `drop` pode ser usado, mas o parâmetro `add` não é compatível.
- O parâmetro `healthCheck` é compatível apenas com tarefas Fargate que usam a versão de plataforma 1.1.0 ou posterior.
- Se você usar o parâmetro `portMappings`, deverá especificar somente a `containerPort`. A `hostPort` pode ser deixada em branco ou ser definida para o mesmo valor da `containerPort`.

Para garantir que a sua definição de tarefa seja válida para o tipo de inicialização Fargate, você pode especificar o seguinte ao registrar a definição da tarefa:

- No Console de gerenciamento da AWS, para o campo Requires Compatibilities (Requer compatibilidades), especifique FARGATE.
- Na AWS CLI, para a opção `--requires-compatibilities`, especifique `FARGATE`.
- Na API, especifique o sinalizador `requiresCompatibilities`.

Modo de rede

As definições de tarefa do Fargate exigem que o modo de rede seja definido como `awsvpc`. O modo de rede `awsvpc` fornece uma interface de rede elástica própria para cada tarefa. Também é necessário configurar a rede ao criar o serviço ou executar as tarefas manualmente. Para obter mais informações, consulte Redes de tarefas com o modo de rede do `awsvpc` (p. 62).

CPU e memória da tarefa

As definições de tarefas do Fargate exigem que você especifique a CPU e a memória no nível da tarefa. Embora você também possa especificar a CPU e a memória no nível do contêiner para tarefas do Fargate,

isso é opcional. A maioria dos casos de uso são atendidos com a especificação desses recursos somente no nível de tarefa. A tabela a seguir mostra as combinações válidas para CPU e memória em nível de tarefa.

Valor de CPU	Valor de memória
256 (0,25 vCPU)	0,5 GB, 1 GB, 2 GB
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2048 (2 vCPU)	Entre 4 GB e 16 GB em incrementos de 1 GB
4096 (4 vCPU)	Entre 8 GB e 30 GB em incrementos de 1 GB

Registro

As definições de tarefas do Fargate só oferecem suporte ao driver de log `awslogs` para a configuração de log. Dessa forma, as tarefas do Fargate são configuradas para enviar informações de log para o Amazon CloudWatch Logs. A tabela a seguir mostra um trecho de uma definição de tarefa onde o driver de log awslogs é configurado:

```
"logConfiguration": {
   "logDriver": "awslogs",
   "options": {
      "awslogs-group" : "/ecs/fargate-task-definition",
      "awslogs-region": "us-east-1",
      "awslogs-stream-prefix": "ecs"
}
```

Para obter mais informações sobre como usar o driver de log `awslogs` nas definições de tarefa para enviar os logs de contêiner para o CloudWatch Logs, consulte Como usar o driver de log awslogs (p. 64).

Função do IAM da execução de tarefas do Amazon ECS

Há uma função opcional do IAM para execução de tarefas que você pode especificar com o Fargate para permitir que as tarefas do Fargate façam chamadas de API para o Amazon ECR. As chamadas de API extraem imagens de contêiner, além de chamar o CloudWatch para armazenar os logs do aplicativo do contêiner. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Exemplo de definição de tarefa

Veja a seguir um exemplo de definição de tarefa usando o tipo de inicialização Fargate que configura um servidor web:

```
{
   "containerDefinitions": [
      {
         "command": [
```

```
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div
 style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!
</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></
html>' >  /usr/local/apache2/htdocs/index.html && httpd-foreground\""
         ],
         "entryPoint": [
            "sh",
            "-c"
         ],
         "essential": true,
         "image": "httpd:2.4",
         "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
               "awslogs-group" : "/ecs/fargate-task-definition",
               "awslogs-region": "us-east-1",
               "awslogs-stream-prefix": "ecs"
            }
         },
         "name": "sample-fargate-app",
         "portMappings": [
            {
               "containerPort": 80,
               "hostPort": 80,
               "protocol": "tcp"
            }
         ]
      }
   ],
   "cpu": "256",
   "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
   "family": "fargate-task-definition",
   "memory": "512",
   "networkMode": "awsvpc",
   "requiresCompatibilities": [
       "FARGATE"
    ]
}
```

Armazenamento de tarefas

Quando provisionada, cada tarefa do Fargate recebe o seguinte armazenamento. O armazenamento de tarefas é efêmero. Depois que uma tarefa do Fargate é interrompida, o armazenamento é excluído.

- 10 GB de armazenamento de camadas do Docker
- Mais 4 GB para montagens de volume. Eles podem ser montados e compartilhados entre os contêineres usando os parâmetros `volumes`, `mountPoints` e `volumesFrom` na definição da tarefa.

  Note

  Os parâmetros `host` e `sourcePath` não oferecem suporte a tarefas do Fargate.

Para obter mais informações sobre os limites de serviço padrão do Amazon ECS, consulte Limites de serviço do Amazon ECS (p. 277).

Para fornecer armazenamento vazio não persistente para contêineres em tarefas do Fargate

Neste exemplo, você pode ter dois contêineres de banco de dados que precisam acessar o mesmo local de armazenamento de arquivos temporários durante uma tarefa.

1. Na seção `volumes` da definição da tarefa, defina um volume com o nome `database_scratch`.

```
  "volumes": [ { "name": "database_scratch", "host": {} } ]
```

2. Na seção `containerDefinitions`, crie as definições de contêiner de banco de dados para que elas montem o armazenamento não persistente.

```
  "containerDefinitions": [ { "name": "database1", "image": "my-repo/database",
 "cpu": 100, "memory": 100, "essential": true, "mountPoints": [ { "sourceVolume":
 "database_scratch", "containerPath": "/var/scratch" } ] }, { "name": "database2",
 "image": "my-repo/database", "cpu": 100, "memory": 100, "essential": true,
 "mountPoints": [ { "sourceVolume": "database_scratch", "containerPath": "/var/
scratch" } ] } ]
```

Arquitetura do aplicativo

A forma como você arquiteta seu aplicativo no Amazon ECS depende de vários fatores; o tipo de inicialização que você está usando é um importante diferencial. Oferecemos as seguintes orientações que podem auxiliar no processo.

Uso do tipo de inicialização Fargate

Ao arquitetar seu aplicativo usando o tipo de inicialização Fargate para suas tarefas, a pergunta principal é se você deve colocar vários contêineres na mesma definição de tarefa ou se deve implantar contêineres separadamente em várias definições de tarefa.

Você deverá colocar vários contêineres na mesma definição de tarefa se:

- Os contêineres compartilharem um ciclo de vida em comum (ou seja, eles devem ser iniciados e encerrados juntos).
- Os contêineres precisarem ser executados no mesmo host subjacente (ou seja, um contêiner faz referência ao outro na porta localhost).
- É preferível que seus contêineres compartilhem recursos.
- Os contêineres compartilham volumes de dados.

Caso contrário, você deve definir seus contêineres em definições de tarefas separadas, para que possa escalá-los, provisioná-los e desprovisioná-los separadamente.

Como criar uma definição de tarefa

Para executar contêineres do Docker no Amazon ECS, você deve criar uma definição de tarefa. Você pode definir vários contêineres e volumes de dados em uma definição de tarefa. Para obter mais informações sobre os parâmetros disponíveis em uma definição de tarefa, consulte Parâmetros de definição de tarefa (p. 34).

Para criar uma nova definição de tarefa

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, escolha Task Definitions (Definições de tarefa), Create new Task Definition (Criar nova definição de tarefa).
3. Na página Select launch type compatibilities (Selecionar compatibilidades do tipo de inicialização), escolha FARGATE, Next step (Próxima etapa).

Note

O tipo de inicialização Fargate não é compatível com os contêineres do Windows.

4. (Opcional) Se você tiver uma representação JSON de sua definição de tarefa, conclua as seguintes etapas:

   a. Na página Configure task and container definitions (Configurar tarefa e definições de contêiner), vá até o final da página e escolha Configure via JSON (Configurar via JSON).

   b. Cole o JSON de definição de tarefa na área de texto e escolha Save (Salvar).

   c. Verifique suas informações e escolha Create (Criar).

   Role para o final da página e escolha Configure via JSON (Configurar via JSON).

5. Em Task Definition Name (Nome da definição da tarefa), digite um nome para a sua definição de tarefa. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.

6. Em Task execution IAM role (Função do IAM de execução de tarefas), selecione a função de execução de tarefas ou Create new role (Criar nova função) para que o console crie uma para você. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

7. Em Task size (Tamanho da tarefa), escolha um valor para Task memory (GB) (Memória da tarefa (GB)) e Task CPU (vCPU) (CPU da tarefa (vCPU)). A tabela a seguir mostra as combinações válidas.

Valor de CPU	Valor de memória
256 (0,25 vCPU)	512 MB, 1 GB, 2 GB
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB
2048 (2 vCPU)	Entre 4 GB e 16 GB em incrementos de 1 GB
4096 (4 vCPU)	Entre 8 GB e 30 GB em incrementos de 1 GB

8. Para cada contêiner em sua definição de tarefa, conclua as seguintes etapas:

   a. Selecione Add container (Adicionar contêiner).

   b. Preencha os campos obrigatórios e qualquer campo opcional a ser usado em suas definições de contêiner. Mais parâmetros de definição de contêiner estão disponíveis no menu Advanced container configuration (Configuração de contêiner avançada). Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34).

   c. Selecione Add (Adicionar) para adicionar o contêiner à definição de tarefa.

9. (Opcional) Em Service Integration (Integração de serviço), para configurar os parâmetros para a integração do App Mesh, escolha Enable App Mesh integration (Habilitar a integração do App Mesh) e faça o seguinte:

   a. Em Application container name (Nome do contêiner de aplicativo), escolha o nome do contêiner a ser usado para o aplicativo do App Mesh. Esse contêiner já deve estar definido na definição da tarefa.

   b. Em Envoy image (Imagem do Envoy), use a imagem do contêiner do Envoy preenchida automaticamente, que é 840364872350.dkr.ecr.*us-west-2*.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-prod.

   c. Em Mesh name (Nome da malha), escolha a malha do serviço App Mesh a ser usada. Ela já deve ter sido criada para que seja exibida. Para obter mais informações, consulte Malhas de serviços no Guia do usuário do AWS App Mesh.

d. Em Virtual node name (Nome do nó virtual), escolha o nó virtual do App Mesh a ser usado. Ele já deve ter sido criada para que seja exibido. Para obter mais informações, consulte Nós virtuais no Guia do usuário do AWS App Mesh.

e. Em Virtual node port (Porta do nó virtual), isso será preenchido automaticamente com a porta do listener definida no nó virtual.

f. Escolha Apply (Aplicar), Confirm (Confirmar). Isso criará um novo contêiner do proxy Envoy para a definição de tarefa, assim como as configurações para oferecer suporte a ele. Ele preencherá automaticamente as definições de configuração do proxy App Mesh para a etapa seguinte.

10. (Opcional) Em Proxy Configuration (Configuração do proxy), verifique todos os valores preenchidos automaticamente. Para obter mais informações sobre essas chaves, consulte Proxy Configuration (p. 283).
11. (Opcional) Para definir volumes de dados para sua tarefa, escolha Add volume (Adicionar volume). Para obter mais informações, consulte Como usar volumes de dados em tarefas (p. 61).

    - Em Name (Nome), digite um nome para o volume. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.

12. Na seção Tags, especifique a chave e o valor de cada tag para associá-la à definição da tarefa. Para obter mais informações, consulte Como marcar seus recursos do Amazon ECS.
13. Escolha Create (Criar).

Modelo de definição de tarefa

Um modelo de definição de tarefa vazio é mostrado abaixo. Você pode usar esse modelo para criar a definição de tarefa que pode acabar sendo colada na área de entrada JSON do console ou salva em um arquivo e usada com a opção da AWS CLI `--cli-input-json`. Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34).

```
{
    "family": "",
    "taskRoleArn": "",
    "executionRoleArn": "",
    "networkMode": "none",
    "containerDefinitions": [
        {
            "name": "",
            "image": "",
            "repositoryCredentials": {
                "credentialsParameter": ""
            },
            "cpu": 0,
            "memory": 0,
            "memoryReservation": 0,
            "links": [
                ""
            ],
            "portMappings": [
                {
                    "containerPort": 0,
                    "hostPort": 0,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": [
                ""
            ],
            "command": [
```

```
            ""
        ],
        "environment": [
            {
                "name": "",
                "value": ""
            }
        ],
        "mountPoints": [
            {
                "sourceVolume": "",
                "containerPath": "",
                "readOnly": true
            }
        ],
        "volumesFrom": [
            {
                "sourceContainer": "",
                "readOnly": true
            }
        ],
        "linuxParameters": {
            "capabilities": {
                "add": [
                    ""
                ],
                "drop": [
                    ""
                ]
            },
            "devices": [
                {
                    "hostPath": "",
                    "containerPath": "",
                    "permissions": [
                        "mknod"
                    ]
                }
            ],
            "initProcessEnabled": true,
            "sharedMemorySize": 0,
            "tmpfs": [
                {
                    "containerPath": "",
                    "size": 0,
                    "mountOptions": [
                        ""
                    ]
                }
            ]
        },
        "secrets": [
            {
                "name": "",
                "valueFrom": ""
            }
        ],
        "dependsOn": [
            {
                "containerName": "",
                "condition": "SUCCESS"
            }
        ],
        "startTimeout": 0,
        "stopTimeout": 0,
        "hostname": "",
```

```
            "user": "",
            "workingDirectory": "",
            "disableNetworking": true,
            "privileged": true,
            "readonlyRootFilesystem": true,
            "dnsServers": [
                ""
            ],
            "dnsSearchDomains": [
                ""
            ],
            "extraHosts": [
                {
                    "hostname": "",
                    "ipAddress": ""
                }
            ],
            "dockerSecurityOptions": [
                ""
            ],
            "interactive": true,
            "pseudoTerminal": true,
            "dockerLabels": {
                "KeyName": ""
            },
            "ulimits": [
                {
                    "name": "rss",
                    "softLimit": 0,
                    "hardLimit": 0
                }
            ],
            "logConfiguration": {
                "logDriver": "syslog",
                "options": {
                    "KeyName": ""
                }
            },
            "healthCheck": {
                "command": [
                    ""
                ],
                "interval": 0,
                "timeout": 0,
                "retries": 0,
                "startPeriod": 0
            },
            "systemControls": [
                {
                    "namespace": "",
                    "value": ""
                }
            ],
            "resourceRequirements": [
                {
                    "value": "",
                    "type": "GPU"
                }
            ]
        }
    ],
    "volumes": [
        {
            "name": "",
            "host": {
                "sourcePath": ""
```

```
            },
            "dockerVolumeConfiguration": {
                "scope": "task",
                "autoprovision": true,
                "driver": "",
                "driverOpts": {
                    "KeyName": ""
                },
                "labels": {
                    "KeyName": ""
                }
            }
        }
    ],
    "placementConstraints": [
        {
            "type": "memberOf",
            "expression": ""
        }
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "",
    "memory": "",
    "tags": [
        {
            "key": "",
            "value": ""
        }
    ],
    "pidMode": "host",
    "ipcMode": "host",
    "proxyConfiguration": {
        "type": "APPMESH",
        "containerName": "",
        "properties": [
            {
                "name": "",
                "value": ""
            }
        ]
    }
}
```

É possível gerar esse modelo de definição de tarefa usando o seguinte comando da AWS CLI:

```
aws ecs register-task-definition --generate-cli-skeleton
```

Parâmetros de definição de tarefa

As definições de tarefas são divididas em partes separadas: família de tarefas, a função de tarefas do IAM, o modo de rede, as definições de contêiner, volumes, restrições de posicionamento de tarefas e tipos de inicialização. As definições de contêiner e família são obrigatórias na definição de tarefa, enquanto função de tarefa, modo de rede, volumes, restrições de posicionamento da tarefa e tipo de inicialização.

Veja a seguir as descrições mais detalhadas de cada parâmetro de definição de tarefa.

Família

`family`

Tipo: string

Obrigatório: sim

Ao registrar uma definição de tarefa, você dá a ela uma família, semelhante a um nome para várias versões da definição de tarefa, especificada com um número de revisão. A primeira definição de tarefa registrada em uma determinada família recebe uma revisão 1, e todas as definições de tarefa registradas depois receberão um número de revisão sequencial.

Função de execução de tarefas

`executionRoleArn`

Tipo: string

Obrigatório: não

Ao registrar uma definição de tarefa, é possível fornecer uma função de execução de tarefas que permite que os contêineres da tarefa obtenham imagens de contêiner e publiquem registros de contêiner no CloudWatch em seu nome. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Modo de rede

`networkMode`

Tipo: string

Obrigatório: não

O modo de rede do Docker a ser usado para os contêineres na tarefa. Quando usar o tipo de inicialização Fargate, o modo de rede `awsvpc` será necessário.

Quando o modo de rede for `awsvpc`, será alocada uma interface de rede elástica para a tarefa e você deverá especificar uma `NetworkConfiguration` ao criar um serviço ou executar uma tarefa com a definição da tarefa. Para obter mais informações, consulte Redes de tarefas com o modo de rede do `awsvpc` (p. 62).

O modo de rede `awsvpc` oferece o melhor desempenho de rede para contêineres porque usa a pilha de rede do Amazon EC2. As portas do contêiner expostas são mapeadas diretamente para a porta da interface de rede elástica conectada, portanto, não é possível aproveitar os mapeamentos dinâmicos da porta do host.

Definições de contêiner

Ao registrar uma definição de tarefa, você deve especificar uma lista de definições de contêiner passadas para o daemon do Docker em uma instância de contêiner. Os parâmetros a seguir são permitidos em uma definição de contêiner.

Tópicos

- Parâmetros de definição de contêiner padrão (p. 36)

- Parâmetros de definição de contêiner avançados (p. 39)
- Outros parâmetros de definição de contêiner padrão (p. 49)

Parâmetros de definição de contêiner padrão

Os parâmetros de definição de tarefa a seguir são obrigatórios ou usados na maioria das definições de contêiner.

Tópicos

- Nome (p. 36)
- Imagem (p. 36)
- Memória (p. 37)
- Mapeamentos de porta (p. 37)

Nome

`name`

Tipo: string

Obrigatório: sim

O nome de um contêiner. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados. Caso você esteja vinculando vários contêineres em uma definição de tarefa, o `name` de um contêiner pode ser informado no `links` de outro contêiner para conectar os contêineres.

Imagem

`image`

Tipo: string

Obrigatório: sim

A imagem usada para iniciar um contêiner. Esta string é passada diretamente para o daemon do Docker. As imagens no registro do Docker Hub estão disponíveis por padrão. Você também pode especificar outros repositórios com *`repository-url`*`/`*`image`*`:`*`tag`* ou *`repository-url`*`/`*`image`*`@`*`digest`*. São permitidos até 255 letras (caixa alta e baixa), números, hífens, sublinhados, dois pontos, ponto, barras e sinais numéricos. Este parâmetro é mapeado para `Image` na seção Criar um contêiner do Docker Remote API e o parâmetro `IMAGE` de docker run.

- Quando uma nova tarefa é iniciada, o agente do contêiner do Amazon ECS obtém a versão mais recente da imagem especificada e a tag do contêiner a ser usado. No entanto, as atualizações subsequentes feitas em um repositório de imagens não são propagadas para tarefas já em execução.
- As imagens em registros privados são suportadas. Para obter mais informações, consulte Autenticação de registro privado para tarefas (p. 71).
- As imagens nos repositórios do Amazon ECR podem ser especificadas usando a convenção de nomenclatura `registry/repository:tag` ou `registry/repository@digest`. Por exemplo, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`* ou *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`@`*`sha256:94afd1f2e64d908bc90dbca0035a5b567EXAMPLE`*
- As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).

- As imagens em outros repositórios no Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
- As imagens em outros repositórios on-line são ainda mais qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).

Memória

`memory`

Tipo: inteiro

Obrigatório: não

A quantidade (em MiB) de memória a ser apresentada ao contêiner. Caso tente exceder a memória especificada aqui, o contêiner será excluído. A quantidade total de memória reservada para todos os contêineres dentro da tarefa deve ser menor que o valor da tarefa `memory`, se estiver especificado. Este parâmetro é mapeado para `Memory` na seção Criar um contêiner do Docker Remote API e a opção `--memory` para docker run.

Se estiver usando o tipo de inicialização Fargate, esse parâmetro será opcional.

O daemon do Docker reserva pelo menos 4 MiB de memória para um contêiner, de maneira que você não deve especificar menos de 4 MiB de memória para os contêineres.

`memoryReservation`

Tipo: inteiro

Obrigatório: não

O limite flexível (em MiB) de memória a ser reservado para o contêiner. Quando a memória do sistema está em contenção, o Docker tenta manter a memória do contêiner nesse limite flexível. Porém, o contêiner pode consumir mais memória quando necessário, até o limite fixo especificado com o parâmetro `memory` (se aplicável) ou toda a memória disponível na instância de contêiner, o que ocorrer primeiro. Este parâmetro é mapeado para `MemoryReservation` na seção Criar um contêiner do Docker Remote API e a opção `--memory-reservation` para docker run.

Se um valor de memória no nível de tarefa não for especificado, será necessário especificar um número inteiro diferente de zero para um ou ambos `memory` ou `memoryReservation` em uma definição de contêiner. Caso você especifique ambos, `memory` deve ser maior que `memoryReservation`. Caso você especifique `memoryReservation`, o valor é subtraído dos recursos de memória disponíveis para a instância de contêiner na qual o contêiner está colocado. Caso contrário, o valor de `memory` é usado.

Por exemplo, caso o contêiner normalmente use 128 MiB de memória, mas às vezes chega a 256 MiB de memória em períodos curtos, você pode definir um `memoryReservation` de 128 MiB e um limite fixo `memory` de 300 MiB. Essa configuração permitiria que contêiner reservasse apenas 128 MiB de memória dos recursos restantes na instância de contêiner, mas também permite que o contêiner consuma mais recursos de memória quando necessário.

O daemon do Docker reserva pelo menos 4 MiB de memória para um contêiner, de maneira que você não deve especificar menos de 4 MiB de memória para os contêineres.

Mapeamentos de porta

`portMappings`

Tipo: matriz de objeto

Obrigatório: não

Os mapeamentos de porta permitem que os contêineres acessem portas na instância de contêiner host para enviar ou receber tráfego.

Para definições de tarefa que usam o modo de rede `awsvpc`, você só deve especificar `containerPort`. A `hostPort` pode ser deixada em branco ou ser o mesmo valor de `containerPort`.

Este parâmetro é mapeado para `PortBindings` na seção Criar um contêiner do Docker Remote API e a opção `--publish` para docker run. Caso o modo de rede de uma definição de tarefa seja definido como `host`, as portas host devem ser indefinidas ou corresponder à porta de contêiner no mapeamento de porta.

> **Note**
>
> Quando a tarefa obter o status `RUNNING`, as atribuições automáticas e manuais da porta do contêiner e do host estarão visíveis nos seguintes locais:
>
> - Console: a seção Network Bindings (Associações de rede) de uma descrição de contêiner para uma tarefa selecionada.
> - AWS CLI: a seção `networkBindings` do resultado do comando describe-tasks.
> - API: a resposta `DescribeTasks`.

`containerPort`

Tipo: inteiro

Obrigatório: sim, quando `portMappings` são usados

O número da porta no contêiner vinculado à porta host atribuída automaticamente ou especificada pelo usuário.

Se estiver usando contêineres em uma tarefa com o tipo de inicialização Fargate, as portas expostas deverão ser especificadas usando `containerPort`.

Se estiver usando contêineres em uma tarefa com o tipo de inicialização EC2 e especificar uma porta de contêiner e não uma porta de host, seus contêineres receberão uma porta de host no intervalo de portas efêmeras. Para obter mais informações, consulte `hostPort`. Os mapeamentos de porta atribuídos automaticamente dessa maneira não contam para o limite de 100 portas reservadas de uma instância de contêiner.

> **Important**
>
> Você não pode expor a mesma porta de contêiner para vários protocolos. Um erro será retornado se for feita essa tentativa.

`hostPort`

Tipo: inteiro

Obrigatório: não

O número da porta na instância de contêiner a ser reservado para o contêiner.

Se você estiver usando contêineres em uma tarefa com o tipo de inicialização Fargate, `hostPort` poderá ser deixado em branco ou ser o mesmo valor que `containerPort`.

Se estiver usando contêineres em uma tarefa com o tipo de inicialização EC2, você poderá especificar uma porta de host não reservada para o mapeamento da porta do contêiner (conhecido como mapeamento estático da porta do host) ou omitir a `hostPort` (ou defini-la como `0`) enquanto especifica um `containerPort`. O contêiner receberá automaticamente uma porta

(conhecido como mapeamento dinâmico da porta do host) no intervalo de portas efêmero para o sistema operacional da instância do contêiner e a versão do Docker.

O intervalo de portas efêmero padrão para a versão 1.6.0 e posterior do Docker é listado na instância em `/proc/sys/net/ipv4/ip_local_port_range`. Se esse parâmetro de kernel estiver indisponível, o intervalo de portas efêmero padrão de `49153-65535` será usado. Não tente especificar uma porta do host no intervalo de portas efêmero, pois elas estão reservadas para atribuição automática. Em geral, as portas abaixo de `32768` estão fora do intervalo de portas efêmero.

As portas reservadas padrão são `22` para SSH, as portas do Docker `2375` e `2376` e as portas do agente de contêiner do Amazon ECS `51678-51680`. Qualquer porta host que tenha sido especificada pelo usuário anteriormente para uma tarefa em execução também é reservada enquanto a tarefa está em execução (depois que uma tarefa para, a porta host é liberada). As portas reservadas são exibidas na janela `remainingResources` da saída de describe-container-instances, e uma instância de contêiner pode ter até 100 portas reservadas de cada vez, incluindo as portas reservadas padrão. As portas atribuídas automaticamente não contam em relação ao limite de 100 portas reservadas.

`protocol`

Tipo: string

Obrigatório: não

O protocolo usado no mapeamento da porta. Os valores válidos são `tcp` e `udp`. O padrão é `tcp`.

Caso você esteja especificando uma porta host, use a seguinte sintaxe:

```
"portMappings": [
    {
        "containerPort": integer,
        "hostPort": integer
    }
    ...
]
```

Caso você queira uma porta host atribuída automaticamente, use a seguinte sintaxe:

```
"portMappings": [
    {
        "containerPort": integer
    }
    ...
]
```

Parâmetros de definição de contêiner avançados

Os parâmetros de definição de contêiner avançados a seguir fornecem recursos estendidos ao comando docker run usado para ativar contêineres nas instâncias de contêiner do Amazon ECS.

Tópicos

- Verificação de integridade (p. 40)
- Ambiente (p. 41)
- Configurações de rede (p. 44)
- Armazenamento e registro em log (p. 44)
- Segurança (p. 48)

- Limites de recurso (p. 48)
- Rótulos do Docker (p. 49)

Verificação de integridade

`healthCheck`

O comando da verificação de integridade e os parâmetros de configuração associados para o contêiner. Este parâmetro é mapeado para `HealthCheck` na seção Criar um contêiner do Docker Remote API e o parâmetro `HEALTHCHECK` do docker run.

> **Note**
>
> O agente de contêiner do Amazon ECS só monitora e gera relatórios das verificações de integridade especificadas na definição da tarefa. O Amazon ECS não monitora verificações de integridade do Docker integradas a uma imagem de contêiner e não especificadas na definição de contêiner. Os parâmetros de verificação de integridade especificados em uma definição de contêiner substituem as verificações de integridade do Docker existentes na imagem do contêiner.

A integridade da tarefa é comunicada por seu `healthStatus`, que é determinado pela integridade dos principais contêineres na tarefa. Se todos os contêineres essenciais da tarefa forem comunicados como `HEALTHY`, o status dela também será comunicado como `HEALTHY`. Se quaisquer contêineres essenciais da tarefa forem comunicados como `UNHEALTHY` ou `UNKNOWN`, o status dela também será comunicado como `UNHEALTHY` ou `UNKNOWN`. Se for comunicada como não íntegra, a tarefa de um serviço será removida dele e substituída.

Veja a seguir observações sobre a compatibilidade de verificação de integridade do contêiner:

- As verificações de integridade do contêiner serão compatíveis com tarefas de Fargate se você estiver usando a versão 1.1.0 ou posterior da plataforma. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).
- As verificações de integridade do contêiner não são compatíveis com tarefas que fazem parte de um serviço configurado para usar um Classic Load Balancer.

`command`

Uma matriz de strings que representa o comando executado pelo contêiner para determinar se ela está íntegra. A matriz de strings pode começar com `CMD` para executar os argumentos de comando diretamente ou `CMD-SHELL` para executar o comando com o shell padrão do contêiner. Se nenhum for especificado, `CMD` é usado como padrão.

Ao registrar uma definição de tarefa no Console de gerenciamento da AWS, use uma lista de comandos separados por vírgulas que será automaticamente convertida em uma string após a criação da definição de tarefa. Uma entrada de exemplo para uma verificação de integridade pode ser:

```
CMD-SHELL, curl -f http://localhost/ || exit 1
```

Ao registrar uma definição de tarefa usando o painel JSON do Console de gerenciamento da AWS, a AWS CLI ou as APIs, você deve estabelecer a lista de comandos entre colchetes. Uma entrada de exemplo para uma verificação de integridade pode ser:

```
[ "CMD-SHELL", "curl -f http://localhost/ || exit 1" ]
```

Um código de saída 0 indica sucesso, e um código de saída diferente de zero indica falha. Para obter mais informações, consulte `HealthCheck` na seção Criar um contêiner do Docker Remote API.

`interval`

O período em segundos entre cada execução de verificação de integridade. É possível especificar entre 5 e 300 segundos. O valor de padrão é de 30 segundos.

`timeout`

O período de espera em segundos para que uma verificação de integridade seja bem-sucedida antes de ser considerada uma falha. É possível especificar entre 2 e 60 segundos. O valor de padrão é de 5 segundos.

`retries`

O número de novas tentativas de uma verificação de integridade com falha até o contêiner ser considerado não íntegro. É possível especificar entre 1 e 10 novas tentativas. O valor padrão é três novas tentativas.

`startPeriod`

O período de carência opcional para que os contêineres possam inicializar antes de as verificações de integridade com falha serem contabilizadas no número máximo de novas tentativas. É possível especificar entre 0 e 300 segundos. O `startPeriod` é desabilitado por padrão.

Ambiente

`cpu`

Tipo: inteiro

Obrigatório: não

O número de unidades de `cpu` que o agente de contêiner do Amazon ECS reservará para o contêiner. Este parâmetro é mapeado para `CpuShares` na seção Criar um contêiner do Docker Remote API e a opção `--cpu-shares` para docker run.

Esse campo é opcional para tarefas que usam o tipo de inicialização Fargate e a única exigência é que a quantidade total de CPU reservada para todos os contêineres dentro da tarefa seja inferior ao valor `cpu` em nível de tarefa.

Note

É possível determinar o número de unidades de CPU disponíveis por tipo de instância do Amazon EC2 multiplicando o número de vCPUs indicadas para esse tipo de instância na página de detalhes Instâncias do Amazon EC2 por 1.024.

Os contêineres do Linux compartilham unidades de CPU não alocadas com outros contêineres na instância de contêiner que tem a mesma proporção que a respectiva quantidade alocada. Por exemplo, caso você execute uma tarefa de contêiner único em um tipo de instância de núcleo único com 512 unidades de CPU especificadas para esse contêiner e essa seja a única tarefa em execução na instância de contêiner, o contêiner pode usar todo o compartilhamento de 1.024 unidades de CPU a qualquer momento. Porém, se você tivesse iniciado outra cópia da mesma tarefa nessa instância de contêiner, cada tarefa teria garantidas pelo menos 512 unidades de CPU quando necessário, e cada contêiner poderia flutuar para um uso de CPU mais alto se o outro contêiner não o estivesse usando, mas se ambas as tarefas estivessem 100% ativas o tempo todo, e elas estariam limitadas a 512 unidades de CPU.

Nas instâncias de contêiner do Linux, o daemon do Docker na instância de contêiner usa o valor de CPU para calcular as proporções de compartilhamento de CPU para contêineres em execução. Para obter mais informações, consulte CPU share constraint na documentação do Docker. O valor mínimo válido de compartilhamento da CPU permitido pelo kernel do Linux é 2. No entanto, o parâmetro

de CPU não é necessário, e é possível usar valores de CPU abaixo de 2 em suas definições de contêiner. Para valores de CPU abaixo de 2 (inclusive nulo), o comportamento varia com base na versão do agente de contêiner do Amazon ECS:

- Versões de agente <= 1.1.0: os valores de CPU nulo e zero são passados para o Docker como 0, que o Docker acaba convertendo em 1.024 compartilhamentos de CPU. Os valores de CPU de 1 são passados para o Docker como 1, que o kernel do Linux converte para dois compartilhamentos de CPU.
- Versões de agente >= 1.2.0: os valores nulo, zero e de CPU de 1 são passados para o Docker como dois compartilhamentos de CPU.

Nas instâncias de contêiner do Windows, o limite de CPU é imposto como um limite absoluto ou uma cota. Os contêineres do Windows só têm acesso à quantidade de CPU especificada na definição de tarefa.

`essential`

Tipo: booliano

Obrigatório: não

Caso o parâmetro `essential` de um contêiner esteja marcado como `true` e esse contêiner falhe ou pare por algum motivo, todos os outros contêineres que fazem parte da tarefa são parados. Caso o parâmetro `essential` de um contêiner esteja marcado como `false`, a falha não afeta o restante dos contêineres em uma tarefa. Caso esse parâmetro seja omitido, um contêiner pressupõe-se que um contêiner seja essencial.

Todas as tarefas devem ter pelo menos um contêiner essencial. Caso tenha um aplicativo composto de vários contêineres, você deve agrupar contêineres usados com uma finalidade em comum em componentes e separar os componentes diferentes em várias definições de tarefa. Para obter mais informações, consulte Arquitetura do aplicativo (p. 29).

```
"essential": true|false
```

`entryPoint`

> **Important**
>
> As versões anteriores do agente de contêiner do Amazon ECS não lidam corretamente com parâmetros `entryPoint`. Caso você enfrente problemas para usar `entryPoint`, atualize o agente de contêiner ou informe os comandos e os argumentos como itens de matriz `command` em vez disso.

Tipo: matriz de strings

Obrigatório: não

O ponto de entrada passado para o contêiner. Este parâmetro é mapeado para `Entrypoint` na seção Criar um contêiner do Docker Remote API e a opção `--entrypoint` para docker run. Para obter mais informações sobre o parâmetro `ENTRYPOINT` do Docker, vá até https://docs.docker.com/engine/reference/builder/#entrypoint.

```
"entryPoint": ["string", ...]
```

`command`

Tipo: matriz de strings

Obrigatório: não

O comando passado para o contêiner. Este parâmetro é mapeado para `Cmd` na seção Criar um contêiner do Docker Remote API e o parâmetro `COMMAND` para docker run. Para obter mais informações sobre o parâmetro `CMD` do Docker, vá até https://docs.docker.com/engine/reference/builder/#cmd. Se houver vários argumentos, cada argumento deverá ser uma string separada na matriz.

```
"command": ["string", ...]
```

`workingDirectory`

Tipo: string

Obrigatório: não

O diretório de trabalho no qual executar comandos dentro do contêiner. Este parâmetro é mapeado para `WorkingDir` na seção Criar um contêiner do Docker Remote API e a opção `--workdir` para docker run.

```
"workingDirectory": "string"
```

`environment`

Tipo: matriz de objeto

Obrigatório: não

As variáveis de ambiente a serem passadas para um contêiner. Este parâmetro é mapeado para `Env` na seção Criar um contêiner do Docker Remote API e a opção `--env` para docker run.

> **Important**
>
> Não recomendamos o uso de variáveis de ambiente de texto simples para informações confidenciais, tais como dados de credenciais.

`name`

Tipo: string

Obrigatório: sim, quando `environment` for usado

O nome da variável de ambiente.

`value`

Tipo: string

Obrigatório: sim, quando `environment` for usado

O valor da variável de ambiente.

```
"environment" : [
    { "name" : "string", "value" : "string" },
    { "name" : "string", "value" : "string" }
]
```

`secrets`

Tipo: matriz de objeto

Exigido: Não

Um objeto que representa o segredo a ser exposto ao seu contêiner. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

`name`

Tipo: string

Obrigatório: sim

O valor a ser definido como a variável de ambiente no contêiner.

`valueFrom`

Tipo: string

Obrigatório: sim

O segredo a ser exposto ao contêiner. Os valores com suporte são o ARN completo do segredo do AWS Secrets Manager ou o ARN completo do parâmetro no Parameter Store do AWS Systems Manager.

Note

Se o parâmetro do Store parameter do Systems Manager existir na mesma região da tarefa que você está iniciando, você poderá usar o ARN completo ou o nome do segredo. Se o parâmetro existir em uma região diferente, o ARN completo deverá ser especificado.

```
"secrets": [
    {
        "name": "environment_variable_name",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }
]
```

Configurações de rede

`dnsServers`

Tipo: matriz de strings

Obrigatório: não

Uma lista de servidores DNS apresentados ao contêiner. Este parâmetro é mapeado para `Dns` na seção Criar um contêiner da Docker Remote API e a opção `--dns` para docker run.

Note

Esse parâmetro não oferece suporte para contêineres do Windows ou tarefas que usam o modo de rede `awsvpc`.

```
"dnsServers": ["string", ...]
```

Armazenamento e registro em log

`readonlyRootFilesystem`

Tipo: booliano

Obrigatório: não

Quando esse parâmetro é verdadeiro, o contêiner recebe acesso somente leitura ao sistema de arquivos raiz. Este parâmetro é mapeado para `ReadonlyRootfs` na seção Criar um contêiner do Docker Remote API e a opção `--read-only` para docker run.

Note

Este parâmetro não é compatível com contêineres do Windows.

```
"readonlyRootFilesystem": true|false
```

`mountPoints`

Type: Object

Required: No

The mount points for data volumes in your container.

This parameter maps to `Volumes` in the Create a container section of the Docker Remote API and the `--volume` option to docker run.

Windows containers can mount whole directories on the same drive as `$env:ProgramData`. Windows containers cannot mount directories on a different drive, and mount point cannot be across drives.

`sourceVolume`

Type: String

Required: Yes, when `mountPoints` are used

The name of the volume to mount.

`containerPath`

Type: String

Required: Yes, when `mountPoints` are used

The path on the container to mount the volume at.

`readOnly`

Type: Boolean

Required: No

If this value is `true`, the container has read-only access to the volume. If this value is `false`, then the container can write to the volume. The default value is `false`.

`volumesFrom`

Tipo: matriz de objeto

Obrigatório: não

Volumes de dados a serem montados de outro contêiner. Este parâmetro é mapeado para `VolumesFrom` na seção Criar um contêiner do Docker Remote API e a opção `--volumes-from` para docker run.

`sourceContainer`

Tipo: string

Obrigatório: sim, quando `volumesFrom` for usado

O nome do contêiner com base no qual montar volumes.

`readOnly`

Tipo: booliano

Obrigatório: não

Caso o valor seja `true`, o contêiner tem acesso somente leitura ao volume. Caso esse valor seja `false`, o contêiner pode gravar no volume. O valor padrão é `false`.

```
"volumesFrom": [
                {
                  "sourceContainer": "string",
                  "readOnly": true|false
                }
              ]
```

`logConfiguration`

Tipo: objeto LogConfiguration

Obrigatório: não

A especificação de configuração do log para o contêiner.

Para obter definições de tarefa de exemplo que usam uma configuração de log, consulte Definições de tarefa de exemplo (p. 79).

Este parâmetro é mapeado para `LogConfig` na seção Criar um contêiner da Docker Remote API e a opção `--log-driver` para `docker run`. Por padrão, os contêineres usam o mesmo driver de registro em log usado pelo daemon do Docker. Porém, o contêiner pode usar um driver de registro em log diferente do daemon do Docker especificando um driver de log com esse parâmetro na definição de contêiner. Para usar um driver de registro em log diferente para um contêiner, o sistema de log deve ser configurado corretamente na instância de contêiner (ou em um servidor de log diferente para opções de registro em log remotas). Para obter mais informações sobre as opções para drivers de log diferentes com suporte, consulte Configurar drivers de registro em log na documentação do Docker.

Deve-se observar o seguinte ao especificar uma configuração de log para seus contêineres:

- No momento, o Amazon ECS oferece suporte a um subconjunto de drivers de registro em log disponíveis para o daemon do Docker (mostrado nos valores válidos abaixo). Drivers de log adicionais podem estar disponíveis em versões futuras do agente de contêiner do Amazon ECS.
- Este parâmetro requer a versão 1.18 da API remota do Docker ou posterior em sua instância de contêiner.
- Para tarefas usando o tipo de inicialização do Fargate, como você não tem acesso à infraestrutura subjacente na qual suas tarefas estão hospedadas, qualquer software adicional necessário terá que ser instalado fora da tarefa. Por exemplo, os agregadores de saída Fluentd ou um host remoto executando o Logstash para o qual enviar logs Gelf.

```
"logConfiguration": {
      "logDriver": "awslogs","fluentd","gelf","json-
file","journald","logentries","splunk","syslog","awsfirelens",
      "options": {"string": "string"
        ...},
 "secretOptions": [{
  "name": "string",
  "valueFrom": "string"
 }]
}
```

`logDriver`

Tipo: string

Valores válidos: `"awslogs","fluentd","gelf","json-file","journald","logentries","splunk","syslog","awsfirelens`

Obrigatório: sim, quando `logConfiguration` for usado

O driver de log a ser usado para o contêiner. Os valores válidos listados acima são drivers de log com os quais o agente de contêiner do Amazon ECS pode se comunicar por padrão.

Para tarefas que usam o tipo de inicialização Fargate, os drivers de log compatíveis são `awslogs`, `splunk` e `awsfirelens`.

Para obter mais informações sobre como usar o driver de log `awslogs` em definições de tarefa a fim de enviar os logs de contêiner para o CloudWatch Logs, consulte Como usar o driver de log awslogs (p. 64).

Para obter mais informações sobre como usar o driver de log `awsfirelens`, consulte Roteamento de logs personalizados.

Este parâmetro requer que a versão 1.18 da API remota do Docker ou posterior em sua instância de contêiner.

`options`

Tipo: mapa string para string

Obrigatório: não

As opções de configuração a serem enviadas para o driver de log.

Este parâmetro requer que a versão 1.19 da API remota do Docker ou posterior em sua instância de contêiner.

`secretOptions`

Tipo: matriz de objeto

Obrigatório: não

Um objeto que representa o segredo a ser passado para a configuração de log. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

`name`

Tipo: string

Obrigatório: sim

O valor a ser definido como a variável de ambiente no contêiner.

`valueFrom`

Tipo: string

Obrigatório: sim

O segredo a ser exposto à configuração de log do contêiner.

```
"logConfiguration": {
 "logDriver": "splunk",
 "options": {
```

```
  "splunk-url": "https://cloud.splunk.com:8080",
  "splunk-token": "...",
  "tag": "...",
  ...
 },
 "secretOptions": [{
  "name": "splunk-token",
  "valueFrom": "/ecs/logconfig/splunkcred"
 }]
}
```

Segurança

`user`

Tipo: string

Obrigatório: não

O nome de usuário a ser usado dentro do contêiner. Este parâmetro é mapeado para `User` na seção Criar um contêiner do Docker Remote API e a opção `--user` para docker run.

Você pode usar os formatos a seguir. Se especificar um UID ou um GID, você deverá especificá-lo como um número inteiro positivo.

- `user`
- `user:group`
- `uid`
- `uid:gid`
- `user:gid`
- `uid:group`

> **Note**
>
> Este parâmetro não é compatível com contêineres do Windows.

```
"user": "string"
```

Limites de recurso

`ulimits`

Tipo: matriz de objeto

Obrigatório: não

Uma lista de `ulimits` a ser definida no contêiner. Este parâmetro é mapeado para `Ulimits` na seção Criar um contêiner do Docker Remote API e a opção `--ulimit` para docker run.

Este parâmetro requer que a versão 1.18 da API remota do Docker ou posterior em sua instância de contêiner.

> **Note**
>
> Este parâmetro não é compatível com contêineres do Windows.

```
"ulimits": [
```

```
      {
        "name":
 "core"|"cpu"|"data"|"fsize"|"locks"|"memlock"|"msgqueue"|"nice"|"nofile"|"nproc"|"rss"|"rtprio"|"r
        "softLimit": integer,
        "hardLimit": integer
      }
      ...
    ]
```

`name`

Tipo: string

Valores válidos: `"core" | "cpu" | "data" | "fsize" | "locks" | "memlock" | "msgqueue" | "nice" | "nofile" | "nproc" | "rss" | "rtprio" | "rttime" | "sigpending" | "stack"`

Obrigatório: sim, quando `ulimits` são usados

O `type` do `ulimit`.

`hardLimit`

Tipo: inteiro

Obrigatório: sim, quando `ulimits` são usados

O limite rígido do tipo `ulimit`.

`softLimit`

Tipo: inteiro

Obrigatório: sim, quando `ulimits` são usados

O limite flexível do tipo `ulimit`.

Rótulos do Docker

`dockerLabels`

Tipo: mapa string para string

Obrigatório: não

Um mapa de chave/valor de rótulos a ser adicionado ao contêiner. Este parâmetro é mapeado para `Labels` na seção Criar um contêiner do Docker Remote API e a opção `--label` para docker run.

Este parâmetro requer que a versão 1.18 da API remota do Docker ou posterior em sua instância de contêiner.

```
"dockerLabels": {"string": "string"
      ...}
```

Outros parâmetros de definição de contêiner padrão

Os parâmetros de definição de contêiner a seguir podem ser usados durante o registro das definições de tarefa no console do Amazon ECS usando a opção Configure via JSON (Configurar via JSON). Para obter mais informações, consulte Como criar uma definição de tarefa (p. 29).

Tópicos

- Parâmetros do Linux (p. 50)
- Dependência de contêiner (p. 51)
- Tempos limite de contêiner (p. 52)
- Controles do sistema (p. 52)
- Interativo (p. 53)
- Pseudoterminal (p. 53)

Parâmetros do Linux

`linuxParameters`

Tipo: objeto LinuxParameters

Obrigatório: não

Opções específicas do Linux que são aplicadas ao contêiner, como KernelCapabilities.

Note

Este parâmetro não é compatível com contêineres do Windows.

```
"linuxParameters": {
      "capabilities": {
        "add": ["string", ...],
        "drop": ["string", ...]
        }
      }
```

`capabilities`

Tipo: objeto KernelCapabilities

Obrigatório: não

Os recursos do Linux para o contêiner que são à configuração padrão fornecida pelo Docker ou descartados dela. Para obter mais informações sobre os recursos padrão e não padrão disponíveis, consulte Privilégio de tempo de execução e recursos do Linux na Referência de execução do Docker. Para obter informações mais detalhadas sobre esses recursos do Linux, consulte a página do manual do Linux recursos(7).

`drop`

Tipo: matriz de strings

Valores válidos: `"ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"`

Obrigatório: não

Os recursos do Linux para o contêiner que são removidos da configuração padrão fornecida pelo Docker. Este parâmetro é mapeado para `CapDrop` na seção Criar um contêiner do Docker Remote API e a opção `--cap-drop` para docker run.

`initProcessEnabled`

Execute um processo `init` dentro do contêiner que encaminha sinais e colhe processos. Esse parâmetro mapeia para a opção `--init` para docker run.

Este parâmetro requer a versão 1.25 ou posterior da API remota do Docker em sua instância de contêiner.

Dependência de contêiner

`dependsOn`

Tipo: matriz de objetos ContainerDependency

Obrigatório: não

As dependências definidas para inicialização e desligamento do contêiner. Um contêiner pode conter várias dependências. Quando uma dependência é definida para a inicialização do contêiner, ela é revertida para o desligamento do contêiner. Para ver um exemplo, consulte Exemplo: dependência de contêiner (p. 82).

Para as tarefas que usam o tipo de inicialização Fargate, esse parâmetro exige que a tarefa ou o serviço use a plataforma versão 1.3.0 ou posterior.

```
"dependsOn": [
    {
        "containerName": "string",
        "condition": "string"
    }
]
```

`containerName`

Tipo: string

Obrigatório: sim

O nome do contêiner que deve atender à condição especificada.

`condition`

Tipo: string

Obrigatório: sim

A condição de dependência do contêiner. A seguir estão as condições disponíveis e seus comportamentos:

- `START` – essa condição emula o comportamento de links e volumes hoje. Ela valida que um contêiner dependente seja iniciado antes de permitir que outros contêineres sejam iniciados.
- `COMPLETE` – essa condição valida que um contêiner dependente seja executado até a conclusão (encerramento) antes de permitir que outros contêineres sejam iniciados. Isso pode ser útil para os contêineres não essenciais que executam um script e depois são encerrados.
- `SUCCESS` – essa condição é igual à `COMPLETE`, mas também exige que o contêiner seja encerrado com um status `zero`.
- `HEALTHY` – essa condição valida que o contêiner dependente passe sua verificação de integridade do Docker antes de permitir que outros contêineres sejam iniciados. Isso requer que

o contêiner dependente tenha as verificações de integridade configuradas. Essa condição é confirmada apenas na inicialização da tarefa.

Tempos limite de contêiner

`startTimeout`

Tipo: inteiro

Obrigatório: não

Valores de exemplo: `120`

Tempo a ser aguardado (em segundos) antes de desistir de resolver dependências para um contêiner. Por exemplo, você especifica dois contêineres em uma definição de tarefa com o containerA tendo uma dependência de o containerB atingir um status `COMPLETE`, `HEALTHY` ou `SUCCESS`. Se um valor `startTimeout` for especificado para o containerB e ele não atingir o status desejado dentro desse tempo, o containerA desistirá e não será iniciado. Isso resulta na transição da tarefa para um estado `STOPPED`.

Para as tarefas que usam o tipo de inicialização Fargate, esse parâmetro exige que a tarefa ou o serviço use a plataforma versão 1.3.0 ou posterior.

`stopTimeout`

Tipo: inteiro

Obrigatório: não

Valores de exemplo: `120`

Período (em segundos) a ser aguardado antes de o contêiner ser eliminado de maneira forçada se não for encerrado normalmente por conta própria. Para as tarefas que usam o tipo de inicialização Fargate, o valor do tempo limite máximo de interrupção é `120` segundos. Para as tarefas que usam o tipo de inicialização Fargate, esse parâmetro exige que a tarefa ou o serviço use a plataforma versão 1.3.0 ou posterior.

Controles do sistema

`systemControls`

Tipo: objeto SystemControl

Obrigatório: não

Uma lista de parâmetros de kernel com namespace a ser definida no contêiner. Este parâmetro é mapeado para `Sysctls` na seção Criar um contêiner do Docker Remote API e a opção `--sysctl` para docker run.

Não é recomendável especificar parâmetros `systemControls` relacionados à rede para vários contêineres em uma única tarefa que também usa o modo de rede `awsvpc` ou `host` pelos seguintes motivos:

- Para tarefas que usam o modo de rede `awsvpc`, se você definir `systemControls` para qualquer contêiner, ele se aplicará a todos os contêineres na tarefa. Se você definir um `systemControls` diferente para vários contêineres em uma única tarefa, o contêiner iniciado por último determinará que `systemControls` entre em vigor.
- Para tarefas que usam o modo de rede `host`, o namespace de rede `systemControls` não é compatível.

Se você estiver configurando um namespace de recurso IPC para usar nos contêineres da tarefa, os itens a seguir serão aplicados aos controles do sistema. Para obter mais informações, consulte Modo IPC (p. 58).

- Para tarefas que usam o modo IPC `host`, o namespace IPC `systemControls` não é compatível.
- Para tarefas que usam o modo IPC de `task`, os valores de `systemControls` do namespace IPC serão aplicados a todos os contêineres em uma tarefa.

  Note

  Não há suporte para esse parâmetro para os contêineres do Windows ou tarefas que usam o tipo de inicialização Fargate.

```
"systemControls": [
    {
         "namespace":"string",
         "value":"string"
    }
]
```

`namespace`

Tipo: string

Obrigatório: não

O parâmetro de kernel com namespace para definição de um `value`.

Valores de namespace IPC válidos: `"kernel.msgmax" | "kernel.msgmnb" | "kernel.msgmni" | "kernel.sem" | "kernel.shmall" | "kernel.shmmax" | "kernel.shmmni" | "kernel.shm_rmid_forced"`, bem como Sysctls começando com `"fs.mqueue.*"`

Valores de namespace de rede válidos: Sysctls começando com `"net.*"`

`value`

Tipo: string

Obrigatório: não

O valor do parâmetro de kernel do namespace especificado em `namespace`.

Interativo

`interactive`

Tipo: booliano

Obrigatório: não

Quando esse parâmetro é `true`, isso permite implantar aplicativos em contêineres que exigem a alocação de stdin ou um tty. Este parâmetro é mapeado para `OpenStdin` na seção Criar um contêiner do Docker Remote API e a opção `--interactive` para docker run.

Pseudoterminal

`pseudoTerminal`

Tipo: booliano

Obrigatório: não

Quando esse parâmetro é `true`, um TTY é alocado. Este parâmetro é mapeado para `Tty` na seção Criar um contêiner do Docker Remote API e a opção `--tty` para docker run.

Volumes

Se preferir, ao registrar uma definição de tarefa, você poderá especificar uma lista de volumes a serem passados para o daemon do Docker em uma instância de contêiner, que se tornará disponível para o acesso de outros contêineres da mesma instância de contêiner.

Para obter mais informações, consulte Como usar volumes de dados em tarefas (p. 61).

Os parâmetros a seguir são permitidos em uma definição de contêiner:

`name`

Tipo: sequência

Exigido: Não

O nome do volume. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados. Este nome é referenciado no parâmetro `sourceVolume` da definição de contêiner `mountPoints`.

`host`

Required: No

This parameter is specified when using bind mounts. To use Docker volumes, specify a `dockerVolumeConfiguration` instead. The contents of the `host` parameter determine whether your bind mount data volume persists on the host container instance and where it is stored. If the `host` parameter is empty, then the Docker daemon assigns a host path for your data volume, but the data is not guaranteed to persist after the containers associated with it stop running.

Bind mount host volumes are supported when using either the EC2 or Fargate launch types.

Windows containers can mount whole directories on the same drive as `$env:ProgramData`.

`sourcePath`

Type: String

Required: No

When the `host` parameter is used, specify a `sourcePath` to declare the path on the host container instance that is presented to the container. If this parameter is empty, then the Docker daemon has assigned a host path for you. If the `host` parameter contains a `sourcePath` file location, then the data volume persists at the specified location on the host container instance until you delete it manually. If the `sourcePath` value does not exist on the host container instance, the Docker daemon creates it. If the location does exist, the contents of the source path folder are exported.

Tipos de inicialização

Ao registrar uma definição de tarefa, você especifica o tipo de inicialização a ser usado para ela. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).

O seguinte parâmetro é permitido em uma definição de tarefa:

`requiresCompatibilities`

Tipo: matriz de strings

Obrigatório: não

Valores válidos: `EC2` | `FARGATE`

O tipo de inicialização que a tarefa está usando. Isso permite uma verificação para garantir que todos os parâmetros usados na definição de tarefa atendam aos requisitos do tipo de inicialização.

Os valores válidos são `FARGATE` e `EC2`. Para obter mais informações sobre tipos de inicialização, consulte Tipos de inicialização Amazon ECS (p. 59).

Tamanho da tarefa

Ao registrar uma definição de tarefa, você pode especificar o total de CPU e memória usados para a tarefa. Isso é separado dos valores de `cpu` e `memory` no nível de definição de contêiner. Se estiver usando o tipo de inicialização EC2, esses campos serão opcionais. Se você estiver usando o tipo de inicialização Fargate, esses campos serão obrigatórios e haverá valores específicos para `cpu` e `memory` que sejam compatíveis.

Note

Os parâmetros de CPU e memória em nível de tarefa são ignorados para contêineres do Windows. É recomendável especificar recursos em nível de contêiner para contêineres do Windows.

O seguinte parâmetro é permitido em uma definição de tarefa:

`cpu`

Tipo: string

Obrigatório: não

Note

Este parâmetro não é compatível com contêineres do Windows.

O limite rígido de unidades de CPU a ser apresentado para a tarefa. Ele pode ser expresso como um número inteiro usando unidades de CPU, por exemplo, `1024`, ou como uma string usando vCPUs, por exemplo, `1 vCPU` ou `1 vcpu`, em uma definição de tarefa. Quando a definição de tarefa for registrada, um valor de vCPU será convertido em um inteiro indicando as unidades de CPU.

Se estiver usando o tipo de inicialização Fargate, esse campo será obrigatório e você deverá usar um dos valores a seguir, que determina o intervalo de valores compatíveis para o parâmetro `memory`:

Valor de CPU	Valor de memória (MiB)
256 (0,25 vCPU)	512 (0,5 GB), 1024 (1 GB), 2048 (2 GB)
512 (0,5 vCPU)	1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)
1024 (1 vCPU)	2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)
2048 (2 vCPU)	Entre 4096 (4 GB) e 16384 (16 GB) em incrementos de 1024 (1 GB)

Valor de CPU	Valor de memória (MiB)
4096 (4 vCPU)	Entre 8192 (8 GB) e 30720 (30 GB) em incrementos de 1024 (1 GB)

`memory`

Tipo: string

Obrigatório: não

Note

Este parâmetro não é compatível com contêineres do Windows.

O limite rígido (em MiB) de memória a ser apresentado para a tarefa. Ela pode ser expressa como um inteiro usando MiB, por exemplo `1024`, ou como uma string usando GB, por exemplo `1GB` ou `1 GB`, em uma definição de tarefa. Quando a definição de tarefa for registrada, um valor em GB será convertido em um inteiro indicando o MiB.

Se estiver usando o tipo de execução Fargate, esse campo será obrigatório e você deverá usar um dos valores a seguir, que determina o intervalo de valores compatíveis para o parâmetro `cpu`:

Valor de memória (MiB)	Valor de CPU
512 (0,5 GB), 1024 (1 GB), 2048 (2 GB)	256 (0,25 vCPU)
1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)	512 (0,5 vCPU)
2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)	1024 (1 vCPU)
Entre 4096 (4 GB) e 16384 (16 GB) em incrementos de 1024 (1 GB)	2048 (2 vCPU)
Entre 8192 (8 GB) e 30720 (30 GB) em incrementos de 1024 (1 GB)	4096 (4 vCPU)

Configuração do proxy

`proxyConfiguration`

Tipo: objeto ProxyConfiguration

Obrigatório: não

Os detalhes de configuração do proxy do App Mesh.

Para tarefas que usam o tipo de inicialização Fargate, esse recurso exige que a tarefa ou o serviço use a plataforma versão 1.3.0 ou posterior.

Note

Este parâmetro não é compatível com contêineres do Windows.

```
"proxyConfiguration": {
    "type": "APPMESH",
```

```
    "containerName": "string",
    "properties": [
        {
           "name": "string",
           "value": "string"
        }
    ]
}
```

`type`

Tipo: string

Valor: `APPMESH`

Obrigatório: não

O tipo de proxy. O único valor suportado é `APPMESH`.

`containerName`

Tipo: string

Obrigatório: sim

O nome do contêiner que servirá como proxy do App Mesh.

`properties`

Tipo: matriz de objetos KeyValuePair

Obrigatório: não

O conjunto de parâmetros de configuração de rede para fornecer o plug-in Container Network Interface (CNI), especificado como pares de chave/valor.

- `IgnoredUID` – (Obrigatório) O ID de usuário (UID) do contêiner de proxy conforme definido pelo parâmetro `user` em uma definição de contêiner. Isso é usado para garantir que o proxy ignore seu próprio tráfego. Se `IgnoredGID` for especificado, esse campo poderá estar vazio.
- `IgnoredGID` – (Obrigatório) O ID de grupo (GID) do contêiner de proxy conforme definido pelo parâmetro `user` em uma definição de contêiner. Isso é usado para garantir que o proxy ignore seu próprio tráfego. Se `IgnoredUID` for especificado, esse campo poderá estar vazio.
- `AppPorts` – (Obrigatório) A lista de portas que o aplicativo usa. O tráfego de rede para essas portas é encaminhado para `ProxyIngressPort` e `ProxyEgressPort`.
- `ProxyIngressPort` – (Obrigatório) Especifica a porta para a qual o tráfego de entrada para `AppPorts` é direcionado.
- `ProxyEgressPort` – (Obrigatório) Especifica a porta para a qual o tráfego de saída de `AppPorts` é direcionado.
- `EgressIgnoredPorts` – (Obrigatório) O tráfego de saída para essas portas especificadas é ignorado e não é redirecionado para `ProxyEgressPort`. Ele pode ser uma lista vazia.
- `EgressIgnoredIPs` – (Obrigatório) O tráfego de saída para esses endereços IP especificados é ignorado e não é redirecionado para `ProxyEgressPort`. Ele pode ser uma lista vazia.

`name`

Tipo: string

Obrigatório: não

O nome do par de chave/valor.

`value`

Tipo: string

Obrigatório: não

O valor do par de chave/valor.

Outros parâmetros de definição de tarefa

Os parâmetros de definição de tarefa a seguir podem ser usados durante o registro das definições de tarefa no console do Amazon ECS usando a opção Configure via JSON (Configurar via JSON). Para obter mais informações, consulte Como criar uma definição de tarefa (p. 29).

Tópicos

- Modo IPC (p. 58)
- Modo PID (p. 58)

Modo IPC

`ipcMode`

Tipo: string

Obrigatório: não

O namespace de recurso IPC a ser usado para os contêineres na tarefa. Os valores válidos são `host`, `task` ou `none`. Se o `host` for especificado, todos os contêineres das tarefas que especificaram o modo IPC `host` na mesma instância de contêiner compartilham os mesmos recursos IPC com a instância host do Amazon EC2. Se a `task` for especificada, todos os contêineres da tarefa especificada compartilharão os mesmos recursos IPC. Se `none` for especificado, os recursos IPC nos contêineres de uma tarefa serão privados e não serão compartilhados com outros contêineres em uma tarefa ou na instância de contêiner. Se nenhum valor for especificado, o compartilhamento do namespace de recurso IPC dependerá da configuração do daemon do Docker na instância de contêiner. Para obter mais informações, consulte Configurações IPC na Referência de execução do Docker.

Se o modo IPC `host` for usado, esteja ciente de que há um risco elevado de exposição indesejada do namespace IPC. Para obter mais informações, consulte Segurança do Docker.

Se você estiver definindo parâmetros de kernel com namespace usando `systemControls` para os contêineres na tarefa, os itens a seguir serão aplicados ao seu namespace de recurso IPC. Para obter mais informações, consulte Controles do sistema (p. 52).

- Para tarefas que usam o modo IPC `host`, o namespace IPC `systemControls` relacionado não é compatível.
- Para tarefas que usam o modo IPC `task`, o namespace IPC `systemControls` relacionado será aplicado a todos os contêineres em uma tarefa.

Note

Não há suporte para esse parâmetro para os contêineres do Windows ou tarefas que usam o tipo de inicialização Fargate.

Modo PID

`pidMode`

Tipo: string

Obrigatório: não

O namespace do processo a ser usado para os contêineres na tarefa. Os valores válidos são `host` ou `task`. Se o `host` for especificado, todos os contêineres nas tarefas que especificaram o modo PID `host` na mesma instância de contêiner compartilharão o mesmo namespace do processo com a instância host do Amazon EC2. Se a `task` for especificada, todos os contêineres da tarefa especificada compartilharão o mesmo namespace de processo. Se nenhum valor for especificado, o padrão será um namespace privado. Para obter mais informações, consulte Configurações PID na Referência de execução do Docker.

Se o modo PID `host` for usado, esteja ciente de que há um risco elevado de exposição indesejada do namespace de processo. Para obter mais informações, consulte Segurança do Docker.

Note

Não há suporte para esse parâmetro para os contêineres do Windows ou tarefas que usam o tipo de inicialização Fargate.

Tipos de inicialização Amazon ECS

Um tipo de inicialização Amazon ECS determina o tipo de infraestrutura na qual suas tarefas e seus serviços serão hospedados.

Tipo de inicialização Fargate

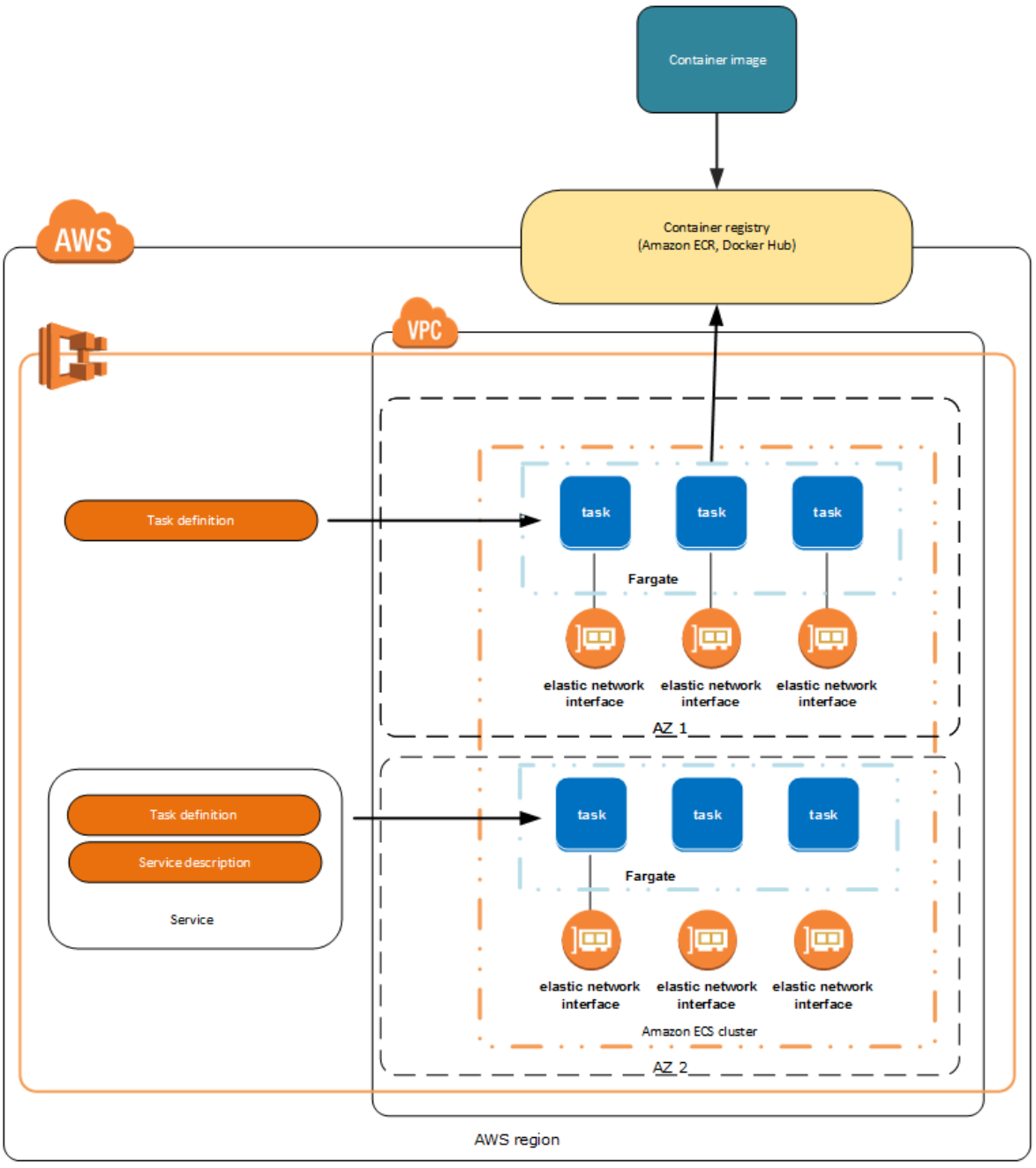
O tipo de inicialização Fargate permite que você execute seus aplicativos em contêineres sem a necessidade de provisionar e gerenciar a infraestrutura de back-end. Basta registrar sua definição de tarefa para o Fargate iniciar o contêiner para você.

No momento, o tipo de inicialização AWS Fargate está disponível nas seguintes regiões:

Nome da região	Região
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2
Oeste dos EUA (Norte da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
Ásia Pacífico (Mumbai)	ap-south-1
UE (Irlanda)	eu-west-1
UE (Londres)	eu-west-2
UE (Frankfurt)	eu-central-1
Ásia-Pacífico (Tóquio)	ap-northeast-1
Canadá (Central)	ca-central-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Cingapura)	ap-southeast-1

Nome da região	Região
Ásia-Pacífico (Sydney)	ap-southeast-2

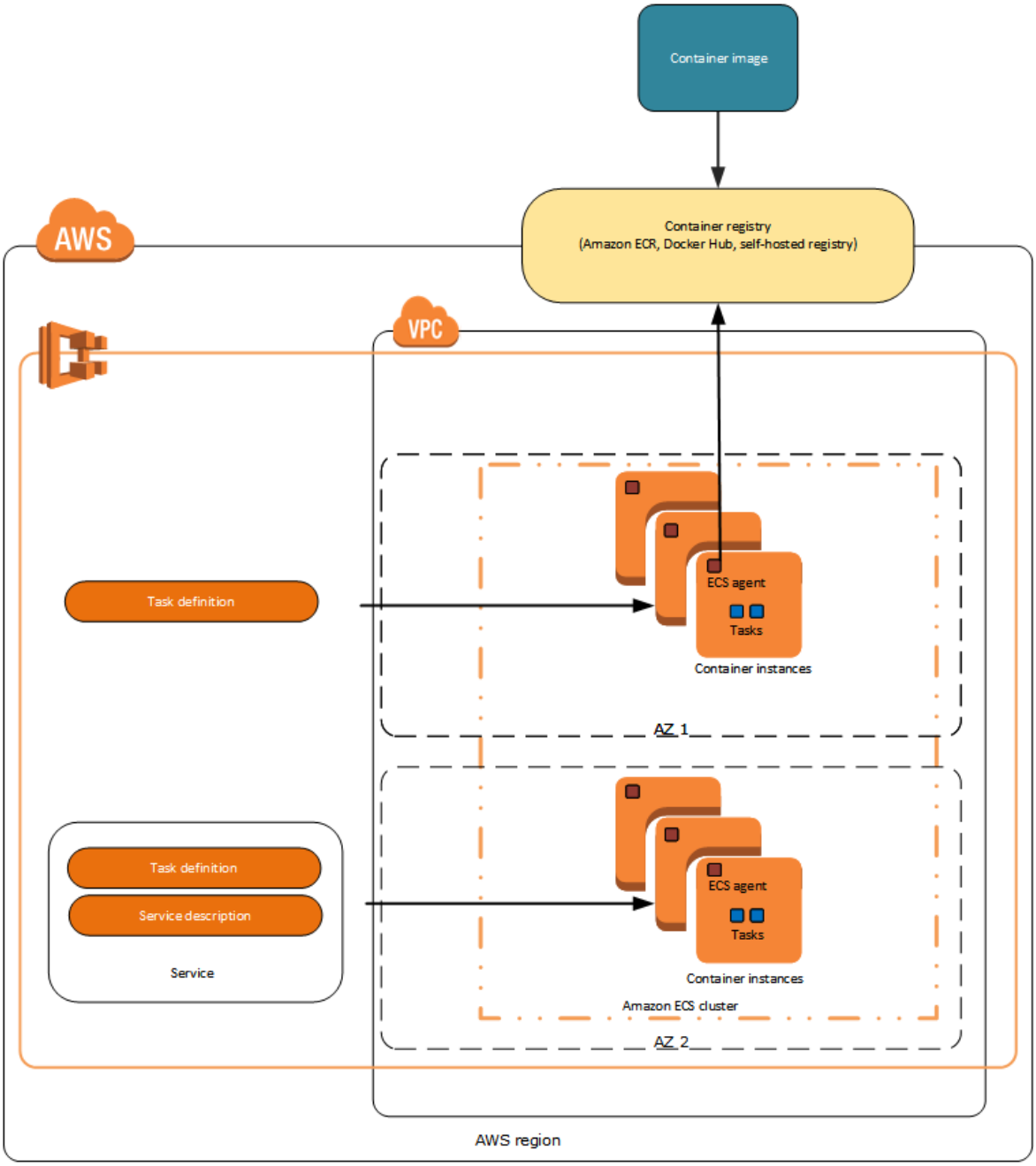
Este diagrama mostra a arquitetura geral:



Tipo de inicialização EC2

O tipo de inicialização EC2 permite que você execute seus aplicativos em contêineres em um cluster de instâncias do Amazon EC2 que você gerencia.

Este diagrama mostra a arquitetura geral:



Como usar volumes de dados em tarefas

- Para fornecer volumes de dados persistentes para uso com um contêiner
- Para definir um volume de dados não persistente e vazio, além de montá-lo em vários contêineres
- Para compartilhar volumes de dados definidos em locais diferentes em contêineres diferentes na mesma instância de contêiner
- Para fornecer um volume de dados para sua tarefa gerenciada por um driver de volume de terceiros

O ciclo de vida do volume pode ser vinculado a uma tarefa específica ou ao ciclo de vida de uma instância de contêiner específica.

Quando provisionada, cada tarefa do Fargate recebe o seguinte armazenamento. O armazenamento de tarefas é efêmero. Depois que uma tarefa do Fargate é interrompida, o armazenamento é excluído.

- 10 GB de armazenamento de camadas do Docker
- Mais 4 GB para montagens de volume. Eles podem ser montados e compartilhados entre os contêineres usando os parâmetros `volumes`, `mountPoints` e `volumesFrom` na definição da tarefa.

  Note

  Os parâmetros `host` e `sourcePath` não oferecem suporte a tarefas do Fargate.

Para obter mais informações sobre os limites de serviço padrão do Amazon ECS, consulte Limites de serviço do Amazon ECS (p. 277).

Para fornecer armazenamento vazio não persistente para contêineres em tarefas do Fargate

Neste exemplo, você pode ter dois contêineres de banco de dados que precisam acessar o mesmo local de armazenamento de arquivos temporários durante uma tarefa.

1. Na seção `volumes` da definição da tarefa, defina um volume com o nome `database_scratch`.

   ```
   "volumes": [ { "name": "database_scratch", "host": {} } ]
   ```

2. Na seção `containerDefinitions`, crie as definições de contêiner de banco de dados para que elas montem o armazenamento não persistente.

   ```
   "containerDefinitions": [ { "name": "database1", "image": "my-repo/database",
   "cpu": 100, "memory": 100, "essential": true, "mountPoints": [ { "sourceVolume":
   "database_scratch", "containerPath": "/var/scratch" } ] }, { "name": "database2",
   "image": "my-repo/database", "cpu": 100, "memory": 100, "essential": true,
   "mountPoints": [ { "sourceVolume": "database_scratch", "containerPath": "/var/
   scratch" } ] } ]
   ```

Redes de tarefas com o modo de rede do `awsvpc`

Os recursos de redes de tarefas fornecidos pelo modo de rede do `awsvpc` conferem às tarefas do Amazon ECS as mesmas propriedades de redes que as instâncias do Amazon EC2. Quando você usa o modo de rede `awsvpc` em suas definições de tarefa, todas as tarefas executadas a partir dessa definição obterão sua própria interface de rede elástica (ENI), um endereço IP privado principal e um nome de host DNS interno. O recurso de rede de tarefas simplifica a rede de contêineres e oferece a você mais controle sobre como aplicativos em contêineres se comunicam entre si e com outros serviços dentro das suas VPCs.

Note

Para obter informações sobre outros modos de rede disponíveis para tarefas, consulte Modo de rede (p. 35).

A rede de tarefas também fornece maior segurança para seus contêineres, permitindo que você use grupos de segurança e ferramentas de monitoramento de rede em um nível mais granular dentro das tarefas. Como cada tarefa recebe sua própria ENI, você também pode aproveitar outros recursos de rede do Amazon EC2, como VPC Flow Logs, para que possa monitorar o tráfego que entra e sai das suas tarefas. Além disso, os contêineres que pertencem à mesma tarefa podem se comunicar por meio da interface `localhost`. Uma tarefa só pode ter uma ENI associada a ela em determinado momento.

A ENI de tarefa criada é totalmente gerenciada pelo Amazon ECS. O A tarefa envia e recebe tráfego de rede na ENI da mesma maneira que as instâncias do Amazon EC2 fazem com suas interfaces de rede principais. Essas ENIs estão visíveis no console do Amazon EC2 para a sua conta, mas não podem ser separadas manualmente nem modificadas por sua conta. Isso visa a evitar a exclusão acidental de uma ENI associada a uma tarefa em execução. Você pode visualizar as informações do anexo da ENI para tarefas no console do Amazon ECS ou com a operação da API DescribeTasks. Quando a tarefa for interrompida ou se o serviço for escalado para baixo, a ENI de tarefa será desassociada e excluída.

Considerações sobre redes de tarefas

Há vários aspectos a considerar ao usar a tarefa de rede.

- As tarefas e os serviços que usam o modo de rede `awsvpc` exigem a função vinculada ao serviço do Amazon ECS para fornecer ao Amazon ECS as permissões para fazer chamadas a outros serviços da AWS em seu nome. Essa função será automaticamente criada quando você criar um cluster ou se criar ou atualizar um serviço no Console de gerenciamento da AWS. Para obter mais informações, consulte Usar funções vinculadas ao serviço do Amazon ECS (p. 216). Você também pode criar a função vinculada ao serviço com o seguinte comando da AWS CLI:

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- Atualmente, apenas as variantes do Linux da Amazon ECS-optimized AMI ou outras variantes do Amazon Linux com o pacote `ecs-init` são compatíveis com as redes de tarefas.
- As tarefas Fargate podem ser configuradas para receber endereços IP públicos.
- Há um limite de 16 sub-redes e 5 grupos de segurança que podem ser especificados na `awsvpcConfiguration` ao executar uma tarefa ou ao criar um serviço que usa o modo de rede `awsvpc`. Para obter mais informações, consulte AwsVpcConfiguration no Amazon Elastic Container Service API Reference.
- As ENIs criadas e associadas pelo Amazon ECS não podem ser separadas manualmente nem modificadas pela sua conta. Isso visa a evitar a exclusão acidental de uma ENI associada a uma tarefa em execução. Para liberar as ENIs para uma tarefa, interrompa-a.
- Quando uma tarefa é iniciada com o modo de rede `awsvpc`, o agente de contêiner do Amazon ECS cria um contêiner `pause` adicional para cada tarefa antes de iniciar os contêineres na definição de tarefa. Em seguida, ele configura o namespace de rede do contêiner `pause` executando os plug-ins amazon-ecs-cni-plugins do CNI. O agente inicia, então, o resto dos contêineres na tarefa para que eles compartilhem a pilha de rede do contêiner `pause`. Isso significa que todos os contêineres em uma tarefa são endereçáveis por endereços IP da ENI, e que eles podem se comunicar entre eles por meio da interface `localhost`.
- Serviços com tarefas que usam o modo de rede `awsvpc`, por exemplo, aqueles com o tipo de execução Fargate, oferecem suporte somente a Balanceador de carga de aplicaçõess e Load balancer de redes. Não há suporte para Classic Load Balancers. Além disso, ao criar grupos de destino para esses serviços, você precisa escolher `ip` como o tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` são associadas a uma ENI, não a uma instância do Amazon EC2. Para obter mais informações, consulte Balanceamento de carga do serviço (p. 115).
- Se uma VPC for atualizada, por exemplo, para alterar o conjunto de opções DHCP que ela usa, e você quiser que as tarefas usando a VPC peguem as alterações, essas tarefas deverão ser interrompidas e novas tarefas iniciadas.

Habilitar a rede de tarefas

As tarefas Fargate exigem o uso do modo de rede `awsvpc` para que as redes de tarefas sejam ativadas por padrão. Sua definição de tarefa deve especificar o modo de rede `awsvpc`. Para obter mais informações, consulte Modo de rede (p. 35). Depois, quando você executar uma tarefa ou criar um serviço, especifique uma configuração de rede que inclua uma ou mais sub-redes nas quais colocar suas

tarefas e os grupos de segurança para associar à ENI associada. As tarefas Fargate são executadas nessas sub-redes e os grupos de segurança especificados são associados à ENI provisionada para a tarefa.

Como usar o driver de log awslogs

Você pode configurar os contêineres nas tarefas para enviar informações de log ao CloudWatch Logs. Isso permite que você visualize os logs dos contêineres nas tarefas Fargate.Este tópico ajuda nos conceitos básicos de como usar o driver de log `awslogs` nas definições de tarefa.

Note

O tipo de informações registradas em log pelos contêineres em sua tarefa depende principalmente do comando `ENTRYPOINT`. Por padrão, os logs capturados mostram a saída do comando que você normalmente veria em um terminal interativo, se executasse o contêiner localmente, que são os fluxos de E/S `STDOUT` e `STDERR`. O driver de log `awslogs` simplesmente passa esses logs do Docker para o CloudWatch Logs. Para obter mais informações sobre como os logs do Docker são processados, incluindo maneiras alternativas de capturar fluxos ou dados de arquivos diferentes, consulte Visualizar logs de um contêiner ou serviço na documentação do Docker.

Ativação do driver de logs awslogs para seus contêineres

Se estiver usando o tipo de inicialização Fargate para suas tarefas, para habilitar o driver de logs `awslogs` você só precisa adicionar à definição de tarefa os parâmetros `logConfiguration` requeridos. Para obter mais informações, consulte Como especificar uma configuração de log na definição da tarefa (p. 66).

Criar um grupo de logs

O driver de log `awslogs` pode enviar fluxos de log para um grupo de logs existente no CloudWatch Logs ou pode criar um novo grupo de logs em seu nome. O Console de gerenciamento da AWS fornece uma opção de configuração automática que cria um grupo de logs em seu nome usando o nome da família de definição da tarefa com `ecs` como prefixo. Como alternativa, é possível especificar manualmente as opções de configuração de log e especificar a opção `awslogs-create-group` com um valor de `true` que criará os grupos de logs em seu nome.

Note

Para usar a opção `awslogs-create-group` para criar seu grupo de logs, sua política do IAM deve incluir a permissão `logs:CreateLogGroup`.

Uso do recurso de configuração automática para criar um grupo de logs

Ao registrar uma definição de tarefa no console do Amazon ECS, você terá a opção de permitir que o Amazon ECS configure automaticamente os logs do CloudWatch. Essa opção cria um grupo de logs em seu nome usando o nome da família de definição da tarefa com `ecs` como prefixo.

Como usar a opção de configuração automática do grupo de logs no console do Amazon ECS

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.

2. No painel de navegação à esquerda, escolha Task Definitions, Create new Task Definition.
3. Escolha suas opções de compatibilidade e selecione Next Step (Próxima etapa).
4. Escolha Add container.
5. Na seção Storage and Logging (Armazenamento e registro em log), em Log configuration (Configuração de log), selecione Auto-configure CloudWatch Logs (Configurar o CloudWatch Logs automaticamente).
6. Insira as opções do driver de logs awslogs. Para obter mais informações, consulte Como especificar uma configuração de log na definição da tarefa (p. 66).
7. Preencha o restante do assistente de definição de tarefas.

Opções do driver de log awslogs disponíveis

O driver de log `awslogs` dá suporte às seguintes opções em definições de tarefa do Amazon ECS. Para obter mais informações, consulte Driver de registro em log do CloudWatch Logs.

`awslogs-create-group`

Obrigatório: não

Especifique se você deseja que o grupo de logs seja criado automaticamente. Se esta opção não for especificada, o padrão será `false`.

Note

Sua política do IAM deve incluir a permissão `logs:CreateLogGroup` antes de tentar usar `awslogs-create-group`.

`awslogs-region`

Obrigatório: sim

Especifique a região para a qual o driver de log `awslogs` deve enviar os logs do Docker. Você pode optar por enviar todos os logs de clusters em regiões diferentes para uma única região no CloudWatch Logs, de maneira que eles permaneçam todos visíveis em um local, ou é possível separá-los por região para mais granularidade. Certifique-se de que o grupo de logs especificado exista na região especificada com essa opção.

`awslogs-group`

Obrigatório: sim

Você deve especificar um grupo de logs para o qual o driver de log `awslogs` enviará os fluxos de log. Para obter mais informações, consulte Criar um grupo de logs (p. 64).

`awslogs-stream-prefix`

Obrigatório: sim, ao usar o tipo de inicialização Fargate.

A opção `awslogs-stream-prefix` permite associar um fluxo de log ao prefixo especificado, ao nome do contêiner e à ID da tarefa do Amazon ECS a que o contêiner pertence. Caso você especifique um prefixo com essa opção, o fluxo de log utiliza o seguinte formato:

```
prefix-name/container-name/ecs-task-id
```

Para serviços do Amazon ECS, você pode usar o nome do serviço como o prefixo, o que permitiria que você rastreasse fluxos de log até o serviço a que o contêiner pertence, o nome do contêiner que os enviou e o ID da tarefa a que o contêiner pertence.

`awslogs-datetime-format`

Obrigatório: Não

Essa opção define um padrão de início de várias linhas no formato `strftime` em Python. Uma mensagem de log é formada por uma linha em conformidade com o padrão e as linhas seguintes que não correspondem ao padrão. Assim, a linha em conformidade é o delimitador entre as mensagens de log.

Um exemplo de um caso de uso para esse formato é a análise da saída, como um despejo de pilha, que poderia ser registrado em várias entradas. O padrão correto permite que ele seja capturado em uma única entrada.

Para obter mais informações, consulte awslogs-datetime-format.

Essa opção sempre terá precedência se os `awslogs-datetime-format` e `awslogs-multiline-pattern` estiverem configurados.

> **Note**
>
> O registro em várias linhas executa a análise da expressão regular e a correspondência de todas as mensagens de log, o que pode ter um impacto negativo no desempenho do registro em log.

`awslogs-multiline-pattern`

Obrigatório: não

Essa opção define um padrão inicial de várias linhas usando uma expressão regular. Uma mensagem de log é formada por uma linha em conformidade com o padrão e as linhas seguintes que não correspondem ao padrão. Assim, a linha em conformidade é o delimitador entre as mensagens de log.

Para obter mais informações, consulte awslogs-multiline-pattern.

Essa opção será ignorada se `awslogs-datetime-format` também estiver configurado.

> **Note**
>
> O registro em várias linhas executa a análise da expressão regular e a correspondência de todas as mensagens de log. Isso pode ter um impacto negativo no desempenho do registro em log.

Como especificar uma configuração de log na definição da tarefa

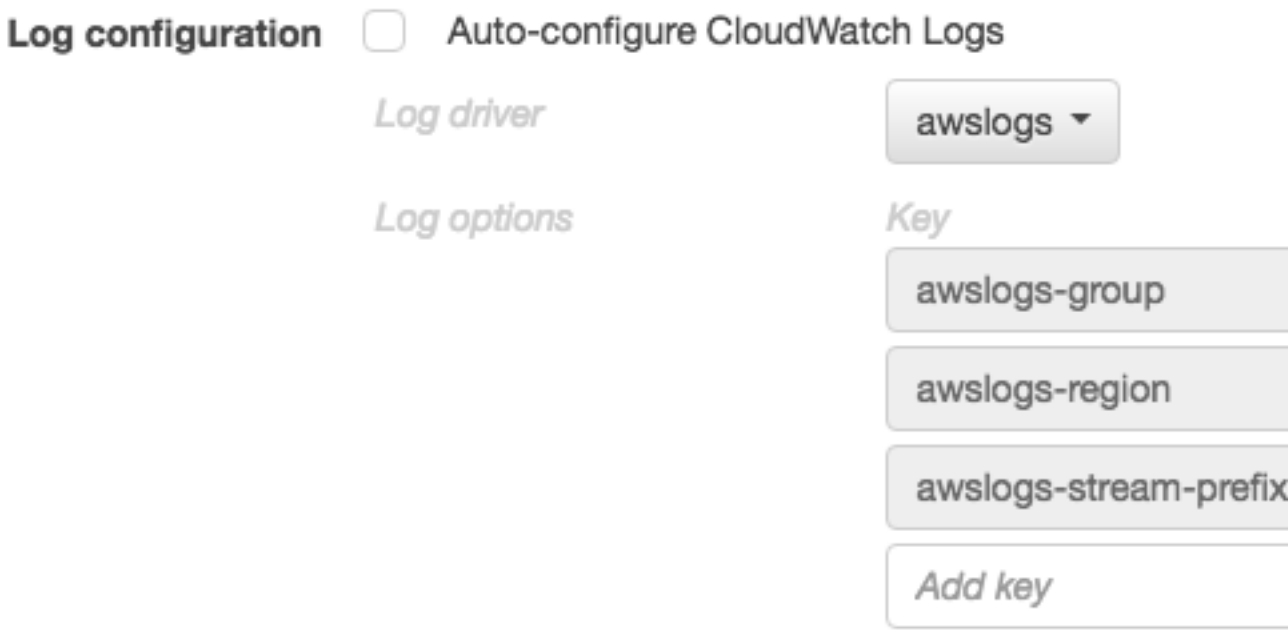
Para os contêineres enviarem logs ao CloudWatch, você deve especificar o driver de log `awslogs` para contêineres na definição da tarefa. Esta seção descreve a configuração de log para um contêiner usar o driver de log `awslogs`. Para obter mais informações, consulte Como criar uma definição de tarefa (p. 29).

O JSON de definição da tarefa mostrado abaixo tem um objeto `logConfiguration` especificado para cada contêiner; um para o contêiner do Word Press que envia logs para um grupo de logs chamado `awslogs-wordpress` e um para um contêiner do MySQL que envia logs para um grupo de logs chamado `awslogs-mysql`. Ambos os contêineres usam o prefixo de fluxo de log `awslogs-example`.

```
{
```

```
    "containerDefinitions": [
        {
            "name": "wordpress",
            "links": [
                "mysql"
            ],
            "image": "wordpress",
            "essential": true,
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80
                }
            ],
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "awslogs-wordpress",
                    "awslogs-region": "us-west-2",
                    "awslogs-stream-prefix": "awslogs-example"
                }
            },
            "memory": 500,
            "cpu": 10
        },
        {
            "environment": [
                {
                    "name": "MYSQL_ROOT_PASSWORD",
                    "value": "password"
                }
            ],
            "name": "mysql",
            "image": "mysql",
            "cpu": 10,
            "memory": 500,
            "essential": true,
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "awslogs-mysql",
                    "awslogs-region": "us-west-2",
                    "awslogs-stream-prefix": "awslogs-example"
                }
            }
        }
    ],
    "family": "awslogs-example"
}
```

No console do Amazon ECS, a configuração de log do contêiner `wordpress` é especificada conforme mostrado na imagem abaixo.

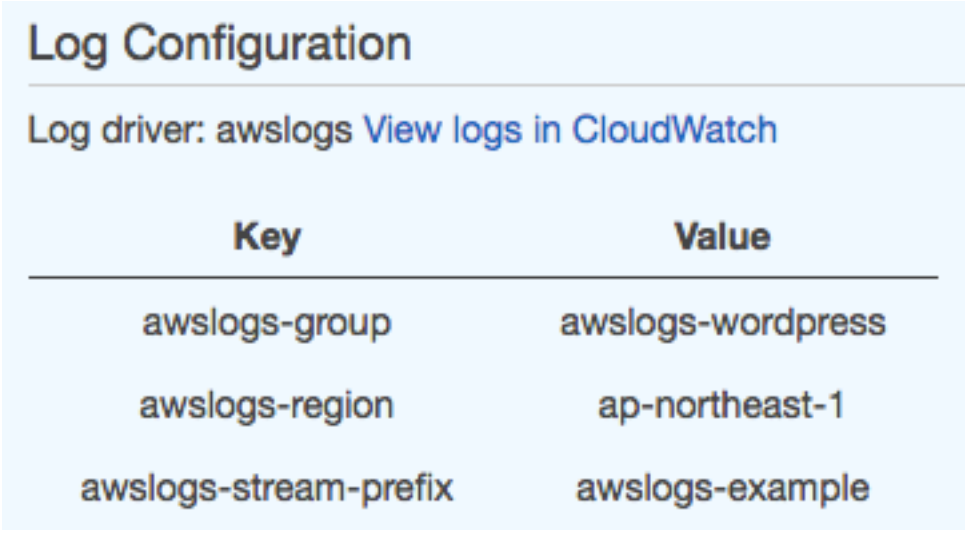


Como visualizar logs de contêiner awslogs no CloudWatch Logs

Depois que suas tarefas Fargate que usam o driver de log `awslogs` forem executadas, seus contêineres configurados devem enviar seus dados de registro para o CloudWatch Logs. Você pode visualizar e pesquisar esses logs no console.

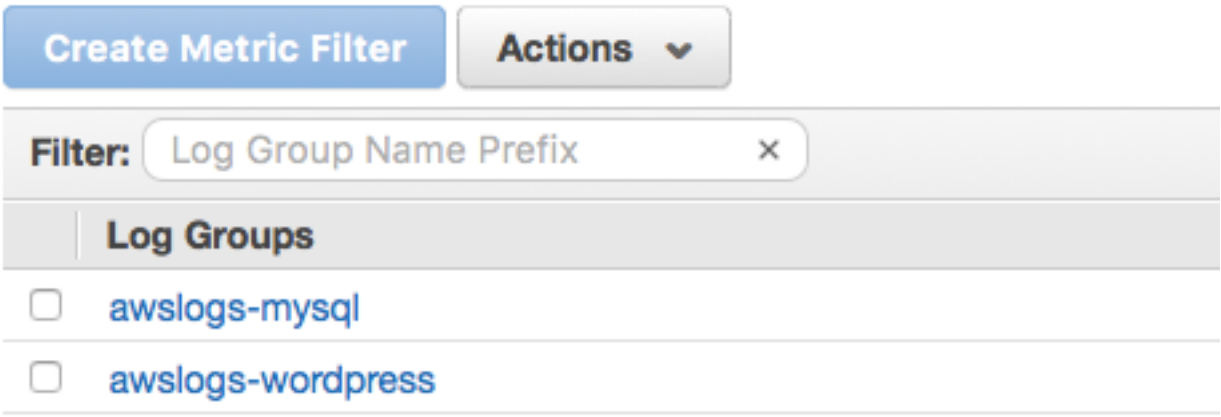
Para visualizar os seus dados do CloudWatch Logs para um contêiner do console do Amazon ECS

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na página Clusters, selecione o cluster que contém a tarefa a ser exibida.
3. Na página Cluster: ***`cluster_name,`***, escolha Tasks e selecione a tarefa a ser exibida.
4. Na página Task: ***`task_id`***, expanda a visualização de contêiner, escolhendo a seta para a esquerda do nome do contêiner.
5. Na seção Log Configuration (Configuração de log), escolha View logs in CloudWatch (Exibir logs no CW), que abre o streaming de logs associado no console do CloudWatch.

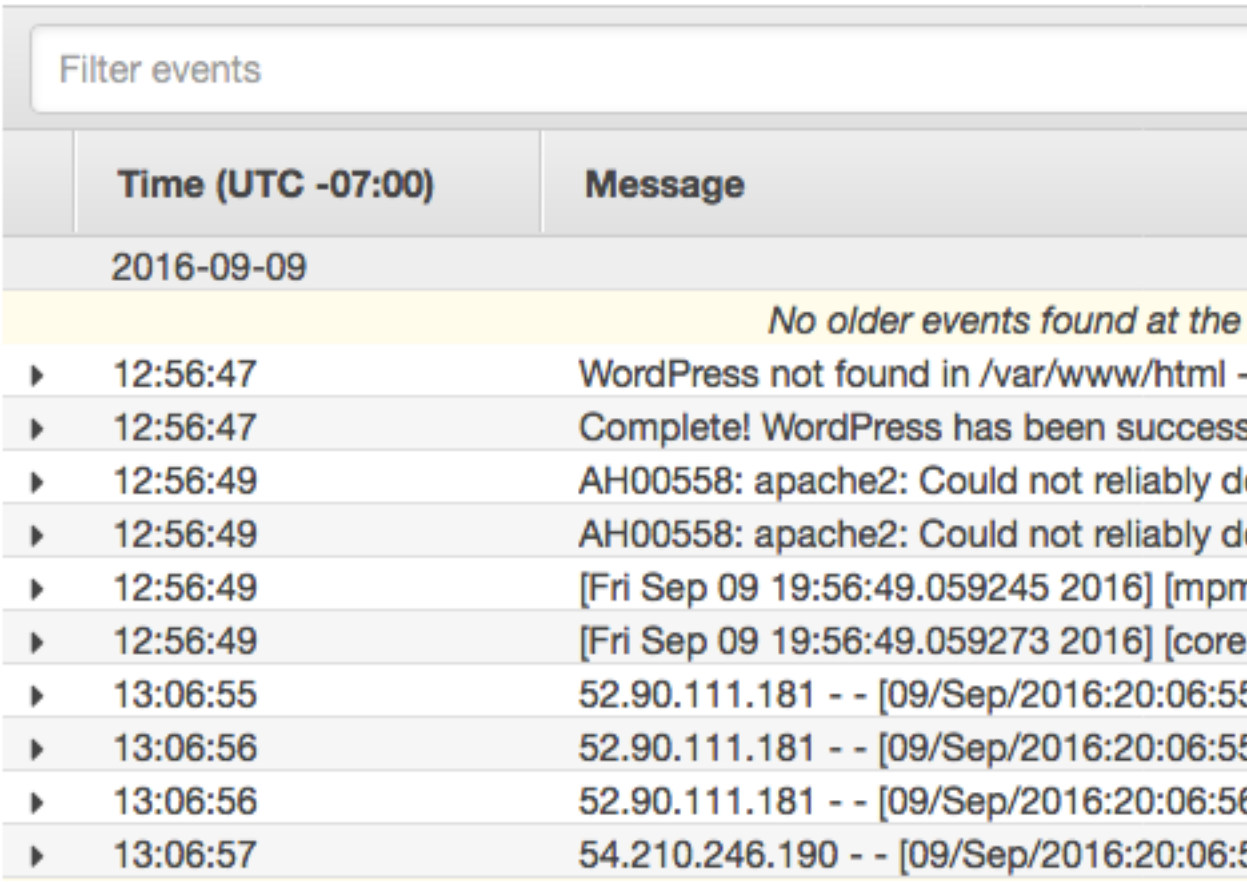


Para visualizar seus dados do CloudWatch Logs no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação esquerdo, selecione Logs.
3. Selecione um grupo de logs para exibir. Você deve ver os grupos de logs criados em Criar um grupo de logs (p. 64).



4. Escolha um stream de logs para visualizar.



Rotear logs personalizados

FireLens for Amazon ECS is in open preview. The preview is open to all AWS accounts and you do not need to request access. Features may be added or changed before announcing General Availability. Don't hesitate to contact us with any feedback by commenting on https://github.com/aws/containers-roadmap/issues/10

O FireLens para Amazon ECS permite usar parâmetros de definição de tarefa a fim de rotear logs para um serviço da AWS ou destino de um parceiro para armazenamento e análise de logs. O FireLens funciona com Fluentd e Fluent Bit. Fornecemos o plug-in AWS for Fluent Bit, ou é possível trazer seu próprio plug-in de saída Fluentd ou Fluent Bit para usar.

Durante a visualização pública, estamos fornecendo ao FireLens um conjunto básico de funcionalidades para permitir que você o teste e nos dê feedback. Assim que anunciarmos a disponibilidade geral do FireLens, ele estará pronto para cargas de trabalho de produção e permitirá outros casos de uso.

A criação de definições de tarefa do Amazon ECS com uma configuração do FireLens tem suporte por meio da AWS CLI, do Console de gerenciamento da AWS e dos SDKs da AWS. Ao usar o Console de gerenciamento da AWS para registrar uma nova definição de tarefa, é necessário usar a opção Configure via JSON (Configurar via JSON) .

> **Note**
>
> O FireLens está disponível para tarefas do Amazon ECS que usam os tipos de inicialização EC2 e Fargate.

Para obter mais informações, consulte https://github.com/aws/containers-roadmap/tree/master/preview-programs/firelens.

Autenticação de registro privado para tarefas

A autenticação de registro privado para tarefas usando o AWS Secrets Manager permite armazenar suas credenciais de forma segura e, então, referenciá-las em sua definição de contêiner. Isso permite que suas tarefas usem imagens de repositórios privados. Esse recurso é compatível com tarefas que usam o tipo de inicialização Fargate ou EC2.

> **Important**
>
> Se a definição da tarefa faz referência a uma imagem armazenada no Amazon ECR, este tópico não se aplica. Para obter mais informações, consulte Uso de imagens do Amazon ECR com o Amazon ECS no Guia do usuário do Amazon Elastic Container Registry.

Para tarefas que usam o tipo de execução Fargate, esse recurso exige a plataforma versão 1.2.0 ou posterior. Para obter informações, consulte Versões de plataforma do AWS Fargate (p. 20).

Em sua definição de contêiner, especifique `repositoryCredentials` com o ARN completo do segredo que você criou. O segredo que você referencia não precisa ser da mesma região da tarefa que o está usando, mas deve estar na mesma conta.

> **Note**
>
> Ao usar a API do Amazon ECS, a AWS CLI ou o SDK da AWS, se o segredo existir na mesma região da tarefa que estiver ativando, você poderá usar o ARN completo ou o nome do segredo. Ao usar o Console de gerenciamento da AWS, o ARN completo do segredo deve ser especificado.

Veja a seguir um trecho de uma definição de tarefa que mostra os parâmetros necessários:

```
"containerDefinitions": [
    {
        "image": "private-repo/private-image",
        "repositoryCredentials": {
            "credentialsParameter":
 "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"
        }
    }
]
```

Permissões do IAM necessárias para a autenticação de registro privado

A função de execução da tarefa do Amazon ECS é necessária para usar esse recurso. Isso permite que o agente de contêiner obtenha a imagem do contêiner. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Para dar acesso aos segredos criados por você, adicione manualmente as permissões a seguir como uma política em linha à função de execução da tarefa. Para obter mais informações, consulte Adicionar e remover políticas do IAM.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`—Só será exigido se a chave usar uma chave do KMS personalizada e não a chave padrão. O ARN da chave personalizada deve ser adicionado como um recurso.

Veja abaixo um exemplo de política em linha adicionando as permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
        "arn:aws:kms:<region>:<aws_account_id>:key:key_id"
      ]
    }
  ]
}
```

Habilitar a autenticação de registro privado

Para criar um segredo básico

Use o AWS Secrets Manager para criar um segredo para suas credenciais de registro privado.

1. Abra o console do AWS Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
2. Selecione Store a new secret (Armazenar um novo segredo).
3. Em Select secret type (Selecionar tipo de segredo), selecione Other type of secrets (Outro tipo de segredos).
4. Selecione Plaintext (Texto simples) e insira suas credenciais de registro privado usando o seguinte formato:

   ```
   {
     "username" : "privateRegistryUsername",
     "password" : "privateRegistryPassword"
   }
   ```
5. Escolha Next.
6. Em Secret name (Nome de segredo), digite um nome e caminho opcionais, como **`production/MyAwesomeAppSecret`** ou **`development/TestSecret`** e escolha Next (Avançar). Se preferir, adicione uma descrição para ajudá-lo a lembrar a finalidade desse segredo mais tarde.

   O nome do segredo deve ter somente letras ASCII, números ou qualquer um dos seguintes caracteres: /_+=.@-
7. (Opcional) Nesse momento, você poderá configurar a rotação para o segredo. Para esse procedimento, deixe Disable automatic rotation (Desabilitar rotação automática) e escolha Next (Avançar).

   Para obter informações sobre como configurar a rotação em segredos novos ou existentes, consulte Alternar os segredos do AWS Secrets Manager.
8. Reveja suas configurações e, em seguida, selecione Store secret (Armazenar segredo) para salvar tudo o que inseriu como um novo segredo no Secrets Manager.

Para criar uma definição de tarefa que usa a autenticação de registro privado

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, selecione Task Definitions (Definições de tarefas).
3. Na página Task Definitions (Definições de tarefas), escolha Create new Task Definition (Criar nova definição de tarefa).
4. Na página Select launch type compatibility (Selecionar compatibilidade do tipo de inicialização), escolha o tipo de inicialização das tarefas e Next step (Próxima tarefa).
5. Em Task Definition Name (Nome da definição da tarefa), digite um nome para a sua definição de tarefa. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.
6. Em Task execution role (Função de execução da tarefa), selecione a função de execução da tarefa existente ou selecione Create new role (Criar outra função) para que ela seja criada para você. Essa função autoriza o Amazon ECS a extrair imagens privadas da sua tarefa. Para obter mais informações, consulte Permissões do IAM necessárias para a autenticação de registro privado (p. 71).

   Important

   Se o campo Task execution role (Função de execução de tarefas) não aparecer, escolha Configure via JSON (Configurar via JSON) e adicione manualmente o campo `executionRoleArn` para especificar a função de execução da tarefa. Veja a seguir a sintaxe:

   ```
   "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole"
   ```
7. Para cada contêiner a ser criado em sua definição de tarefa, conclua as seguintes etapas:
   a. Na seção Container Definitions (Definições de contêiner), selecione Add container (Adicionar contêiner).
   b. Em Container name (Nome do contêiner), digite um nome para o contêiner. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.
   c. Em Image (Imagem), digite o nome da imagem ou o caminho para sua imagem privada. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.
   d. Selecione a opção Private repository authentication (Autenticação do repositório privado).
   e. Em Secrets manager ARN (ARN do Secrets Manager), insira o nome de recurso da Amazon (ARN) do segredo que você criou anteriormente. O valor de deve ter entre 20 e 2048 caracteres.
   f. Preencha os demais campos obrigatórios e qualquer campo opcional a ser usado em suas definições de contêiner. Mais parâmetros de definição de contêiner estão disponíveis no menu Advanced container configuration (Configuração de contêiner avançada). Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34).
   g. Escolha Adicionar.
8. Quando seus contêineres forem adicionados, selecione Create (Criar).

Especificação de dados confidenciais

O Amazon ECS permite injetar dados confidenciais nos contêineres, armazenando seus dados confidenciais nos segredos do AWS Secrets Manager ou nos parâmetros do Repositório de parâmetros do AWS Systems Manager e, então, fazendo referência a eles na definição de contêiner. Esse recurso é compatível com tarefas que usam os tipos de inicialização do EC2 e do Fargate.

Os segredos podem ser expostos a um contêiner das seguintes formas:

- Para injetar dados confidenciais em seus contêineres como variáveis de ambiente, use o parâmetro de definição de contêiner `secrets`.

- Para fazer referência a informações confidenciais na configuração de log de um contêiner, use o parâmetro de definição de contêiner `secretOptions`.

Considerações para especificar dados confidenciais

O seguinte deve ser considerado ao especificar dados confidenciais para contêineres:

- Para as tarefas que usam o tipo de inicialização do Fargate, esse recurso exige que sua tarefa use a versão 1.3.0 da plataforma ou posterior. Para obter informações, consulte Versões de plataforma do AWS Fargate (p. 20).
-
- Os dados confidenciais são injetados no contêiner inicialmente quando o contêiner é iniciado. Se o segredo ou o parâmetro Parameter Store for posteriormente atualizado ou modificado, o contêiner não receberá o valor atualizado automaticamente. Você deve executar uma nova tarefa ou se a tarefa for parte de um serviço, você poderá atualizar o serviço e usar a opção Force new deployment (Forçar nova implantação) para forçar o serviço a iniciar uma nova tarefa.
- Esse recurso não está disponível na região GovCloud (Leste dos EUA).

Injetar dados confidenciais como uma variável de ambiente

Em sua definição de contêiner, especifique `secrets` com o nome da variável de ambiente a ser definida no contêiner e o ARN completo do segredo do Secrets Manager ou do parâmetro do Parameter Store do Systems Manager que contém os dados confidenciais a serem apresentados ao contêiner. O parâmetro ao qual você faz referência deve estar na mesma conta, mas deve ser de uma região diferente da do contêiner que está usando o parâmetro.

> **Important**
>
> Se o parâmetro do Parameter Store do Systems Manager existir na mesma região da tarefa que está sendo iniciada, você poderá usar o ARN completo ou o nome do parâmetro. Se o parâmetro existir em uma região diferente, o ARN completo deverá ser especificado.

Para um tutorial completo sobre como criar um segredo do Secrets Manager e injetá-lo em um contêiner como uma variável de ambiente, consulte Tutorial: Especificação de dados confidenciais usando segredos do Secrets Manager (p. 291).

Veja a seguir um trecho de uma definição de tarefa mostrando o formato ao fazer referência a um segredo do Secrets Manager.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
    }]
  }]
}
```

Veja a seguir um trecho de uma definição de tarefa mostrando o formato ao fazer referência a um parâmetro do Parameter Store do Systems Manager.

```
{
  "containerDefinitions": [{
```

```
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Injetar dados confidenciais em uma configuração de log

Em sua definição de contêiner, ao especificar `logConfiguration`, você poderá especificar `secretOptions` com o nome da opção de driver de log a ser definida no contêiner e o ARN completo do segredo do Secrets Manager ou do parâmetro do Parameter Store do Systems Manager que contém os dados confidenciais a serem apresentados ao contêiner. O parâmetro ao qual você faz referência deve estar na mesma conta, mas deve ser de uma região diferente da do contêiner que está usando o parâmetro.

> **Important**
>
> Se o parâmetro do Parameter Store do Systems Manager existir na mesma região da tarefa que está sendo iniciada, você poderá usar o ARN completo ou o nome do parâmetro. Se o parâmetro existir em uma região diferente, o ARN completo deverá ser especificado.

Veja a seguir um trecho de uma definição de tarefa mostrando o formato ao fazer referência a um segredo do Secrets Manager.

```
{
  "containerDefinitions": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}
```

Veja a seguir um trecho de uma definição de tarefa mostrando o formato ao fazer referência a um parâmetro do Parameter Store do Systems Manager.

```
{
  "containerDefinitions": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter:parameter_name"
      }]
    }]
  }]
}
```

Permissões necessárias do IAM para segredos do Amazon ECS

Para usar esse recurso, você deve ter a função de execução de tarefas do Amazon ECS e fazer referência a ela na definição de tarefa. Isso permite que o agente de contêiner obtenha os recursos necessários do AWS Systems Manager ou do Secrets Manager. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Para dar acesso aos parâmetros do Parameter Store do AWS Systems Manager criados por você, adicione manualmente as permissões a seguir como uma política em linha à função de execução da tarefa. Para obter mais informações, consulte Adicionar e remover políticas do IAM.

- `ssm:GetParameters` — Exigido se você estiver fazendo referência a um parâmetro do Repositório de parâmetros do Systems Manager em uma definição de tarefa.
- `secretsmanager:GetSecretValue` — Exigido se você estiver fazendo referência a um segredo do Secrets Manager diretamente ou se o parâmetro do Repositório de parâmetros do Systems Manager estiver fazendo referência a um segredo do Secrets Manager em uma definição de tarefa.
- `kms:Decrypt` — Exigido somente se o segredo usar uma chave personalizada do KMS e não a chave padrão. O ARN da chave personalizada deve ser adicionado como um recurso.

O exemplo de política em linha a seguir adiciona as permissões necessárias:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/parameter_name",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
        "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
      ]
    }
  ]
}
```

Criação de um parâmetro do Parameter Store do AWS Systems Manager

Você pode usar o console do AWS Systems Manager para criar um parâmetro do Parameter Store do Systems Manager para seus dados confidenciais. Para obter mais informações, consulte Demonstração: criar e usar um parâmetro em um comando (console) no Guia do usuário do AWS Systems Manager.

Para criar um parâmetro do Parameter Store

1. Abra o console do AWS Systems Manager em https://console.aws.amazon.com/systems-manager/.
2. No painel de navegação, escolha Parameter Store, Create parameter (Criar parâmetro).
3. Em Name (Nome), digite uma hierarquia e um nome de parâmetro. Por exemplo, digite `test/database_password`.
4. Em Description (Descrição), digite uma descrição opcional.

5. Para Type, escolha String, StringList ou SecureString.

   Note

   - Se escolher SecureString,, o campo KMS Key ID (ID da chave do KMS) será exibido. Se não fornecer o ID da CMK do KMS, o ARN da CMK do KMS, um nome de alias ou um ARN de alias, o sistema usará `alias/aws/ssm`, que é a CMK padrão do KMS para o Systems Manager. Para evitar o uso dessa chave, escolha uma chave personalizada. Para obter mais informações, consulte Usar parâmetro de string segura no Guia do usuário do AWS Systems Manager.
   - Ao criar um parâmetro de string segura no console usando o parâmetro `key-id` com um nome de alias personalizado da CMK do KMS ou um ARN de alias, você deve especificar o prefixo `alias/` antes do alias. Veja um exemplo de ARN a seguir:

     ```
     arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
     ```

     Veja a seguir um exemplo de nome de alias:

     ```
     alias/MyAliasName
     ```

6. Em Value (Valor), digite um valor. Por exemplo, `MyFirstParameter`. Se você escolheu SecureString, o valor será mascarado conforme você digitar.
7. Escolha Create parameter.

Criar uma definição de tarefa que faz referência a um segredo

Você pode usar o console do Amazon ECS para criar uma definição de tarefa que faz referência a um segredo do Secrets Manager ou a um parâmetro do Repositório de parâmetros do Systems Manager.

Para criar uma definição de tarefa que especifica um segredo

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, escolha Task Definitions (Definições de tarefa), Create new Task Definition (Criar nova definição de tarefa).
3. Na página Select launch type compatibility (Selecionar compatibilidade do tipo de inicialização), escolha o tipo de inicialização das tarefas e escolha Next step (Próxima tarefa).
4. Em Task Definition Name (Nome da definição da tarefa), digite um nome para a sua definição de tarefa. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.
5. Em Task execution role (Função de execução da tarefa), selecione a função de execução da tarefa existente ou selecione Create new role (Criar outra função) para que ela seja criada para você. Essa função autoriza o Amazon ECS a extrair imagens privadas da sua tarefa. Para obter mais informações, consulte Permissões do IAM necessárias para a autenticação de registro privado (p. 71).

   Important

   Se o campo Task execution role (Função de execução de tarefas) não aparecer, escolha Configure via JSON (Configurar via JSON) e adicione manualmente o campo `executionRoleArn` para especificar a função de execução da tarefa. O código a seguir mostra a sintaxe:

   ```
   "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole"
   ```

6. Para cada contêiner a ser criado em sua definição de tarefa, conclua as seguintes etapas:

   a. Em Container Definitions (Definições de contêiner), selecione Add container (Adicionar contêiner).

   b. Em Container name (Nome do contêiner), digite um nome para o contêiner. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.

   c. Em Image (Imagem), digite o nome da imagem ou o caminho para sua imagem privada. São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados.

   d. Expanda Advanced container configuration (Configuração de contêiner avançada).

   e. Para segredos de contêiner referenciados como variáveis de ambiente, em Environment (Ambiente), em Environment variables (Variáveis de ambiente), preencha os seguintes campos:

      i. Em Key (Chave), insira o nome da variável de ambiente a ser definida no contêiner. Isso corresponde ao campo `name` na seção `secrets` de definição de contêiner.

      ii. Em Value (Valor), escolha ValueFrom. Em Add value (Adicionar valor), insira o ARN completo do segredo do Secrets Manager ou o nome ou ARN completo do parâmetro do Repositório de parâmetros do AWS Systems Manager que contém os dados a serem apresentados ao contêiner como uma variável de ambiente.

         Note

         Se o parâmetro Parameter Store do Systems Manager existir na mesma região da tarefa que você está iniciando, você poderá usar o ARN completo ou o nome do segredo. Se o parâmetro existir em uma região diferente, o ARN completo deverá ser especificado.

   f. Para segredos referenciados na configuração de log para um contêiner, em Storage and Logging (Armazenamento e registro em log), em Log configuration (Configuração de log), preencha os seguintes campos:

      i. Desmarque a opção Auto-configure CloudWatch Logs (Configurar automaticamente o &CWL;).

      ii. Em Log options (Opções de log), em Key (Chave), insira o nome da opção de configuração de log a ser definida.

      iii. Em Value (Valor), escolha ValueFrom. Em Add value (Adicionar valor), insira o ARN completo do segredo do Secrets Manager ou o nome ou ARN completo do parâmetro do Parameter Store do AWS Systems Manager que contém os dados a serem apresentados ao contêiner como uma variável de ambiente.

         Note

         Se o parâmetro do Store Parameter Store do Systems Manager existir na mesma região da tarefa que você está iniciando, você poderá usar o ARN completo ou o nome do segredo. Se o parâmetro existir em uma região diferente, o ARN completo deverá ser especificado.

   g. Preencha os demais campos obrigatórios e qualquer campo opcional a ser usado em suas definições de contêiner. Mais parâmetros de definição de contêiner estão disponíveis no menu Advanced container configuration (Configuração de contêiner avançada). Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34).

   h. Escolha Adicionar.

7. Quando seus contêineres forem adicionados, selecione Create (Criar).

Definições de tarefa de exemplo

Esta seção fornece alguns exemplos de definição de tarefa que você pode usar para começar a criar suas próprias definições de tarefa. Para obter mais informações, consulte Parâmetros de definição de tarefa (p. 34) e Como criar uma definição de tarefa (p. 29).

Tópicos

- Exemplo: servidor da Web (p. 79)
- Exemplo: driver de log do awslogs (p. 80)
- Exemplo: driver de log do splunk (p. 80)
- Exemplo: driver de log fluentd (p. 81)
- Exemplo: driver de log gelf (p. 81)
- Exemplo: dependência de contêiner (p. 82)

Exemplo: servidor da Web

Veja a seguir um exemplo de definição de tarefa usando o tipo de inicialização Fargate que configura um servidor web:

```
{
   "containerDefinitions": [
      {
         "command": [
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div
 style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!
</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></
html>' >  /usr/local/apache2/htdocs/index.html && httpd-foreground\""
         ],
         "entryPoint": [
            "sh",
            "-c"
         ],
         "essential": true,
         "image": "httpd:2.4",
         "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
               "awslogs-group" : "/ecs/fargate-task-definition",
               "awslogs-region": "us-east-1",
               "awslogs-stream-prefix": "ecs"
            }
         },
         "name": "sample-fargate-app",
         "portMappings": [
            {
               "containerPort": 80,
               "hostPort": 80,
               "protocol": "tcp"
            }
         ]
      }
   ],
   "cpu": "256",
   "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
   "family": "fargate-task-definition",
   "memory": "512",
   "networkMode": "awsvpc",
```

```
   "requiresCompatibilities": [
       "FARGATE"
    ]
}
```

Exemplo: driver de log do awslogs

O exemplo a seguir demonstra como usar o driver de log awslogs em uma definição de tarefa que usa o tipo de inicialização Fargate. O contêiner nginx envia os logs para o grupo de logs ecs-log-streaming na região us-west-2. Para obter mais informações, consulte Como usar o driver de log awslogs (p. 64).

```
{
  "containerDefinitions": [
    {
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "name": "nginx-container",
      "image": "nginx",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "ecs-log-streaming",
          "awslogs-region": "us-west-2",
          "awslogs-stream-prefix": "fargate-task-1"
        }
      },
      "cpu": 0
    }
  ],
  "networkMode": "awsvpc",
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "memory": "2048",
  "cpu": "1024",
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "family": "example_task_1"
}
```

Exemplo: driver de log do splunk

O exemplo a seguir demonstra como usar o driver de log splunk em uma definição de tarefa que envia os logs para um serviço remoto. O parâmetro de token do Splunk é especificado como uma opção secreta, pois ele pode ser tratado como dados confidenciais. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

```
"containerDefinitions": [{
  "logConfiguration": {
   "logDriver": "splunk",
   "options": {
    "splunk-url": "https://cloud.splunk.com:8080",
    "tag": "tag_name",
```

```
   },
   "secretOptions": [{
    "name": "splunk-token",
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:splunk-token-KnrBkD"
}],
```

Exemplo: driver de log `fluentd`

O exemplo a seguir demonstra como usar o driver de log `fluentd` em uma definição de tarefa que envia os logs para um serviço remoto. O valor `fluentd-address` é especificado como uma opção secreta, pois ele pode ser tratado como dados confidenciais. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

```
"containerDefinitions": [{
 "logConfiguration": {
  "logDriver": "fluentd",
  "options": {
   "tag": "fluentd demo"
  },
  "secretOptions": [{
   "name": "fluentd-address",
   "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:fluentd-address-
KnrBkD"
  }]
 },
 "entryPoint": [],
 "portMappings": [{
             "hostPort": 80,
             "protocol": "tcp",
             "containerPort": 80
             },
             {
  "hostPort": 24224,
  "protocol": "tcp",
  "containerPort": 24224
 }]
}],
```

Exemplo: driver de log `gelf`

O exemplo a seguir demonstra como usar o driver de log `gelf` em uma definição de tarefa que envia os logs para um host remoto executando o Logstash que leva logs Gelf como uma entrada. Para obter mais informações, consulte `logConfiguration` (p. 46).

```
"containerDefinitions": [{
 "logConfiguration": {
  "logDriver": "gelf",
  "options": {
   "gelf-address": "udp://logstash-service-address:5000",
   "tag": "gelf task demo"
  }
 },
 "entryPoint": [],
 "portMappings": [{
   "hostPort": 5000,
   "protocol": "udp",
   "containerPort": 5000
  },
  {
   "hostPort": 5000,
```

```
   "protocol": "tcp",
   "containerPort": 5000
  }
 ]
}],
```

Exemplo: dependência de contêiner

Este exemplo demonstra a sintaxe para uma definição de tarefa com vários contêineres em que a dependência de contêiner é especificada. Na definição de tarefa a seguir, o contêiner `envoy` deve alcançar um status íntegro, determinado pelos parâmetros de verificação de integridade de contêiner necessários, antes de o contêiner `app` ser iniciado. Para obter mais informações, consulte Dependência de contêiner (p. 51).

```
{
  "family": "appmesh-gateway",
  "proxyConfiguration":{
      "type": "APPMESH",
      "containerName": "envoy",
      "properties": [
          {
              "name": "IgnoredUID",
              "value": "1337"
          },
          {
              "name": "ProxyIngressPort",
              "value": "15000"
          },
          {
              "name": "ProxyEgressPort",
              "value": "15001"
          },
          {
              "name": "AppPorts",
              "value": "9080"
          },
          {
              "name": "EgressIgnoredIPs",
              "value": "169.254.170.2,169.254.169.254"
          }
      ]
  },
  "containerDefinitions": [
    {
      "name": "app",
      "image": "application_image",
      "portMappings": [
        {
          "containerPort": 9080,
          "hostPort": 9080,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "dependsOn": [
        {
          "containerName": "envoy",
          "condition": "HEALTHY"
        }
      ]
    },
    {
      "name": "envoy",
```

```
      "image": "840364872350.dkr.ecr.us-west-2.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-
prod",
      "essential": true,
      "environment": [
        {
          "name": "APPMESH_VIRTUAL_NODE_NAME",
          "value": "mesh/meshName/virtualNode/virtualNodeName"
        },
        {
          "name": "ENVOY_LOG_LEVEL",
          "value": "info"
        }
      ],
      "healthCheck": {
        "command": [
          "CMD-SHELL",
          "echo hello"
        ],
        "interval": 5,
        "timeout": 2,
        "retries": 3
      }
    }
  ],
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "networkMode": "awsvpc"
}
```

Como atualizar uma definição de tarefa

Para atualizar uma definição de tarefa, crie uma revisão de definição de tarefa. Se a definição de tarefa for usada em um serviço, será necessário atualizá-lo para usar a definição de tarefa atualizada.

Para criar uma revisão de definição de tarefa

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região que contém a definição de tarefa.
3. No painel de navegação, selecione Task Definitions (Definições de tarefas).
4. Na página Task Definitions, selecione a caixa à esquerda da definição de tarefa a ser revisada e escolha Create new revision.
5. Na página Create new revision of Task Definition, faça as alterações. Por exemplo, para alterar as definições de contêiner existentes (como a imagem de contêiner, limites de memória ou mapeamentos de porta), selecione o contêiner, faça as alterações e escolha Update.
6. Verifique as informações e escolha Create.
7. Se a sua definição de tarefa for usada em um serviço, atualize o serviço com a definição de tarefa atualizada. Para obter mais informações, consulte Atualizar um serviço (p. 160).

Como cancelar o registro das definições de tarefa

Caso decida que não precisa mais de uma definição de tarefa no Amazon ECS, você pode cancelar o registro da definição de tarefa, de maneira que ela não seja mais exibida nas chamadas à API `ListTaskDefinition` ou no console quando deseja executar uma tarefa ou atualizar um serviço.

Quando você cancela o registro de uma definição de tarefa, ela é marcada imediatamente como `INACTIVE`. As tarefas e os serviços existentes que referenciam uma definição de tarefa `INACTIVE`

continuam sendo executados sem interrupção, e os serviços existentes que referenciam uma definição de tarefa `INACTIVE` ainda podem ser aumentados ou diminuídos modificando-se a contagem desejada do serviço.

Você não pode usar uma definição de tarefa `INACTIVE` para executar novas tarefas ou para criar novos serviços, e não pode atualizar um serviço existente para referenciar uma definição de tarefa `INACTIVE` (embora talvez haja uma janela de até 10 minutos após o cancelamento do registro em que essas restrições ainda não entraram em vigor).

> **Note**
>
> Neste momento, as definições de tarefa `INACTIVE` continuam detectáveis indefinidamente na conta. Porém, esse comportamento estará sujeito a alterações no futuro, de maneira que você não deve confiar em definições de tarefa `INACTIVE` que vão além do ciclo de vida de qualquer tarefa e serviço associados.

Use o procedimento a seguir para cancelar o registro de uma definição de tarefa.

Para cancelar o registro de uma definição de tarefa

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região que contém a definição de tarefa.
3. No painel de navegação, selecione Task Definitions (Definições de tarefas).
4. Na página Task Definitions, escolha o nome da definição de tarefa que contém uma ou mais revisões cujo registro você deseja cancelar.
5. Na página Task Definition Name, selecione a caixa à esquerda de cada revisão de definição de tarefa cujo registro deseja cancelar.
6. Escolha Actions e Deregister.
7. Verifique se as informações na janela Deregister Task Definition e escolha Deregister para terminar.

Configurações da conta

O Amazon ECS fornece as seguintes configurações de conta, que permitem que você aceite ou não recursos específicos.

Nomes de recursos da Amazon (ARNs) e IDs

Nomes de recursos: `serviceLongArnFormat`, `taskLongArnFormat` e `containerInstanceLongArnFormat`

O Amazon ECS está apresentando um novo formato para nomes de recursos da Amazon (ARNs) e IDs de recursos para serviços, tarefas e instâncias de contêiner do Amazon ECS. Você deve aceitar o novo formato para que cada tipo de recurso use funcionalidades como a marcação de recursos. Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).

CloudWatch Container Insights

Nome do recurso: `containerInsights`

O CloudWatch Container Insights coleta, agrega e resume métricas e logs de seus aplicativos conteinerizados e microsserviços. As métricas incluem a utilização de recursos, como CPU, memória, disco e rede. O Container Insights também fornece informações de diagnóstico, como falhas de reinicialização de contêiner, para ajudá-lo a isolar problemas e resolvê-los rapidamente. Você também pode definir alarmes do CloudWatch em métricas que o Container Insights coleta. Para obter mais informações, consulte CloudWatch Container Insights do Amazon ECS (p. 186).

Quando a configuração da conta `containerInsights` for escolhida, todos os novos clusters criados após a escolha terão o Container Insights habilitado, a menos que você o desabilite durante a criação do cluster. Os clusters individuais podem ser habilitados ou desabilitados durante a criação ou usando a API UpdateClusterSettings.

Para cada região, você pode aceitar ou recusar cada configuração no nível de conta ou em relação a um usuário ou função do IAM. As configurações de conta disponíveis para aceitação ou recusa incluem o novo formato de ID de recurso e ARN e o recurso de entroncamento awsvpc.

Há suporte para os seguintes cenários:

- Um usuário ou uma função do IAM pode aceitar ou recusar a conta de usuário individual.
- Um usuário ou uma função do IAM pode definir a configuração de seleção ou cancelamento padrão para todos os usuários na conta.
- O usuário raiz tem a opção de aceitar ou recusar qualquer função ou usuário específico do IAM na conta. Se a configuração da conta do usuário raiz for alterada, ela definirá o padrão para todos os usuários e funções do IAM para os quais nenhuma configuração de conta individual foi definida.

A opção de aceitação ou recusa deve ser selecionada para cada configuração de conta separadamente. O formato do ARN e do ID de um recurso será definido pelo status de aceitação do usuário ou da função do IAM que criou o recurso.

Somente recursos executados após a aceitação recebem o novo formato de ARN e ID do recurso . Nenhum recurso existente é afetado. Para que os serviços do Amazon ECS e tarefas façam a transição para os novos formatos de ARN e ID de recurso, a tarefa ou o serviço deverá ser recriado. Para fazer a transição de uma instância de contêiner para o novo formato do ARN e do ID do recurso , a instância de contêiner deve ser drenada e uma nova instância de contêiner deve ser registrada no cluster.

Note

As tarefas iniciadas por um serviço do Amazon ECS só podem receber o novo formato do ARN e do ID de recurso se o serviço foi criado em 16 de novembro de 2018 ou após essa data e se o usuário do IAM que criou o serviço optou pelo novo formato para tarefas.

Tópicos

- Nomes de recursos da Amazon (ARNs) e IDs (p. 86)
- Visualizar configurações da conta (p. 87)
- Como modificar configurações da conta (p. 88)

Nomes de recursos da Amazon (ARNs) e IDs

Quando recursos do Amazon ECS são criados, é atribuído a cada recurso um nome de recurso da Amazon (ARN) e umidentificador de recurso (ID) exclusivos. Se você estiver usando uma ferramenta de linha de comando ou a API do Amazon ECS para trabalhar com o Amazon ECS, os ARNs ou IDs de recursos serão necessários para determinados comandos. Por exemplo, se você estiver usando o comando stop-task da AWS CLI para interromper uma tarefa, deverá especificar o ARN ou o ID da tarefa no comando.

As seções a seguir descrevem como os formatos de ARN e ID do recurso estão sendo alterados. Para obter mais informações sobre a transição para os novos formatos, consulte Perguntas frequentes do Amazon Elastic Container Service.

Formato do nome de recurso da Amazon (ARN)

Alguns recursos têm um nome amigável, como um serviço chamado `production`. Em outros casos, você deve especificar um recurso usando o formato de nome de recurso da Amazon (ARN). O novo formato de ARN para tarefas, serviços e instâncias de contêiner do Amazon ECS inclui o nome do cluster. Para obter detalhes sobre a aceitação do novo formato de ARN, consulte Como modificar configurações da conta (p. 88).

Note

O novo formato do ARN não está disponível na região GovCloud (EUA-Leste).

A tabela a seguir mostra o formato atual (antigo) e o novo formato para cada tipo de recurso.

Tipo de recurso	ARN
Instância de contêiner	Antigo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:container-instance/`*`container-instance-id`* Novo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:container-instance/`*`cluster-name`*`/`*`container-instance-id`*
Serviço do Amazon ECS	Antigo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:service/`*`service-name`* Novo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:service/`*`cluster-name`*`/`*`service-name`*
Tarefa do Amazon ECS	Antigo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:task/`*`task-id`* Novo: `arn:aws:ecs:`*`region`*`:`*`aws_account_id`*`:task/`*`cluster-name`*`/`*`task-id`*

Tamanho do ID do recurso

Um ID de recurso assume a forma de uma combinação exclusiva de letras e números. Novos formatos de ID de recurso incluem IDs mais curtos para tarefas e instâncias de contêiner do Amazon ECS. O formato de ID de recurso antigo tinha 36 caracteres. Os novos IDs estão em um formato de 32 caracteres que não incluem hifens. Para obter detalhes sobre a aceitação do novo formato de ID de recurso, consulte Como modificar configurações da conta (p. 88).

Note

O novo formato do ID de recurso não está disponível na região GovCloud (EUA-Leste).

Cronograma

Há um período de aceitação dos novos formatos. Veja a seguir as datas importantes relacionadas a essa alteração.

- Da versão inicial do formato a 31 de março de 2019 – a capacidade de aceitação e recusa dos novos IDs de recurso e nome de recurso da Amazon (ARN) é disponibilizada por região. Todas as novas contas criadas recusam o novo formato por padrão.
- De 1º de abril de 2019 a 31 de dezembro de 2019 – todas as novas contas aceitarão o novo formato por padrão. A opção de aceitação e recusa permanece disponível de acordo com a região.
- 1º de janeiro de 2020 – todas as contas aceitarão o novo formato por padrão. Todos os novos recursos criados receberão o novo formato.

Você pode optar por aceitar ou recusar o novo formato do nome de recurso da Amazon (ARN) e do ID de recurso a qualquer momento durante o período de aceitação. Depois de ter optado por aceitar, todos os novos recursos que você criar usarão o novo formato.

Note

Os IDs de recursos não mudam depois que são criados. Portanto, aceitar ou recusar o novo formato durante o período de aceitação não afeta seus IDs de recursos existentes.

Visualizar configurações da conta

É possível usar as ferramentas Console de gerenciamento da AWS e AWS CLI para visualizar os tipos de recursos que oferecem suporte aos novos formatos do ARN e do ID .

Para visualizar as configurações da conta usando o console

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação na parte superior da tela, selecione a região da qual deseja visualizar as configurações da conta.
3. No painel, escolha Account Settings (Configurações da conta).
4. Nas seções Amazon ECS ARN and resource ID settings (Configurações de ARNs e IDs de recursos do ARN do Amazon ECS) e CloudWatch Container Insights, você pode visualizar o status de suas escolhas para cada configuração de conta para o usuário e função do IAM autenticados.

Para visualizar as configurações da conta usando a linha de comando

Use um dos comandos a seguir para visualizar suas configurações de conta.

- list-account-settings (AWS CLI)

```
aws ecs list-account-settings --effective-settings --region us-east-1
```

- Get-ECSAccountSetting (AWS Tools para Windows PowerShell)

```
Get-ECSAccountSetting -EffectiveSetting true -Region us-east-1
```

Para visualizar as configurações de conta para um usuário do IAMou uma função do IAM em particular utilizando a linha de comando

Use um dos seguintes comandos e especifique o nome de recurso da Amazon (ARN) de um usuário do IAM, uma função do IAM ou uma conta de usuário raiz na solicitação para visualizar suas configurações de conta.

- list-account-settings (AWS CLI)

```
aws ecs list-account-settings --principal-arn
 arn:aws:iam::aws_account_id:user/principalName --effective-settings --region us-east-1
```

- Get-ECSAccountSetting (AWS Tools para Windows PowerShell)

```
Get-ECSAccountSetting -PrincipalArn arn:aws:iam::aws_account_id:user/principalName -
EffectiveSetting true -Region us-east-1
```

Como modificar configurações da conta

É possível usar as ferramentas Console de gerenciamento da AWS e AWS CLI para modificar as configurações da conta.

Para modificar as configurações da conta usando o console

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação na parte superior da tela, selecione a região da qual deseja modificar as configurações da conta.
3. No painel, escolha Account Settings (Configurações da conta).
4. Nas seções Amazon ECS ARN and resource ID settings (Configurações de ARN e ID de recurso do Amazon ECS) e CloudWatch Container Insights, você pode marcar ou desmarcar as caixas de seleção para cada configuração de conta para o usuário e função do IAM autenticados. Escolha Save (Salvar) quando terminar.

   Important

   Os usuários e as funções do IAM precisam da permissão `ecs:PutAccountSetting` para executar essa ação.
5. Na tela de confirmação, escolha Confirm (Confirmar) para salvar a seleção.

Para modificar as configurações de conta padrão para todos os usuários e funções do IAM na sua conta usando a linha de comando

Use um dos comando a seguir para modificar a configuração de conta padrão para todos os usuários ou funções do IAM na sua conta. Essas alterações se aplicarão a toda a conta da AWS, a menos que um usuário ou uma função do IAM cancele explicitamente essas configurações por conta própria.

- put-account-setting-default (AWS CLI)

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled --
region us-east-2
```

Você também pode usar esse comando para modificar as configurações de conta para todas as tarefas (`taskLongArnFormat`), instâncias de contêiner (`containerInstanceLongArnFormat`) e para aceitar os limites maiores da interface de rede elástica (ENI) para instâncias de contêiner (`awsvpcTrunking`). Para fazer isso, substitua o parâmetro `name` pelo tipo de recurso correspondente.

- Write-ECSAccountSetting (AWS Tools para Windows PowerShell)

```
Write-ECSAccountSettingDefault -Name serviceLongArnFormat -Value enabled -Region us-
east-1 -Force
```

Para modificar configurações de conta para sua conta de usuário do IAM usando a linha de comando

Use um dos comandos a seguir para modificar configurações de conta para o seu usuário do IAM. Se você estiver usando esses comandos como o usuário raiz, essas configurações se aplicarão a toda a conta da AWS, a menos que um usuário ou uma função do IAM cancele explicitamente essas configurações por conta própria.

- put-account-setting (AWS CLI)

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled --region us-
east-1
```

Você também pode usar esse comando para modificar as configurações de conta para todas as tarefas (`taskLongArnFormat`), instâncias de contêiner (`containerInstanceLongArnFormat`) e para aceitar os limites maiores da interface de rede elástica (ENI) para instâncias de contêiner (`awsvpcTrunking`). Para fazer isso, substitua o parâmetro `name` pelo tipo de recurso correspondente.

- Write-ECSAccountSetting (AWS Tools para Windows PowerShell)

```
Write-ECSAccountSetting -Name serviceLongArnFormat -Value enabled -Force
```

Para modificar as configurações de conta para um usuário do IAM ou uma função do IAM em particular utilizando a linha de comando

Use um dos comandos a seguir e especifique o nome de recurso da Amazon (ARN) de um usuário do IAM, uma função do IAM ou um usuário raiz na solicitação para modificar as configurações de conta para um usuário do IAM ou uma função do IAM em particular.

- put-account-setting (AWS CLI)

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled --principal-arn
 arn:aws:iam::aws_account_id:user/principalName --region us-east-1
```

Você também pode usar esse comando para modificar as configurações de conta para todas as tarefas (`taskLongArnFormat`), instâncias de contêiner (`containerInstanceLongArnFormat`) e para aceitar os limites maiores da interface de rede elástica (ENI) para instâncias de contêiner (`awsvpcTrunking`). Para fazer isso, substitua o parâmetro `name` pelo tipo de recurso correspondente.

- Write-ECSAccountSetting (AWS Tools para Windows PowerShell)

```
Write-ECSAccountSetting -Name serviceLongArnFormat -Value enabled -PrincipalArn
 arn:aws:iam::aws_account_id:user/principalName -Region us-east-1 -Force
```

Como programar tarefas do Amazon ECS

Amazon Elastic Container Service (Amazon ECS) é um sistema de estado compartilhado e simultaneidade otimista que oferece recursos de programação flexíveis para as tarefas e os contêineres. Os programadores do Amazon ECS utilizam as mesmas informações de estado do cluster fornecidas pela API do Amazon ECS para tomar decisões apropriadas de posicionamento.

Cada tarefa que usa o tipo de inicialização do Fargate tem seu próprio limite de isolamento e não compartilha o kernel, os recursos de CPU, os recursos de memória ou a interface de rede elástica subjacente com outra tarefa.

O Amazon ECS oferece um programador de serviço (para tarefas e aplicativos de longa duração), a capacidade de executar tarefas manualmente (para trabalhos em lote ou tarefas únicas), com o Amazon ECS posicionando tarefas no seu cluster para você. Você pode especificar as estratégias de posicionamento de tarefas e restrições que lhe permitem executar tarefas na configuração escolhida como, por exemplo, distribuídas em zonas de disponibilidade. É possível também integrar com programadores personalizados ou de terceiros.

Programador de serviço

O programador de serviço é ideal para serviços e aplicativos sem estado em execução por muito tempo. O programador de serviços garante que a estratégia de agendamento que você especifica seja seguida e as reprograma quando uma falha (por exemplo, caso a infraestrutura subjacente falhe por algum motivo).

Há duas estratégias de programador de serviços disponíveis:

- `REPLICA` — A estratégia de programação de réplica posiciona e mantém o número desejado de tarefas no seu cluster. Por padrão, o programador de serviços distribui as tarefas nas zonas de disponibilidade. Você pode usar estratégias de posicionamento de tarefas e restrições para personalizar as decisões de posicionamento de tarefas. Para obter mais informações, consulte Réplica (p. 100).
- `DAEMON` — A estratégia de programação do daemon implantará exatamente uma tarefa em cada instância de contêiner ativa que atender a todas as restrições de posicionamento de tarefas que você especificar no seu cluster. Ao usar essa estratégia, não há necessidade de especificar um número desejado de tarefas, uma estratégia de posicionamento de tarefas ou usar políticas de Auto Scaling do serviço. Para obter mais informações, consulte Daemon (p. 100).

  Note

  As tarefas Fargate não são compatíveis com a estratégia de programação do `DAEMON`.

O programador de serviços também se certifica de que as tarefas estejam registradas em um load balancer do Elastic Load Balancing. Você pode atualizar os serviços mantidos pelo programador de serviços, como implementar uma nova definição de tarefa ou alterar o número em execução de tarefas desejadas. Por padrão, o programador de serviço distribui tarefas entre zonas de disponibilidade, mas você pode usar estratégias de posicionamento da tarefa e restrições para personalizar decisões de posicionamento da tarefa. Para obter mais informações, consulte Serviços (p. 99).

Tarefas em execução manual

A ação `RunTask` é indicada para processos como trabalhos em lotes que realizam o trabalho e acabam parando. Por exemplo, você pode ter uma chamada de processo `RunTask` quando o trabalho entra em uma fila. A tarefa extrai o trabalho da fila, realiza o trabalho e acaba saindo. Usando `RunTask`, você pode

permitir que a estratégia de posicionamento da tarefa padrão distribua tarefas aleatoriamente pelo cluster, o que minimiza as chances de uma única instância obter um número desproporcional de tarefas. Você também pode usar `RunTask` para personalizar a maneira como o programador coloca tarefas usando estratégias de posicionamento da tarefa e restrições. Para obter mais informações, consulte Tarefas em execução (p. 91) e RunTask no Amazon Elastic Container Service API Reference.

Como executar tarefas em uma programação do tipo **`cron`**

Caso tenha tarefas a serem executadas em intervalos definidos no cluster, como uma operação de backup ou uma varredura de log, você pode usar o console do Amazon ECS para criar uma regra do Eventos do CloudWatch que executa uma ou mais tarefas no cluster nos momentos especificados. A regra do evento programado pode ser definida como um intervalo específico (executar a cada *N* minutos, horas ou dias) ou, para um agendamento mais complicado, você pode usar uma expressão `cron`. Para obter mais informações, consulte Tarefas programadas (`cron`) (p. 93).

Tópicos

- Tarefas em execução (p. 91)
- Tarefas programadas (cron) (p. 93)
- Desativação da tarefa (p. 96)
- Reciclagem de tarefas Fargate (p. 97)

Tarefas em execução

A execução manual das tarefas é ideal em determinadas situações. Por exemplo, suponhamos que você esteja desenvolvendo uma tarefa, mas não esteja pronto para implantar essa tarefa com o programador de serviço. Talvez a tarefa seja um trabalho em lotes único ou periódico que não faz sentido manter em execução ou reiniciar quando terminar.

Para manter um número de tarefas especificado em execução ou colocar as tarefas atrás de um load balancer, em vez disso, use o programador de serviços do Amazon ECS. Para obter mais informações, consulte Serviços (p. 99).

Tópicos

- Executar uma tarefa usando o tipo de inicialização Fargate (p. 91)

Executar uma tarefa usando o tipo de inicialização Fargate

Para executar uma tarefa usando o tipo de inicialização Fargate, faça o seguinte:

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, selecione Task Definitions e selecione a definição da tarefa a ser executada.
   - Para executar a revisão mais recente de uma definição de tarefa mostrada aqui, selecione a caixa à esquerda da definição de tarefa a ser executada.
   - Para executar uma revisão anterior de uma definição de tarefa mostrada aqui, selecione a definição de tarefa para ver todas as revisões ativas e, em seguida, selecione a revisão a ser executada.
3. Escolha Actions, Run Task.
4. Na seção Run Task (Executar tarefa), conclua as seguintes etapas:
   a. Para Launch type (Tipo de inicialização), escolha FARGATE. Para obter mais informações sobre tipos de inicialização, consulte Tipos de inicialização Amazon ECS (p. 59).

b. Para Platform version, escolha LATEST. Para obter mais informações sobre as versões da plataforma, consulte Versões de plataforma do AWS Fargate (p. 20).

c. Em Cluster, escolha o cluster a ser usado.

d. Em Number of tasks, digite o número de tarefas a serem executadas com essa definição de tarefa.

e. Em Task Group, digite o nome do grupo de tarefas.

5. Na seção VPC and security groups (VPC e grupos de segurança), conclua as seguintes etapas:

a. Para Cluster VPC (VPC do cluster), escolha a VPC a ser usada pelas tarefas. Certifique-se de que a VPC que você escolher não está configurada para exigir locação de hardware dedicada, pois isso não é compatível com as tarefas do Fargate.

b. Para Subnets, escolha as sub-redes disponíveis para sua tarefa.

c. Para Security groups, foi criado um security group para a sua tarefa que permite tráfego HTTP da Internet (0.0.0.0/0). Para editar o nome ou as regras deste security group, ou para escolher um security group existente, escolha Edit e modifique as configurações do seu security group.

d. Em Auto-assign public IP (Atribuir IP público automaticamente), escolha ENABLED (Habilitado) se você deseja atribuir um endereço IP público à interface de rede elástica que está anexada à tarefa do Fargate. Isso será necessário se a tarefa precisar de acesso à rede de saída, por exemplo, para obter uma imagem. Se o acesso à rede de saída não for obrigatório, você poderá escolher DISABLED (Desabilitado).

6. Na seção Advanced Options (Opções avançadas), conclua as seguintes etapas:

- (Opcional) Para enviar substituições de comando, variável de ambiente, uma função do IAM de tarefa ou execução de tarefa para um ou mais contêineres em sua definição de tarefa, escolha Advanced Options (Opções avançadas) e execute as seguintes etapas:

  Note

  Se você estiver usando os valores de parâmetro de sua definição de tarefa, não haverá necessidade de especificar substituições. Esses campos são usadas apenas para substituir os valores especificados na definição de tarefa.

  i. Em Task Role Override (Substituição de função de tarefa), escolha uma função do IAM para essa tarefa para substituir a função do IAM de tarefa especificada na definição de tarefa. Para obter mais informações, consulte Funções do IAM para tarefas (p. 242).

  Somente funções com o relacionamento de confiança `ecs-tasks.amazonaws.com` são mostradas aqui. Para obter mais informações sobre como criar uma função do IAM para suas tarefas, consulte Como criar uma função e uma política do IAM para suas tarefas (p. 243).

  ii. Em Task Execution Role Override (Substituição de função de execução de tarefa), escolha uma função de execução de tarefa para substituir a função de execução da tarefa especificada na definição de tarefa. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

  iii. Em Container Overrides (Substituições de contêiner), escolha um contêiner para enviar uma substituição de comando ou de variável de ambiente.

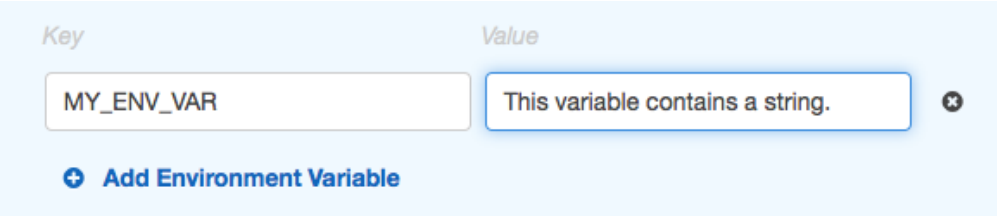
  - Para uma substituição de comando: em Substituição de comando, digite a substituição de comando a ser enviada. Se a sua definição de contêiner não especificar um `ENTRYPOINT`, o formato deve ser uma lista separada por vírgulas de strings sem aspas. Por exemplo:

    ```
    /bin/sh,-c,echo,$DATE
    ```

    Se a sua definição de contêiner especificar um `ENTRYPOINT` (como sh,-c), o formato deverá ser uma string sem aspas, cercada com aspas duplas e passada como um argumento para o comando `ENTRYPOINT`. Por exemplo:

```
while true; do echo $DATE > /var/www/html/index.html; sleep 1; done
```

- Para substituições de variável de ambiente: escolha Add Environment Variable. Em Key, digite o nome de sua variável de ambiente. Em Value, digite um valor de string para o seu valor de ambiente (sem aspas).



Essa substituição de variável de ambiente é enviada para o contêiner como:

```
MY_ENV_VAR="This variable contains a string."
```

7. In the Task tagging configuration section, complete the following steps:

   a. Select Enable ECS managed tags if you want Amazon ECS to automatically tag each task with the Amazon ECS managed tags. For more information, see Tagging Your Amazon ECS Resources.

   b. For Propagate tags from, select one of the following:

   - Do not propagate – This option will not propagate any tags.
   - Task Definitions – This option will propagate the tags specified in the task definition to the task.

     Note

     If you specify a tag with the same `key` in the Tags section, it will override the tag propagated from the task definition.

8. Na seção Tags, especifique a chave e o valor de cada tag para associá-la à tarefa. Para obter mais informações, consulte Como marcar seus recursos do Amazon ECS.
9. Revise as informações de sua tarefa e escolha Run Task.

   Note

   Se a sua tarefa mudar de `PENDING` para `STOPPED` ou se exibir o status `PENDING` e, em seguida, desaparecer das tarefas listadas, sua tarefa poderá ser interrompida devido a um erro. Para obter mais informações, consulte Como verificar se há erros em tarefas interrompidas (p. 310) na seção de solução de problemas.

Tarefas programadas (`cron`)

O Amazon ECS oferece suporte à capacidade de programar tarefas em um cronograma do tipo `cron` ou em uma resposta ao Eventos do CloudWatch. Isso é suportado para tarefas do Amazon ECS usando os tipos de execução Fargate e EC2.

Caso tenha tarefas a serem executadas em intervalos definidos no cluster, como uma operação de backup ou uma varredura de log, você pode usar o console do Amazon ECS para criar uma regra do Eventos do CloudWatch que executa uma ou mais tarefas no cluster nos momentos especificados. A regra do evento programado pode ser definida como um intervalo específico (executar a cada *N* minutos, horas ou dias) ou, para um agendamento mais complicado, você pode usar uma expressão `cron`. Para obter mais informações, consulte Programar expressões para regras no Guia do usuário do Eventos do Amazon CloudWatch.

Você também pode definir suas tarefas Fargate como um destino de tarefa no Eventos do CloudWatch, permitindo que você execute tarefas em resposta a alterações que ocorrerem. Além disso, você pode modificar a configuração de rede ao usar o modo de rede `awsvpc` por meio do console do Eventos do CloudWatch e da AWS CLI, fornecendo às tarefas Fargate acionadas pelo Eventos do CloudWatch as mesmas propriedades de rede que as instâncias do Amazon EC2. Para obter mais informações, consulte Tutorial: Executar uma tarefa do Amazon ECS quando um arquivo é carregado para um bucket do Amazon S3 no Guia do usuário do Eventos do Amazon CloudWatch.

Note

Esse recurso ainda não está disponível para tarefas Fargate nas seguintes regiões:

Nome da região	Região
Ásia-Pacífico (Hong Kong)	ap-east-1

Criação de uma tarefa programada

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Escolha o cluster no qual criar sua tarefa programada.
3. Na página Cluster: ***`cluster-name`***, escolha Scheduled Tasks, Create.
4. Em Schedule rule name, insira um nome exclusivo para a regra de programação. São permitidos até 64 letras, números, pontos, hífens e sublinhados.
5. (Opcional) Em Schedule rule description, digite uma descrição para a regra. Até 512 caracteres são permitidos.
6. Em Schedule rule type, escolha se deseja usar uma programação de intervalo fixo ou uma expressão `cron` para a sua regra de programação. Para obter mais informações, consulte Programar expressões para regras no Guia do usuário do Eventos do Amazon CloudWatch.
   - Em Run at fixed interval, insira o intervalo e a unidade para a programação.
   - Em Cron expression, insira a expressão `cron` para a programação da tarefa. Essas expressões têm seis campos obrigatórios e os campos são separados por um espaço em branco. Para obter mais informações e exemplos de expressões `cron`, consulte Expressões cron no Guia do usuário do Eventos do Amazon CloudWatch.
7. Crie um destino para a sua regra de programação.
   a. Em Target id (ID de destino), insira um identificador exclusivo para o destino. São permitidos até 64 letras, números, pontos, hífens e sublinhados.
   b. Para Launch type (Tipo de inicialização), escolha o tipo de inicialização para as tarefas no seu serviço. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).
   c. Em Task definition, escolha a família e a revisão (family:revision) da definição de tarefa a ser executada para esse destino.
   d. Em Platform version (Versão da plataforma), escolha a versão da plataforma a ser usada para esse destino. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

      Note

      As versões de plataforma só são aplicáveis a tarefas que usam o tipo de inicialização Fargate.
   e. Em Number of tasks, insira o número de instanciações da definição de tarefa especificada para execução no seu cluster quando a regra for executada.
   f. (Opcional) Em Task role override (Substituição da função de tarefa), escolha a função do IAM a ser usada para a tarefa em seu destino, em vez do padrão de definição da tarefa. Para obter

mais informações, consulte Funções do IAM para tarefas (p. 242). Somente funções com o relacionamento de confiança Amazon EC2 Container Service Task Role são mostradas aqui. Para obter mais informações sobre como criar uma função do IAM para suas tarefas, consulte Como criar uma função e uma política do IAM para suas tarefas (p. 243). Você deve adicionar permissões `iam:PassRole` às substituições de funções de tarefa para as funções CloudWatch do IAM. Para obter mais informações, consulte Função do IAM Eventos do CloudWatch (p. 239).

g. Se a definição de tarefa da tarefa programada usar o modo de rede `awsvpc`, você deve fazer as configurações de VPC, sub-rede e do grupo de segurança para a tarefa programada. Para obter mais informações, consulte Redes de tarefas com o modo de rede do `awsvpc` (p. 62).

   i. Em Cluster VPC (VPC de cluster), se tiver selecionado o tipo de inicialização EC2, escolha a VPC na qual as instâncias de contêineres residem. Se tiver selecionado o tipo de inicialização Fargate, selecione a VPC que as tarefas de Fargate deveriam usar. Certifique-se de que a VPC escolhida não está configurada para exigir locação de hardware dedicada, porque isso não é compatível com tarefas de Fargate.

   ii. Para Subnets (Sub-redes), escolha as sub-redes disponíveis para posicionamento da sua tarefa programada.

   > **Important**
   >
   > Somente as sub-redes privadas são compatíveis com o modo de rede `awsvpc`. Como as tarefas não recebem endereços IP públicos, um gateway NAT é necessário para o acesso à Internet de saída, e o tráfego de Internet de entrada deve ser roteado por meio de um load balancer.

   iii. Em Security groups (Grupos de segurança), foi criado um grupo de segurança para as tarefas programadas, que permitem tráfego HTTP da Internet (`0.0.0.0/0`). Para editar o nome ou as regras deste security group, ou para escolher um security group existente, escolha Edit e modifique as configurações do seu security group.

   iv. Para Auto-assign Public IP, defina se suas tarefas devem receber um endereço IP público. Se você estiver usando tarefas Fargate, será necessário que um endereço IP público seja atribuído à interface de rede elástica da tarefa, com um roteamento para a Internet ou um gateway NAT que possa rotear as solicitações para a Internet. Isso permite que a tarefa extraia as imagens do contêiner.

h. Em CloudWatch Events IAM role for this target (Função do IAM do CloudWatch Events para este alvo), escolha uma função de serviço do Eventos do CloudWatch existente (`ecsEventsRole`) que você pode ter criado. Ou escolha Create new role (Criar nova função) para criar a função necessária do IAM que permite ao Eventos do CloudWatch fazer chamadas para o Amazon ECS para executar tarefas em seu nome. Para obter mais informações, consulte Função do IAM Eventos do CloudWatch (p. 239).

   > **Important**
   >
   > Se suas tarefas programadas exigem o uso da função de execução da tarefa ou se elas utilizam uma substituição de função de tarefa, você deve adicionar as permissões `iam:PassRole` na função de execução de tarefa ou na substituição de função de tarefa para a função CloudWatch do IAM. Para obter mais informações, consulte Função do IAM Eventos do CloudWatch (p. 239).

i. (Opcional) Na seção Container overrides, você pode expandir os contêineres individuais e substituir o comando e/ou variáveis de ambiente para esse contêiner que são estabelecidos na definição de tarefa.

8. (Opcional) Para adicionar mais destinos (outras tarefas a serem executadas quando essa regra é executada), escolha Add targets e repita as subetapas anteriores para cada destino adicional.
9. Escolha Criar.

Para editar uma tarefa programada

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Escolha o cluster no qual editar sua tarefa programada.
3. Na página Cluster: ***cluster-name***, escolha Scheduled Tasks.
4. Selecione a caixa à esquerda da regra de programação a ser editada e escolha Edit.
5. Edite os campos a serem atualizados e escolha Update.

Desativação da tarefa

A desativação de tarefas do Amazon ECS afeta tarefas dos tipos de execução Fargate e EC2 e você será notificado por email da desativação pendente.

Uma tarefa pode ser programada para ser desativada nos seguintes cenários:

- A AWS detecta a falha irreparável do hardware subjacente que hospeda a tarefa.
- Sua tarefa usa o tipo de execução Fargate e está em execução em uma versão da plataforma que tenha uma vulnerabilidade de segurança que requer que você substitua as tarefas executando novas tarefas usando uma versão de plataforma corrigida.

Se sua tarefa estiver programada para desativação, você receberá um e-mail antes do evento com o ID e a data de desativação da tarefa. Esse e-mail é enviado para o endereço que está associado à sua conta, o mesmo endereço de e-mail que você usa para fazer login no Console de gerenciamento da AWS. Se você usar uma conta de e-mail que não verifica regularmente, use o AWS Personal Health Dashboard para determinar se alguma de suas tarefas está agendada para desativação. Para atualizar as informações de contato para sua conta, acesse a página Configurações da conta.

Quando uma tarefa atinge sua data de desativação programada, ela é interrompida ou encerrada pela AWS. Se a tarefa fizer parte de um serviço, ela será automaticamente interrompida e o programador do serviço iniciará uma nova no lugar. Se estiver usando tarefas autônomas, você receberá uma notificação da desativação da tarefa e deverá iniciar novas tarefas para substituí-las.

Como trabalhar com tarefas agendadas para desativação

Se a tarefa fizer parte de um serviço, ela será automaticamente interrompida, o programador do serviço iniciará uma nova no lugar depois de atingir a data de desativação programada. Se quiser atualizar as tarefas de serviço antes da data de desativação, você poderá usar as etapas a seguir. Para obter mais informações, consulte Atualizar um serviço (p. 160).

Para atualizar um serviço em execução (Console de gerenciamento da AWS)

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a Região em que seu cluster está localizado.
3. No painel de navegação, escolha Clusters.
4. Na página Clusters, selecione o nome do cluster no qual seu serviço reside.
5. Na página Cluster: ***name***, escolha Services.
6. Marque a caixa à esquerda do serviço para atualizar e escolha Update.
7. Na página Configure service, as informações do serviço são preenchidas antecipadamente. Selecione Force new deployment (Forçar nova implantação) e escolha Next step (Próxima etapa).

Note

Para tarefas que usam o tipo de inicialização Fargate, forçar uma nova implantação inicia novas tarefas usando a versão da plataforma corrigida. As tarefas não exigem que você selecione uma versão da plataforma diferente. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

8. Nas páginas Configure network (Configurar rede) e Set Auto Scaling (optional) (Definir Auto Scaling [opcional]), escolha Next step (Próxima etapa).
9. Escolha Update Service para terminar e atualizar o seu serviço.

Para atualizar um serviço em execução (AWS CLI)

1. Obtenha o ARN do serviço.

```
aws ecs list-services --cluster cluster_name --region region
```

Resultado:

```
{
    "serviceArns": [
        "arn:aws:ecs:region:aws_account_id:service/MyService"
    ]
}
```

2. Atualize o serviço, forçando uma nova implantação que implanta novas tarefas.

```
aws ecs update-service --service serviceArn --force-new-deployment --
cluster cluster_name --region region
```

Se você estiver usando uma tarefa autônoma, poderá iniciar uma tarefa nova para substituí-la. Para obter mais informações, consulte Tarefas em execução (p. 91).

Reciclagem de tarefas Fargate

A reciclagem de tarefas do Amazon ECS afeta apenas as tarefas que usam o Fargate e nenhuma notificação é enviada antes do evento de reciclagem.

Uma tarefa pode ser reciclada nos seguintes cenários:

- A tarefa está usando o tipo de inicialização Fargate e a versão de plataforma 1.3.0 ou posterior. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

  Note

  As tarefas Fargate que usam versões de plataforma antes de 1.3.0 não são afetadas.
- A tarefa é parte de um serviço do Amazon ECS. As tarefas autônomas não são afetadas pela reciclagem de tarefas, mas ainda podem ser programadas para inativação. Para obter mais informações, consulte Desativação da tarefa (p. 96).
- A AWS determina que existe uma razão para reciclar a tarefa, conforme descrição abaixo.

Quando a AWS determina que uma atualização de segurança ou de infraestrutura é necessária para uma tarefa Fargate, ela aplica os patches necessários na tarefa. A maioria desses patches será transparente e

a tarefa não precisará ser interrompida, mas às vezes poderá ser necessário reciclar a tarefa. A partir da plataforma Fargate de versão 1.3.0, todas as tarefas Fargate executadas como parte de um serviço podem ser interrompidas e uma nova iniciada pelo programador de serviços do Amazon ECS para fornecer mais segurança e disponibilidade para a tarefa. A reciclagem da tarefa começará após 1º de fevereiro de 2019 e prosseguirá de forma contínua. O programador de serviço garantirá que a contagem desejada de tarefas para o seu serviço seja mantida.

Para se preparar para esse novo processo, recomendamos testar o comportamento de seu aplicativo simulando este cenário. Você pode fazer isso interrompendo uma tarefa individual em seu serviço para testar sua resiliência.

Serviços

Amazon ECS permite executar e manter simultaneamente um número especificado de instâncias de uma definição de tarefa em um cluster de Amazon ECS. Chamamos isso de um serviço. Se qualquer uma de suas tarefas falharem ou pararem por algum motivo, o programador de serviço Amazon ECS iniciará outra instância de sua definição de tarefa para substitui-la e mantém a contagem desejada de tarefas no serviço dependendo da estratégia de agendamento utilizada.

Além de manter a contagem desejada de tarefas no serviço, outra opção é executar o serviço por meio de um load balancer. O load balancer distribui o tráfego entre as tarefas associadas ao serviço.

Tópicos

- Conceitos do programador de serviço (p. 99)
- Conceitos de serviços adicionais (p. 100)
- Parâmetros de definição de serviço (p. 101)
- Tipos de implantação do Amazon ECS (p. 107)
- Balanceamento de carga do serviço (p. 115)
- Serviço Auto Scaling (p. 129)
- Descoberta de serviço (p. 137)
- Criar um serviço (p. 148)
- Atualizar um serviço (p. 160)
- Excluir um serviço (p. 163)
- Lógica de controle de serviço (p. 164)

Conceitos do programador de serviço

Se uma tarefa em um serviço for interrompida, ela será encerrada e uma nova tarefa iniciada. Esse processo continua até que o serviço atinja o número de tarefas em execução desejada com base na estratégia de programação que você especificou.

O programador de serviços inclui uma lógica que regula a frequência com que as tarefas são reiniciadas, caso elas falhem repetidamente ao tentar iniciar. Se uma tarefa for interrompida sem ter passado para um estado de `RUNNING`, o que é determinado por um time stamp `startedAt` na tarefa, o programador de serviços começa diminuir aos poucos as tentativas de inicialização e emite uma mensagem de evento de serviço. Esse comportamento impede a utilização de recursos desnecessários para tarefas com falha, dando a você uma oportunidade para resolver o problema. Depois que o serviço for atualizado, o programador de serviços continuará com seu comportamento normal. Para obter mais informações, consulte Lógica de controle de serviço (p. 164) e Mensagens de evento de serviço (p. 312).

Há duas estratégias de programador de serviços disponíveis:

- `REPLICA` — A estratégia de programação de réplica posiciona e mantém o número desejado de tarefas no seu cluster. Por padrão, o programador de serviços distribui as tarefas nas zonas de disponibilidade. Você pode usar estratégias de posicionamento de tarefas e restrições para personalizar as decisões de posicionamento de tarefas. Para obter mais informações, consulte Réplica (p. 100).
- `DAEMON` — A estratégia de programação do daemon implantará exatamente uma tarefa em cada instância de contêiner ativa que atender a todas as restrições de posicionamento de tarefas que você especificar no seu cluster. Ao usar essa estratégia, não há necessidade de especificar um número

desejado de tarefas, uma estratégia de posicionamento de tarefas ou usar políticas de Auto Scaling do serviço. Para obter mais informações, consulte Daemon (p. 100).

Note

As tarefas Fargate não são compatíveis com a estratégia de programação do `DAEMON`.

Daemon

A estratégia de agendamento do daemon implanta exatamente uma tarefa em cada instância de contêiner ativa que atende a todas as restrições de posicionamento de tarefas especificado no seu cluster. Ao usar essa estratégia, não há necessidade de especificar um número desejado de tarefas, uma estratégia de posicionamento de tarefas ou usar políticas de Auto Scaling do serviço.

O programador de serviço do daemon não coloca quaisquer tarefas em instâncias que possuem status `DRAINING`. Se uma instância de contêiner transformar `DRAINING`, as tarefas daemon nela são interrompidas. Ele também monitora quando novas instâncias de contêiner são adicionadas ao seu cluster e adiciona as tarefas de daemon tarefas a elas.

Se `deploymentConfiguration` é especificado, o parâmetro de porcentagem máximo deve ser `100`. O valor padrão para um serviço daemon de `maximumPercent` é 100%. O valor padrão de um serviço daemon de `minimumHealthyPercent` é 0% para a AWS CLI, os SDKs da AWS e as APIs, e 50% para o Console de gerenciamento da AWS.

Note

O serviço de programador do daemon não é compatível com o uso de Classic Load Balancers.

Réplica

A estratégia de programação de réplica coloca e mantém o número desejado de tarefas em seu cluster. Por padrão, o programador de serviço distribui tarefas por zonas de disponibilidade. Você pode usar estratégias e limitações de posicionamento de tarefas para personalizar decisões de realização de tarefa.

Quando o programador do serviço, usando a estratégia `REPLICA`, iniciar novas tarefas ou parar de executar tarefas que usam o tipo de inicialização Fargate, ele tentará manter o equilíbrio entre zonas de disponibilidade no seu serviço.

Conceitos de serviços adicionais

- Você também pode executar o serviço por trás de um load balancer. Para obter mais informações, consulte Balanceamento de carga do serviço (p. 115).
- É possível especificar uma configuração de implantação para seu serviço. Durante uma implantação, que é desencadeada atualizando a definição de tarefa ou a contagem desejada de um serviço, o programador de serviço usa os parâmetros de porcentagem íntegra mínima e a porcentagem íntegra máxima para determinar a estratégia de implantação. Para obter mais informações, consulte Parâmetros de definição de serviço (p. 101).
- Você pode configurar o serviço para usar o Amazon ECS descoberta de serviço. Descoberta de serviço usa APIs de nomeação automática Amazon Route 53 a fim de gerenciar entradas de DNS para tarefas do serviço, tornando-as detectáveis na VPC. Para obter mais informações, consulte Descoberta de serviço (p. 137).
- Ao excluir um serviço, se ainda houver tarefas em execução que exijam limpeza, o status do serviço mudará de `ACTIVE` para `DRAINING`, e não será possível visualizar o serviço no console ou na operação de API ListServices. Depois que todas as tarefas tiverem passado para o status `STOPPING` ou `STOPPED`,

o status do serviço mudará de `DRAINING` para `INACTIVE`. Serviços no status `INACTIVE` ou `DRAINING` ainda poderão ser visualizados com a operação de API DescribeServices. No entanto, no futuro, os serviços `INACTIVE` poderão ser limpos e removidos da manutenção de registros do Amazon ECS, e as chamadas DescribeServices nesses serviços retornarão um erro `ServiceNotFoundException`.

Parâmetros de definição de serviço

Uma definição de serviço define qual definição de tarefa usar com o serviço, quantas instanciações dessa tarefa executar e quais load balancers (se houver) associar às suas tarefas, além de outros parâmetros de serviços.

```
{
    "cluster": "",
    "serviceName": "",
    "taskDefinition": "",
    "loadBalancers": [
        {
            "targetGroupArn": "",
            "loadBalancerName": "",
            "containerName": "",
            "containerPort": 0
        }
    ],
    "serviceRegistries": [
        {
            "registryArn": "",
            "port": 0,
            "containerName": "",
            "containerPort": 0
        }
    ],
    "desiredCount": 0,
    "clientToken": "",
    "launchType": "EC2",
    "platformVersion": "",
    "role": "",
    "deploymentConfiguration": {
        "maximumPercent": 0,
        "minimumHealthyPercent": 0
    },
    "placementConstraints": [
        {
            "type": "distinctInstance",
            "expression": ""
        }
    ],
    "placementStrategy": [
        {
            "type": "binpack",
            "field": ""
        }
    ],
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": [
                ""
            ],
            "securityGroups": [
                ""
            ],
            "assignPublicIp": "ENABLED"
        }
```

```
    },
    "healthCheckGracePeriodSeconds": 0,
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "ECS"
    },
    "tags": [
        {
            "key": "",
            "value": ""
        }
    ],
    "enableECSManagedTags": true,
    "propagateTags": "TASK_DEFINITION"
}
```

Note

É possível criar o modelo de definição de serviço acima com o seguinte comando AWS CLI .

```
aws ecs create-service --generate-cli-skeleton
```

É possível especificar os seguintes parâmetros em uma definição de serviço.

`cluster`

O nome abreviado ou o Um nome de recurso da Amazon (ARN) completo do cluster no qual executar o serviço. Se você não especificar um cluster, consideraremos o cluster `default`.

`serviceName`

O nome do serviço. Os nomes de serviço São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados. devem ser exclusivos em um cluster, mas é possível ter serviços nomeados da mesma forma em vários clusters dentro de uma região ou de várias regiões.

Exigido: sim

`taskDefinition`

O `family` e o `revision` (`family:revision`) ou o Um nome de recurso da Amazon (ARN) completo da definição de tarefa a ser executada no serviço. Se um `revision` não for especificado, a revisão mais recente `ACTIVE` será usada.

`loadBalancers`

Um objeto load balancer que representa os load balancers para uso com o serviço. Para serviços que usam um Balanceador de carga de aplicações ou Load balancer de rede, há um limite de cinco grupos de destino que você pode associar a um serviço.

Depois que você cria um serviço, o Nome de região da Amazon (ARN) do grupo de destino ou o nome do load balancer, o nome do contêiner e porta do contêiner especificados na definição do serviço não podem mais ser alterados.

Para Classic Load Balancers, esse objeto deve conter o nome do load balancer, o nome do contêiner (conforme aparece na definição de contêiner) e a porta do contêiner para identificador de acesso a partir do load balancer. Quando uma tarefa desse serviço é colocada em uma instância do contêiner, a instância do contêiner é registrada com o load balancer especificado aqui.

Para Balanceador de carga de aplicaçõess e Load balancer de redes, esse objeto deve conter o Nome de região da Amazon (ARN) do grupo de destino do load balancer, o nome do contêiner (conforme aparece na definição de contêiner) e a porta do contêiner para o identificador de acesso a partir do load balancer. Quando uma tarefa desse serviço é colocada em uma instância do contêiner,

combinação da instância do contêiner e a porta é registrada como um destino no grupo de destino especificado aqui.

`targetGroupArn`

O Nome de recurso da Amazon (ARN) completo do grupo de destino Elastic Load Balancing associado a um serviço.

`loadBalancerName`

O nome do load balancer.

`containerName`

O nome do contêiner (conforme aparece na definição de contêiner) para associar ao load balancer.

`containerPort`

A porta no contêiner para associar ao load balancer. Essa porta deve corresponder a um `containerPort` na definição de tarefa de serviço. As instâncias de contêiner devem permitir o tráfego de entrada em `hostPort` do mapeamento de porta.

`serviceRegistries`

Os detalhes da configuração de descoberta de serviço para o seu serviço. Para obter mais informações, consulte Descoberta de serviço (p. 137).

`registryArn`

O nome de recurso da Amazon (ARN) do registro de serviço. O registro de serviço compatível atualmente é Amazon Route 53 Auto Naming. Para obter mais informações, consulte o Serviço.

`port`

O valor da porta usado se o serviço de descoberta de serviços especificou um registro de SRV. Esse campo é necessário se o modo de rede `awsvpc` e registros de SRV são usados.

`containerName`

O nome do contêiner do valor, já especificado na definição de tarefa, a ser usado para o serviço de descoberta de serviço. Se a definição de tarefa que sua tarefa de serviço especifica usa o modo de rede `bridge` ou `host` você deve especificar uma combinação de `containerName` e `containerPort` da definição de tarefa. Se a definição de tarefa que sua tarefa de serviço especifica usa o modo de rede `awsvpc` e um registro do tipo SRV DNS é usado, você deve especificar uma combinação de `containerName` e `containerPort` ou um valor `port`, mas não ambos.

`containerPort`

O valor da porta, já especificado na definição de tarefa, a ser usado para o serviço de descoberta de serviço. Se a definição de tarefa que sua tarefa de serviço especifica usa o modo de rede `bridge` ou `host` você deve especificar uma combinação de `containerName` e `containerPort` da definição de tarefa. Se a definição de tarefa que sua tarefa de serviço especifica usa o modo de rede `awsvpc` e um registro do tipo SRV DNS é usado, você deve especificar uma combinação de `containerName` e `containerPort` ou um valor `port`, mas não ambos.

`desiredCount`

O número de instâncias de definição de tarefa específica para posicionar e manter em execução no cluster.

`clientToken`

Identificador exclusivo e que diferencia maiúsculas e minúsculas que você fornece para garantir a idempotência da solicitação. Até 32 caracteres ASCII são permitidos.

`launchType`

O tipo de inicialização no qual executar seu serviço. Os valores aceitos são `FARGATE` ou `EC2`. Se não for especificado um tipo de inicialização, o `EC2` será usado por padrão. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).

`platformVersion`

A versão da plataforma na qual suas tarefas no serviço estão em execução. Uma versão da plataforma só é especificada para tarefas que usam o tipo de inicialização Fargate. Se não for especificada, a versão mais recente (`LATEST`) será usada como padrão.

As versões da plataforma AWS Fargate são usadas para fazer referência a um ambiente de tempo de execução específico para a infraestrutura de tarefas do Fargate. Ao especificar a versão da plataforma `LATEST` quando estiver executando uma tarefa ou criando um serviço, você obtém a versão de plataforma mais atual disponível para suas tarefas. Ao escalar seu serviço, essas tarefas recebem a versão de plataforma especificada na implantação atual do serviço. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

`role`

O nome abreviado ou o ARN completo da função do IAM que permite que o Amazon ECS faça chamadas para o load balancer em seu nome. Esse parâmetro só será permitido se você estiver usando um load balancer com seu serviço e a definição de sua tarefa não usar o modo de rede `awsvpc`. Se você especificar o parâmetro `role`, você também deve especificar um objeto do load balancer com o parâmetro `loadBalancers`.

Se a função especificada tiver um caminho diferente de `/`, você deverá especificar a função completa de Nome de região da Amazon (ARN), isso é recomendado, ou prefixar o nome da função com o caminho. Por exemplo, se uma função com o nome `bar` tiver um caminho `/foo/`, você especificaria `/foo/bar` como o nome da função. Para obter mais informações, consulte Nomes amigáveis e caminhos no Guia do usuário do IAM.

> **Important**
>
> Se sua conta já tiver criado a função vinculada ao serviço do Amazon ECS, essa função será usada por padrão para o serviço, a menos que você especifique uma função aqui. A função vinculada ao serviço será necessária se a definição de sua tarefa usar o modo de rede awsvpc e, nesse caso, você não deve especificar uma função aqui. Para obter mais informações, consulte Usar funções vinculadas ao serviço do Amazon ECS (p. 216).

`deploymentConfiguration`

Parâmetros opcionais de implantação que controlam quantas tarefas são executadas durante a implantação e o pedido de encerramento e iniciação de tarefas.

`maximumPercent`

Se um serviço estiver usando o tipo de implantação de atualização contínua (`ECS`), o parâmetro `maximumPercent` representará um limite superior no número de tarefas do serviço que são permitidas no estado `RUNNING` ou `PENDING` durante a implantação, como uma porcentagem do `desiredCount` (arredondado para baixo para o valor inteiro mais próximo). Esse parâmetro permite que você defina o tamanho dos lotes de implantação. Por exemplo, se o serviço estiver usando o programador de serviço `REPLICA` e tiver um `desiredCount` de quatro tarefas e um valor `maximumPercent` de 200%, o programador poderá iniciar quatro novas tarefas antes de interromper as quatro tarefas mais antigas (desde que os recursos de cluster necessários para fazer isso estejam disponíveis). O valor padrão `maximumPercent` para um serviço que usa o programador de serviço `REPLICA` é 200%.

Se o seu serviço estiver usando o tipo de programador de serviço `DAEMON`, `maximumPercent` deverá permanecer em 100%, que é o valor padrão.

O número máximo de tarefas durante uma implantação é `desiredCount` multiplicado por `maximumPercent`/100, arredondado para o valor inteiro mais próximo.

Se um serviço estiver usando os tipos de implantação azul/verde (`CODE_DEPLOY`) ou `EXTERNAL` e tarefas que usam o tipo de inicialização EC2, o valor de porcentagem máxima será definido como o valor padrão e será usado para definir o limite superior para o número de tarefas do serviço que permanecem no estado `RUNNING`, enquanto as instâncias de contêiner estarão no estado `DRAINING`. Se as tarefas do serviço usarem o tipo de inicialização Fargate, o valor de porcentagem máxima não será usado, embora ele seja retornado durante a descrição do seu serviço.

`minimumHealthyPercent`

Se um serviço estiver usando o tipo de implantação de atualização contínua (`ECS`), o parâmetro `minimumHealthyPercent` representará um limite inferior para o número de tarefas do serviço que devem permanecer no estado `RUNNING` durante uma implantação, como uma porcentagem de `desiredCount` (arredondado para cima para o número inteiro mais próximo). Esse parâmetro permite implantar sem usar a capacidade adicional de cluster. Por exemplo, se o serviço tiver um `desiredCount` de quatro tarefas e um `minimumHealthyPercent` de 50%, o programador de serviço poderá interromper duas tarefas existentes para liberar a capacidade do cluster antes de iniciar duas novas tarefas.

As tarefas para serviços que não usam um load balancer serão consideradas íntegras se estiverem no estado `RUNNING`.

As tarefas para serviços que usam um load balancer são consideradas íntegras caso estejam no estado `RUNNING`, sejam aprovadas em todas as verificações de integridade definidas e sejam relatadas como íntegras no load balancer ou no grupo de destino.

O valor padrão de um serviço de réplica de `minimumHealthyPercent` é 50% no Console de gerenciamento da AWS e 100% para a AWS CLI, os AWS SDKs e as APIs. O valor padrão `minimumHealthyPercent` para um serviço que usa o programador de serviço `DAEMON` é 0% para a AWS CLI, os AWS SDKs e as APIs, e 50% para o Console de gerenciamento da AWS.

O número mínimo de tarefas íntegras durante uma implantação é `desiredCount` multiplicado por `minimumHealthyPercent`/100, arredondado para o valor inteiro mais próximo acima.

Se um serviço estiver usando os tipos de implantação azul/verde (`CODE_DEPLOY`) ou `EXTERNAL` e tarefas que usam o tipo de inicialização EC2, o valor de porcentagem mínima de integridade será definido como o valor padrão e será usado para definir o limite inferior para o número de tarefas do serviço que permanecem no estado `RUNNING`, enquanto as instâncias de contêiner estarão no estado `DRAINING`. Se as tarefas do serviço usarem o tipo de inicialização Fargate, o valor de porcentagem mínima de integridade não será usado, embora ele seja retornado ao descrever o seu serviço.

`placementStrategy`

A estratégia de posicionamento de objetos para usar em tarefas no serviço. É possível especificar um máximo de quatro regras de estratégia por serviço.

`type`

O tipo de estratégia de posicionamento. A estratégia de posicionamento `random` posiciona as tarefas de candidatos disponíveis aleatoriamente. A estratégia de posicionamento `spread` distribui o posicionamento entre os candidatos disponíveis uniformemente com base no parâmetro do `field`. A estratégia `binpack` posiciona as tarefas em candidatos disponíveis que tenham a menor quantia disponível do recurso que está especificado no parâmetro do `field`. Por exemplo, se você binpack na memória, uma tarefa será posicionada na instância com a menor quantidade de memória remanescente (mas ainda o suficiente para executar a tarefa).

`field`

O campo em que a estratégia de posicionamento será aplicada. Para a estratégia de posicionamento `spread`, os valores válidos são `instanceId` (ou `host`, que tem o mesmo efeito), ou qualquer atributo de plataforma ou personalizado que seja aplicado em uma

instância de contêiner, como `attribute:ecs.availability-zone`. Para a estratégia de posicionamento `binpack`, os valores válidos são `cpu` e `memory`. Para a estratégia de posicionamento `random`, esse campo não é usado.

`networkConfiguration`

A configuração de rede para o serviço. Esse parâmetro é necessário para definições de tarefa que usam o modo de rede `awsvpc` para receber sua própria interface de rede elástica, sem haver suporte para outros modos de rede. Se for usar o tipo de inicialização Fargate, será necessário o modo de rede `awsvpc`. Para obter mais informações, consulte Redes de tarefas com o modo de rede do awsvpc (p. 62).

`awsvpcConfiguration`

Um objeto que representa as sub-redes e os security groups de uma tarefa ou serviço.

`subnets`

As sub-redes associadas à tarefa ou ao serviço.

`securityGroups`

Os security groups associados à tarefa ou ao serviço. Se você não especificar um security group, será usado o security group padrão da VPC.

`assignPublicIP`

Indica se a interface de rede elástica da tarefa recebe um endereço IP público.

`healthCheckGracePeriodSeconds`

O período, em segundos, que o programador de serviço do Amazon ECS deve ignorar verificações de integridade de destino do Elastic Load Balancing, verificações de integridade de contêiner e verificações de integridade do Route 53 que resultaram não íntegras depois que uma tarefa tiver sido iniciada pela primeira vez. Isso será válido somente se o serviço estiver configurado para usar um load balancer. Se as tarefas do seu serviço demoram para iniciar e responder às verificações de integridade, você pode especificar um período de carência de verificação de integridade de até 2.147.483.647 segundos durante o qual o programador do serviço ECS vai ignorar o status da verificação de integridade. Esse período de carência pode evitar que o programador do serviço ECS marque tarefas como não íntegras e as interrompa antes de terem tempo de surgir.

`schedulingStrategy`

A estratégia de programação para usar. Para obter mais informações, consulte Conceitos do programador de serviço (p. 99).

Há duas estratégias de programador de serviços disponíveis:

- `REPLICA` — A estratégia de programação de réplica posiciona e mantém o número desejado de tarefas no seu cluster. Por padrão, o programador de serviços distribui as tarefas nas zonas de disponibilidade. Você pode usar estratégias de posicionamento de tarefas e restrições para personalizar as decisões de posicionamento de tarefas. Para obter mais informações, consulte Réplica (p. 100).
- `DAEMON` — A estratégia de programação do daemon implantará exatamente uma tarefa em cada instância de contêiner ativa que atender a todas as restrições de posicionamento de tarefas que você especificar no seu cluster. Ao usar essa estratégia, não há necessidade de especificar um número desejado de tarefas, uma estratégia de posicionamento de tarefas ou usar políticas de Auto Scaling do serviço. Para obter mais informações, consulte Daemon (p. 100).

  Note

  As tarefas Fargate não são compatíveis com a estratégia de programação do `DAEMON`.

`deploymentController`

O controlador de implantação a ser usado para o serviço. Para obter mais informações, consulte Tipos de implantação do Amazon ECS (p. 107).

`type`

O tipo de controlador de implantação a ser usado. Existem três tipos de controlador de implantação disponíveis:

`ECS`

O tipo de implantação de atualização contínua (`ECS`) envolve substituir a versão atual em execução do contêiner pela versão mais recente. O número de contêineres que o Amazon ECS adiciona ou remove do serviço durante uma atualização contínua é controlado ajustando-se os números mínimo e máximo de tarefas íntegras permitidas durante uma implantação de serviço, conforme especificado em deploymentConfiguration.

`CODE_DEPLOY`

O tipo de implantação azul/verde (`CODE_DEPLOY`) usa o tipo de implantação azul/verde desenvolvido pela CodeDeploy, que permite que você verifique uma nova implantação de um serviço antes de enviar tráfego de produção para ele.

`EXTERNAL`

O tipo de implantação externa permite que você use qualquer controlador de implantação de terceiros para o controle total do processo de implantação para um serviço do Amazon ECS.

`tags`

Os metadados que você aplica ao serviço para ajudá-lo a categorizá-los e organizá-los. Cada tag consiste em uma chave e um valor opcional, ambos definidos por você. Quando um serviço é excluído, as tags também são excluídas. As chaves de tag podem ter no máximo 128 caracteres de caracteres e os valores da tag podem ter no máximo 256 caracteres. Para obter mais informações, consulte Marcação dos seus recursos do Amazon ECS (p. 165).

`key`

Uma parte de um par de chave/valor que compõe uma tag. Uma chave é um rótulo geral que age como uma categoria para valores de tag mais específicos.

`value`

A parte opcional de um par de chave/valor que compõe uma tag. Um valor atua como um descritor dentro de uma categoria de tag (chave).

`enableECSManagedTags`

Especifica se deve ativar as tags gerenciadas do Amazon ECS para as tarefas no serviço. Para obter mais informações, consulte Marcação dos seus recursos para faturamento (p. 167).

`propagateTags`

Especifica se deve copiar as tags da definição de tarefa ou do serviço para as tarefas no serviço. Se nenhum valor for especificado, as tags não serão copiadas. As tags só podem ser copiadas para tarefas no serviço durante a criação do serviço. Para adicionar tags a uma tarefa após a criação do serviço, use a ação de API `TagResource`.

Tipos de implantação do Amazon ECS

Um tipo de implantação do Amazon ECS determina a estratégia de implantação que seu serviço usa. Existem três tipos de implantação: atualização contínua, azul/verde e externa.

Tópicos

- Atualização contínua (p. 108)
- Implantação azul/verde com o CodeDeploy (p. 108)
- Implantação externa (p. 111)

Atualização contínua

O tipo de implantação atualização contínua é controlado pelo Amazon ECS. Isso envolve o programador de serviços substituindo a versão atual em execução do contêiner pela versão mais recente. O número de tarefas que o Amazon ECS adiciona ou remove do serviço durante uma atualização contínua é controlado pela configuração de implantação. Uma configuração de implantação consiste no número mínimo e máximo de tarefas permitidas durante uma implantação de serviço.

Para criar um novo serviço do Amazon ECS que use o tipo de implantação de atualização contínua, consulte Criar um serviço (p. 148).

Implantação azul/verde com o CodeDeploy

O tipo de implantação azul/verde usa o modelo de implantação azul/verde controlado pelo CodeDeploy. Esse tipo de implantação permite que você verifique uma nova implantação de um serviço antes de enviar tráfego de produção para ele. Para obter mais informações, consulte O que é CodeDeploy? no AWS CodeDeploy User Guide.

Veja a seguir os componentes do CodeDeploy que o Amazon ECS usa quando um serviço usa o tipo de implantação azul/verde:

Aplicativo do CodeDeploy

Uma coleção de recursos do CodeDeploy Isso consiste em um ou mais grupos de implantação.

Grupo de implantação do CodeDeploy

As configurações de implantação. Isso consiste no seguinte:

- Cluster e serviço do Amazon ECS
- Informações sobre o grupo de destino e o listener do load balancer
- Estratégia de reversão de implantação
- Configurações de reencaminhamento de tráfego
- Configurações de finalização de revisão original
- Configuração de implantação
- Configuração de alarmes do CloudWatch que pode ser definida para interromper implantações
- Configurações do Eventos do CloudWatch ou do SNS para notificações

Para obter mais informações, consulte Trabalhar com grupos de implantação no AWS CodeDeploy User Guide.

Configuração de implantação do CodeDeploy

Especifica como o CodeDeploy roteia o tráfego de produção para o seu conjunto de tarefas de substituição durante uma implantação. O único valor compatível neste momento é `CodeDeployDefault.AllAtOnce`, o que significa que todo o tráfego é roteado do conjunto de tarefas original para o conjunto de tarefas de substituição ao mesmo tempo. Para obter mais informações, consulte Trabalhar com configurações de implantação no AWS CodeDeploy User Guide.

Revisão

Uma revisão é o arquivo de especificação de aplicativo do CodeDeploy (arquivo AppSpec). No arquivo AppSpec, você especifica o ARN completo da definição da tarefa e o contêiner e a porta do conjunto de tarefas de substituição em que o tráfego deve ser roteado quando uma nova implantação é criada. O nome do contêiner deve ser um dos nomes de contêineres referenciados em sua definição de tarefa. Se a configuração de rede ou a versão da plataforma tiver sido atualizada na definição de serviço, você também deverá especificar esses detalhes no arquivo AppSpec. Você também pode especificar as funções do Lambda a serem executadas durante os eventos do ciclo de vida da implantação. As funções do Lambda permitem que você execute testes e retorne métricas durante

a implantação. Para obter mais informações, consulte Referência de arquivos AppSpec no AWS CodeDeploy User Guide.

Considerações sobre implantação azul/verde

Considere o seguinte ao usar o tipo de implantação azul/verde:

- Quando um serviço do Amazon ECS que usa o tipo de implantação azul/verde é criado inicialmente, um conjunto de tarefas do Amazon ECS será criado.
- Você deve configurar o serviço para usar um Balanceador de carga de aplicações ou um Load balancer de rede. Não há suporte para Classic Load Balancers. A seguir estão os requisitos do load balancer:
  - Você deve adicionar um listener de produção ao load balancer, que é usado para rotear o tráfego de produção.
  - Um listener de teste opcional pode ser adicionado ao load balancer, que é usado para rotear o tráfego de teste. Se você especificar um listener de teste, o CodeDeploy encaminhará seu tráfego de teste para o conjunto de tarefas de substituição durante uma implantação.
  - Os listeners de produção e teste devem pertencer ao mesmo load balancer.
  - É necessário definir um grupo de destino para o load balancer. O grupo de destino roteia o tráfego para o conjunto de tarefas original em um serviço por meio do listener de produção.
- Não há suporte para a escalabilidade automática do serviço ao usar o tipo de implantação azul/verde.
- Ao criar inicialmente um aplicativo e um grupo de implantação do CodeDeploy, você deve especificar o seguinte:
  - É necessário definir dois grupos de destino para o load balancer. Um grupo de destino deve ser o grupo de destino inicial definido para o load balancer quando o serviço do Amazon ECS for criado. O único requisito do segundo grupo de destino é que ele não pode ser associado a um load balancer diferente do que é usado pelo serviço.
- Quando você cria uma implantação do CodeDeploy para um serviço do Amazon ECS, o CodeDeploy cria um conjunto de tarefas de substituição (ou conjunto de tarefas verde) na implantação. Se você adicionou um listener de teste ao load balancer, o CodeDeploy roteará seu tráfego de teste para o conjunto de tarefas de substituição. É nesse momento que você pode executar quaisquer testes de validação. O CodeDeploy redireciona o tráfego de produção do conjunto de tarefas original para o conjunto de tarefas de substituição, de acordo com as configurações de redirecionamento de tráfego do grupo de implantação.

Experiência de console do Amazon ECS

Os fluxos de trabalho de atualização de serviços e de criação de serviços no console do Amazon ECS são compatíveis com implantações azuis/verdes.

Para criar um serviço do Amazon ECS que use o tipo de implantação azul/verde, consulte Criar um serviço (p. 148).

Para atualizar um serviço existente do Amazon ECS que esteja usando o tipo de implantação azul/verde, consulte Atualizar um serviço (p. 160).

Quando você usa o console do Amazon ECS para criar um serviço do Amazon ECS que use o tipo de implantação azul/verde, um conjunto de tarefas do Amazon ECS e os recursos do CodeDeploy a seguir são criados automaticamente com as configurações padrão a seguir.

Recurso	Configuração padrão
Nome do aplicativo	`AppECS-<cluster[:47]>-<service[:47]>`

Recurso	Configuração padrão
Nome do grupo de implantação	`DgpECS-<cluster[:47]>-<service[:47]>`
Informações sobre o load balancer do grupo de implantação	O listener de produção do load balancer, o listener de teste opcional e os grupos de destino especificados são adicionados à configuração do grupo de implantação.
Configurações de reencaminhamento de tráfego	Redirecionamento de tráfego – a configuração padrão é Reroute traffic immediately (Redirecionar o tráfego imediatamente). Você pode alterá-la no console do CodeDeploy ou atualizando o `TrafficRoutingConfig`. Para obter mais informações, consulte CreateDeploymentConfig na AWS CodeDeploy API Reference.
Configurações de finalização de revisão original	As configurações de finalização da revisão original são configuradas para aguardar 1 hora após o tráfego ter sido reencaminhado antes de finalizar o conjunto de tarefas azul.
Configuração de implantação	A configuração de implantação é definida como `CodeDeployDefault.AllAtOnce`, que roteia todo o tráfego de uma vez do conjunto de tarefas azul para o conjunto de tarefas verde. Você não pode alterar essa configuração.
Configuração de reversão automática	Se uma implantação falhar, as configurações de reversão automática serão configuradas para revertê-la.

Para visualizar detalhes de um serviço do Amazon ECS usando o tipo de implantação azul/verde, use a guia Deployments (Implantações) no console do Amazon ECS.

Para ver os detalhes de um grupo de implantação do CodeDeploy no console do CodeDeploy, consulte Visualizar detalhes do grupo de implantação com o CodeDeploy no AWS CodeDeploy User Guide.

Para modificar as configurações de um grupo de implantação do CodeDeploy no console do CodeDeploy, consulte Alterar configurações do grupo de implantação com o CodeDeploy no AWS CodeDeploy User Guide.

Permissões do IAM exigidas para implantação azul/verde

As implantações azul/verde do Amazon ECS são possíveis graças a uma combinação das APIs do Amazon ECS e do CodeDeploy. Os usuários do IAM devem ter as permissões apropriadas para esses serviços antes de poderem usar as implantações azuis/verdes do Amazon ECS no Console de gerenciamento da AWS ou com a AWS CLI ou SDKs.

Além das permissões do IAM padrão para criar e atualizar serviços, o Amazon ECS exige as permissões a seguir. Essas permissões foram adicionadas à política `AmazonECS_FullAccess` do IAM. Para obter mais informações, consulte AmazonECS_FullAccess (p. 222).

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

```
      "Action": [
        "codedeploy:CreateApplication",
        "codedeploy:CreateDeployment",
        "codedeploy:CreateDeploymentGroup",
        "codedeploy:GetApplication",
        "codedeploy:GetDeployment",
        "codedeploy:GetDeploymentGroup",
        "codedeploy:ListApplications",
        "codedeploy:ListDeploymentGroups",
        "codedeploy:ListDeployments",
        "codedeploy:StopDeployment",
        "codedeploy:GetDeploymentTarget",
        "codedeploy:ListDeploymentTargets",
        "codedeploy:GetDeploymentConfig",
        "codedeploy:GetApplicationRevision",
        "codedeploy:RegisterApplicationRevision",
        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment",
        "sns:ListTopics",
        "cloudwatch:DescribeAlarms",
        "lambda:ListFunctions"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Note

Além das permissões padrão do Amazon ECS necessárias para executar tarefas e serviços, os usuários do IAM também precisam de permissões `iam:PassRole` para usar funções do IAM para tarefas.

O CodeDeploy precisa de permissões para chamar APIs do Amazon ECS, modificar seu Elastic Load Balancing, invocar funções do Lambda e descrever alarmes do CloudWatch, além de permissões para modificar a contagem desejada do seu serviço para você. Antes de criar um serviço do Amazon ECS que use o tipo de implantação azul/verde, você deve criar uma função do IAM (`ecsCodeDeployRole`). Para obter mais informações, consulte Função do IAM CodeDeploy do Amazon ECS (p. 234).

Os exemplos de política Exemplo para criar serviço (p. 207) e Exemplo para atualizar serviço (p. 208) do IAM mostram as permissões que são necessárias para os usuários do IAM usarem implantações azul/verde do Amazon ECS no Console de gerenciamento da AWS.

Implantação externa

O tipo de implantação externa permite que você use qualquer controlador de implantação de terceiros para o controle total do processo de implantação para um serviço do Amazon ECS. Os detalhes do seu serviço são gerenciados por ações da API de gerenciamento de serviços (`CreateService`, `UpdateService` e `DeleteService`) ou ações da API de gerenciamento de conjuntos de tarefas (`CreateTaskSet`, `UpdateTaskSet`, `UpdateServicePrimaryTaskSet` e `DeleteTask`). Cada ação de API tem um subconjunto dos parâmetros de definição de serviço que ela pode gerenciar.

A ação de API `UpdateService` atualiza os parâmetros de contagem desejada e período de carência da verificação de integridade para um serviço. Se o tipo de inicialização, a versão da plataforma, os detalhes

do load balancer, a configuração de rede ou definição de tarefa precisarem ser atualizados, você deverá criar um novo conjunto de tarefas.

A ação de API `UpdateTaskSet` atualiza apenas o parâmetro de escala para um conjunto de tarefas.

A ação de API `UpdateServicePrimaryTaskSet` modifica qual conjunto de tarefas em um serviço é o conjunto de tarefas principal. Quando você chama a ação de API `DescribeServices`, ela retorna todos os campos especificados para um conjunto de tarefas principal. Se o conjunto de tarefas principal de um serviço for atualizado, qualquer valor de parâmetro de conjunto de tarefas existente no novo conjunto de tarefas principal que for diferente do conjunto de tarefas principal antigo em um serviço será atualizado para o novo valor quando um novo conjunto de tarefas principal for definido. Se nenhum conjunto de tarefas principal for definido para um serviço, durante a descrição do serviço, os campos do conjunto de tarefas serão nulos.

> **Important**
>
> Não há suporte para a escalabilidade automática do serviço ao usar um controlador de implantação externo.

A seguir está o fluxo de trabalho básico para gerenciar uma implantação externa no Amazon ECS.

Para gerenciar um serviço do Amazon ECS usando um controlador de implantação externo

1. Crie um serviço do Amazon ECS. O único parâmetro obrigatório é o nome do serviço. Você pode especificar os parâmetros a seguir ao criar um serviço usando um controlador de implantação externo. Todos os outros parâmetros de serviço são especificados durante a criação de um conjunto de tarefas no serviço.

   `serviceName`

   O nome do serviço. Os nomes de serviço São permitidos até 255 letras (caixa alta e baixa), números, hífens e sublinhados. devem ser exclusivos em um cluster, mas é possível ter serviços nomeados da mesma forma em vários clusters dentro de uma região ou de várias regiões.

   Exigido: sim

   `desiredCount`

   O número de instanciações da definição de tarefa do conjunto de tarefas especificado para posicionar e manter em execução no serviço.

   `deploymentConfiguration`

   Parâmetros opcionais de implantação que controlam quantas tarefas são executadas durante uma implantação e a ordem de interrupção e início de tarefas. Para obter mais informações, consulte deploymentConfiguration.

   `tags`

   Os metadados que você aplica ao serviço para ajudá-lo a categorizá-los e organizá-los. Cada tag consiste em uma chave e um valor opcional, ambos definidos por você. Quando um serviço é excluído, as tags também são excluídas. As chaves de tag podem ter no máximo 128 caracteres de caracteres e os valores da tag podem ter no máximo 256 caracteres. Para obter mais informações, consulte Marcação dos seus recursos do Amazon ECS (p. 165).

   `key`

   Uma parte de um par de chave/valor que compõe uma tag. Uma chave é um rótulo geral que age como uma categoria para valores de tag mais específicos.

   `value`

   A parte opcional de um par de chave/valor que compõe uma tag. Um valor atua como um descritor dentro de uma categoria de tag (chave).

`enableECSManagedTags`

Especifica se devem ser habilitadas as tags gerenciadas do Amazon ECS para as tarefas no serviço. Para obter mais informações, consulte Marcação dos seus recursos para faturamento (p. 167).

`propagateTags`

Especifica se deve copiar as tags da definição de tarefa ou do serviço para as tarefas no serviço. Se nenhum valor for especificado, as tags não serão copiadas. As tags só podem ser copiadas para tarefas no serviço durante a criação do serviço. Para adicionar tags a uma tarefa após a criação do serviço, use a ação de API `TagResource`.

`healthCheckGracePeriodSeconds`

O período, em segundos, que o programador de serviço do Amazon ECS deve ignorar verificações de integridade de destino do Elastic Load Balancing, verificações de integridade de contêiner e verificações de integridade do Route 53 que resultaram não íntegras depois que uma tarefa tiver sido iniciada pela primeira vez. Isso será válido somente se o serviço estiver configurado para usar um load balancer. Se as tarefas do seu serviço demoram para iniciar e responder às verificações de integridade, você pode especificar um período de carência de verificação de integridade de até 2.147.483.647 segundos durante o qual o programador do serviço ECS vai ignorar o status da verificação de integridade. Esse período de carência pode evitar que o programador do serviço ECS marque tarefas como não íntegras e as interrompa antes de terem tempo de surgir.

`schedulingStrategy`

A estratégia de programação para usar. Os serviços que usam um controlador de implantação externo oferecem suporte apenas à estratégia de programação de `REPLICA`. Para obter mais informações, consulte Conceitos do programador de serviço (p. 99).

`placementConstraints`

Um array de objetos de restrição de posicionamento para usar em tarefas no serviço. É possível especificar um máximo de 10 restrições por tarefa. Esse limite inclui restrições na definição de tarefa e naquelas especificadas no tempo de execução. Se você estiver usando o tipo de inicialização Fargate, não haverá suporte para restrições de posicionamento de tarefas.

`placementStrategy`

A estratégia de posicionamento de objetos para usar em tarefas no serviço. É possível especificar um máximo de quatro regras de estratégia por serviço.

Veja a seguir uma definição de serviço de exemplo para um serviço usando um controlador de implantação externo.

```
{
    "cluster": "",
    "serviceName": "",
    "desiredCount": 0,
    "role": "",
    "deploymentConfiguration": {
        "maximumPercent": 0,
        "minimumHealthyPercent": 0
    },
    "placementConstraints": [
        {
            "type": "distinctInstance",
            "expression": ""
        }
    ],
```

```
    "placementStrategy": [
        {
            "type": "binpack",
            "field": ""
        }
    ],
    "healthCheckGracePeriodSeconds": 0,
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "EXTERNAL"
    },
    "tags": [
        {
            "key": "",
            "value": ""
        }
    ],
    "enableECSManagedTags": true,
    "propagateTags": "TASK_DEFINITION"
}
```

2. Crie um conjunto de tarefas inicial. O conjunto de tarefas contém os seguintes detalhes sobre o serviço:

   `taskDefinition`

   A definição de tarefa para as tarefas do conjunto de tarefas usarem.

   `launchType`

   O tipo de inicialização no qual executar seu serviço. Os valores aceitos são `FARGATE` ou `EC2`. Se não for especificado um tipo de inicialização, o `EC2` será usado por padrão. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).

   `platformVersion`

   A versão da plataforma na qual suas tarefas no serviço estão em execução. Uma versão da plataforma só é especificada para tarefas que usam o tipo de inicialização Fargate. Se não for especificada, a versão mais recente (`LATEST`) será usada como padrão.

   As versões da plataforma AWS Fargate são usadas para fazer referência a um ambiente de tempo de execução específico para a infraestrutura de tarefas do Fargate. Ao especificar a versão da plataforma `LATEST` quando estiver executando uma tarefa ou criando um serviço, você obtém a versão de plataforma mais atual disponível para suas tarefas. Ao escalar seu serviço, essas tarefas recebem a versão de plataforma especificada na implantação atual do serviço. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

   `loadBalancers`

   Um objeto do load balancer que representa o load balancer para uso no serviço. Quando é usado um controlador de implantação externo, somente Balanceador de carga de aplicaçõess e Load balancer de redes têm suporte. Se você estiver usando um Balanceador de carga de aplicações, apenas um grupo de destino Balanceador de carga de aplicações será permitido por conjunto de tarefas.

   `networkConfiguration`

   A configuração de rede para o serviço. Esse parâmetro é necessário para definições de tarefas que usam o modo de rede `awsvpc` para receber sua própria interface de rede elástica, e não tem suporte para outros modos de rede. Para obter mais informações, consulte Redes de tarefas com o modo de rede do `awsvpc` (p. 62).

`serviceRegistries`

Os detalhes dos registros de descoberta de serviços a serem atribuídos ao serviço. Para obter mais informações, consulte Descoberta de serviço (p. 137).

`scale`

Uma porcentagem de ponto flutuante do número desejado de tarefas para posicionar e manter em execução no conjunto de tarefas. O valor é especificado como uma porcentagem total de `desiredCount` de um serviço. Os valores aceitos são números entre 0 e 100.

3. Quando forem necessárias alterações no serviço, use a ação de API `UpdateService`, `CreateTaskSet` ou `UpdateTaskSet`, dependendo de quais parâmetros você estiver atualizando. Se você tiver criado um conjunto de tarefas, use o parâmetro `scale` para cada tarefa em um serviço para determinar quantas tarefas devem ser mantidas em execução no serviço. Por exemplo, se você tiver um serviço que contenha `tasksetA` e criar um `tasksetB`, poderá testar a validade de `tasksetB` antes de querer fazer a transição do tráfego de produção para ele. Você poderia definir `scale` para os dois conjuntos de tarefas como `100`, e quando estivesse pronto para fazer a transição de todo o tráfego de produção para `tasksetB`, poderia atualizar `scale` para `tasksetA` como `0` para reduzi-lo.

Balanceamento de carga do serviço

O serviço do Amazon ECS também pode ser configurado para usar o Elastic Load Balancing a fim de distribuir o tráfego entre as tarefas no serviço por igual.

Os serviços do Amazon ECS oferecem suporte aos tipos de load balancer Balanceador de carga de aplicações, Load balancer de rede e Classic Load Balancer. Balanceador de carga de aplicaçõess são usados para rotear tráfego HTTP/HTTPS (ou Camada 7). Load balancer de redes e Classic Load Balancers são usados para rotear tráfego TCP (ou Camada 4). Para obter mais informações, consulte Tipos de load balancer (p. 117).

Os Balanceador de carga de aplicaçõess oferecem vários recursos que os tornam atrativos para uso com serviços do Amazon ECS:

- Cada serviço pode atender a tráfego de vários load balancers e expor várias portas de balanceamento de carga especificando vários grupos de destino.
- Esse recurso é compatível com tarefas que usam os tipos de execução Fargate e EC2.
- Os Balanceador de carga de aplicaçõess permitem que os contêineres usem mapeamento de porta host dinâmico (de maneira que várias tarefas do mesmo serviço sejam permitidas por instância de contêiner).
- Os Balanceador de carga de aplicaçõess dão suporte a regras de roteamento com base em caminho e prioridade (de maneira que vários serviços possam usar a mesma porta de escuta em um único Balanceador de carga de aplicações).

Recomendamos que você use Balanceador de carga de aplicaçõess nos seus serviços do Amazon ECS, para que possa aproveitar esses novos recursos, a menos que seu serviço exija um recurso que esteja disponível apenas com Load balancer de redes ou Classic Load Balancers. Para obter mais informações sobre o Elastic Load Balancing e as diferenças entre os dois tipos de load balancer, consulte o Guia do usuário do Elastic Load Balancing.

Tópicos

- Considerações sobre balanceamento de carga de serviço (p. 116)
- Tipos de load balancer (p. 117)
- Como criar um balanceador de carga (p. 119)
- Registro de vários grupos de destino com um serviço (p. 126)

Considerações sobre balanceamento de carga de serviço

Considere o seguinte ao usar o balanceamento de carga do serviço.

Considerações sobre o Balanceador de carga de aplicações e o Load balancer de rede

As seguintes considerações são específicas para serviços do Amazon ECS que usam Application Load Balancers ou Load balancer de redes:

- Para serviços que usam um Balanceador de carga de aplicações ou Load balancer de rede, não é possível associar mais de cinco grupos de destino a um serviço.
- Para serviços com tarefas que usam o modo de rede `awsvpc`, ao criar um grupo de destino para o serviço, é necessário escolher `ip` como o tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.
- Se o serviço que usar um Balanceador de carga de aplicações exigir acesso a várias portas de carga balanceada, como a porta 80 e a porta 443 para um serviço HTTP/HTTPS, será possível configurar dois listeners. Um listener é responsável pelo HTTPS que encaminha a solicitação para o serviço e outro listener que é responsável por redirecionar solicitações HTTP para a porta HTTPS apropriada. Para obter mais informações, consulte Criar um listener para o Balanceador de carga de aplicações no Guia do usuário para Application Load Balancers.
- Depois que você cria um serviço, o Nome de região da Amazon (ARN) ou o nome do load balancer, o nome do contêiner e porta do contêiner especificados na definição do serviço não podem mais ser alterados. Você não pode adicionar, remover ou alterar a configuração do load balancer de um serviço já existente. Se você atualizar a definição de tarefa para o serviço, o contêiner e a porta de contêiner especificados quando o serviço foi criado deverão permanecer na definição da tarefa.
- Caso a tarefa de um serviço não passe nos critérios de verificação de integridade do load balancer, a tarefa é interrompida e reiniciada. Esse processo continuará até que o serviço alcance o número de tarefas em execução desejadas.
- O modo de iniciação lenta do Balanceador de carga de aplicações tem suporte. Para obter mais informações, consulte Considerações sobre o modo de iniciação lenta do Balanceador de carga de aplicações (p. 116). Para obter mais informações sobre o modo de iniciação lenta, consulte Grupos de destino para seus Application Load Balancers.
- Caso você esteja enfrentando problemas com os serviços habilitados pelo balanceador de carga, consulte Como solucionar problemas de load balancers de serviço (p. 316).

Considerações sobre o modo de iniciação lenta do Balanceador de carga de aplicações

Balanceador de carga de aplicaçõess habilitados para o modo de iniciação lenta são compatíveis com os serviços do Amazon ECS. Para obter mais informações sobre o modo de inicialização lenta, consulte Grupos de destino para seus Application Load Balancers.

Para garantir que o programador de serviços ignore as verificações de integridade do contêiner não íntegras até que suas tarefas tenham sido preparadas e estejam prontas para receber tráfego, as seguintes configurações são necessárias:

- É necessário configurar a verificação de integridade do contêiner para retornar um status `UNHEALTHY` até que o período de iniciação lenta termine.

- É necessário configurar o valor do período de carência da verificação de integridade para seu serviço do Amazon ECS pela mesma duração do modo de iniciação lenta.

Considere o seguinte ao usar diferentes modos de rede de tarefas com modo de iniciação lenta do Balanceador de carga de aplicações:

- Ao usar o modo de rede `awsvpc`, cada tarefa recebe sua própria interface de rede elástica (ENI) e endereço IP, que permite que o Balanceador de carga de aplicações registre cada tarefa como um destino no grupo de destino. Isso permite que cada destino recém-registrado tenha o modo de inicialização lento habilitado.
- Ao usar o modo de rede `host`, a tarefa ignora as construções de rede do Docker e mapeia portas de contêiner diretamente para a interface de rede ou interfaces da instância do Amazon EC2. Registre a instância de contêiner como destino do Balanceador de carga de aplicações, em vez de o endereço IP da tarefa. Isso significa que é possível executar somente uma tarefa por instância se quiser que o modo de iniciação lenta funcione efetivamente. Se você atualizar uma tarefa ou serviço existente ou reiniciar a instância de contêiner, isso não registrará novamente a instância de contêiner como um destino do Balanceador de carga de aplicações, o que não causaria o início da duração da iniciação lenta.
- Ao usar o modo de rede `bridge`, de maneira semelhante ao modo de rede `host`, registre a instância de contêiner como destino do Balanceador de carga de aplicações, em vez da tarefa do Amazon ECS, para que as mesmas considerações descritas acima sejam aplicáveis.

Além disso, as seguintes considerações são específicas para usar o modo de iniciação lenta do Balanceador de carga de aplicações e adicionar tarefas do Amazon ECS como destinos:

- Ao habilitar a iniciação lenta para um grupo de destino, os destinos já registrados no grupo não entram no modo de iniciação lenta.
- Ao habilitar a iniciação lenta para um grupo de destino vazio e, em seguida, registrar um ou mais destinos usando uma única operação de registro, esses destinos não entram no modo de iniciação rápida. Os destinos recém-registrados entram no modo de iniciação lenta somente quando houver pelo menos um destino registrado que não esteja no modo de iniciação lenta.
- Se você cancelar o registro de um destino no modo de iniciação lenta, o destino sai do modo. Se registrar o mesmo destino novamente, ele entra no modo de iniciação lenta novamente.
- Se um destino no modo de iniciação lenta se tornar não íntegro e íntegro novamente antes do fim do período de duração, o destino permanece no modo de iniciação lenta até que o período de duração termine e saia do modo de iniciação lenta. Se um destino que não está no modo de iniciação lenta muda de não íntegro para íntegro, ele não entra no modo de iniciação lenta.

Tipos de load balancer

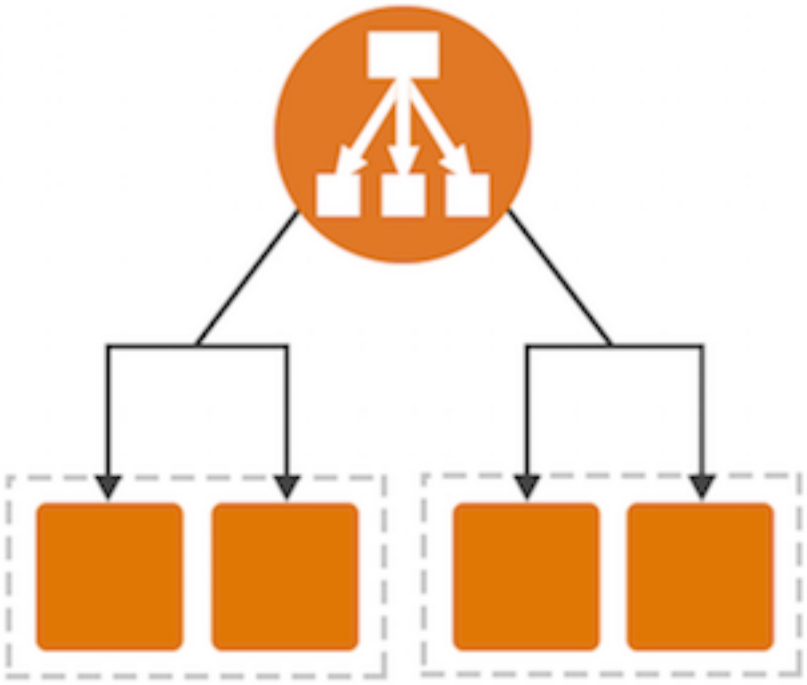
O Elastic Load Balancing dá suporte aos seguintes tipos de load balancers: Balanceador de carga de aplicaçõess, Load balancer de redes e Classic Load Balancers. Os serviços do Amazon ECS podem usar qualquer tipo de load balancer. Os Balanceador de carga de aplicaçõess são usados para rotear tráfego HTTP/HTTPS (Camada 7). Load balancer de redes e Classic Load Balancers são usados para rotear tráfego TCP (Camada 4).

Tópicos

- Balanceador de carga de aplicações (p. 118)
- Load balancer de rede (p. 118)

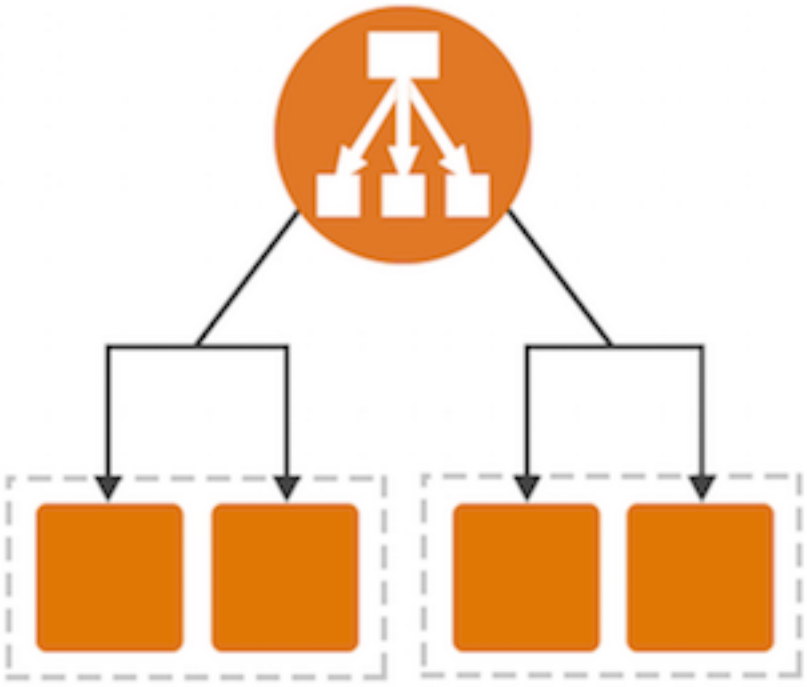
Balanceador de carga de aplicações

Um Balanceador de carga de aplicações toma decisões de roteamento na camada de aplicativo (HTTP/HTTPS), dá suporte ao roteamento com base em caminho e pode rotear solicitações para uma ou mais portas em cada instância de contêiner no cluster. Os Balanceador de carga de aplicaçõess dão suporte ao mapeamento de porta host dinâmico. Por exemplo, caso a definição de contêiner da tarefa especifique a porta 80 para uma porta de contêiner NGINX e a porta 0 para a porta host, a porta host é escolhida dinamicamente com base no intervalo de portas efêmero da instância de contêiner (como de 32768 a 61000 na Amazon ECS-optimized AMI mais recente). Quando a tarefa é ativada, o contêiner NGINX é registrado com o Balanceador de carga de aplicações como uma combinação de ID da instância e porta, e o tráfego é distribuído para o ID e a porta correspondentes a esse contêiner. Esse mapeamento dinâmico permite várias tarefas de um único serviço na mesma instância de contêiner. Para obter mais informações, consulte o Guia do usuário para Application Load Balancers.



Load balancer de rede

Um Load balancer de rede toma decisões de roteamento na camada de transporte (TCP/SSL). Ele pode lidar com milhões de solicitações por segundo. Após o load balancer receber uma conexão, ele seleciona um destino no grupo de destino para a regra padrão usando um algoritmo de roteamento de hash de fluxo. Ele tenta abrir uma conexão TCP para o destino selecionado na porta especificada na configuração do listener. Ele encaminha a solicitação sem modificar os cabeçalhos. Load balancer de redes dão suporte ao mapeamento da porta de host dinâmico. Por exemplo, caso a definição de contêiner da tarefa especifique a porta 80 para uma porta de contêiner NGINX e a porta 0 para a porta host, a porta host é escolhida dinamicamente com base no intervalo de portas efêmero da instância de contêiner (como de 32768 a 61000 na Amazon ECS-optimized AMI mais recente). Quando a tarefa é ativada, o contêiner NGINX é registrado com o Load balancer de rede como uma combinação de ID da instância e porta, e o tráfego é distribuído para o ID e a porta correspondentes a esse contêiner. Esse mapeamento dinâmico permite várias tarefas de um único serviço na mesma instância de contêiner. Para obter mais informações, consulte o Guia do usuário para Network Load Balancers.



Como criar um balanceador de carga

Esta seção apresenta uma introdução prática como usar o Elastic Load Balancing por meio do Console de gerenciamento da AWS a ser usado com os serviços do Amazon ECS. Nesta seção, você cria um load balancer externo que recebe o tráfego de rede pública e o roteia para as instâncias de contêiner do Amazon ECS.

O Elastic Load Balancing dá suporte aos seguintes tipos de load balancers: Balanceador de carga de aplicaçõess, Load balancer de redes e Classic Load Balancers. Os serviços do Amazon ECS podem usar qualquer tipo de load balancer. Os Balanceador de carga de aplicaçõess são usados para rotear tráfego HTTP/HTTPS. Load balancer de redes e Classic Load Balancers são usados para rotear tráfego TCP ou Camada 4.

Os Balanceador de carga de aplicaçõess oferecem vários recursos que os tornam atrativos para uso com serviços do Amazon ECS:

- Os Balanceador de carga de aplicaçõess permitem que os contêineres usem mapeamento de porta host dinâmico (de maneira que várias tarefas do mesmo serviço sejam permitidas por instância de contêiner).
- Os Balanceador de carga de aplicaçõess dão suporte a regras de roteamento com base em caminho e prioridade (de maneira que vários serviços possam usar a mesma porta de escuta em um único Balanceador de carga de aplicações).

Recomendamos que você use Balanceador de carga de aplicaçõess para os serviços do Amazon ECS, de maneira que você possa aproveitar esses recursos mais recentes. Para obter mais informações sobre o Elastic Load Balancing e as diferenças entre os dois tipos de load balancer, consulte o Guia do usuário do Elastic Load Balancing.

Antes de usar um load balancer com o serviço do Amazon ECS, sua conta já deve ter a função de serviço do Amazon ECS criada. Para obter mais informações, consulte Criar a função de serviço da conta (p. 120).

Tópicos

- Criar a função de serviço da conta (p. 120)
- Como criar um Balanceador de carga de aplicações (p. 121)
- Criar um Load balancer de rede (p. 125)

Criar a função de serviço da conta

O Amazon ECS precisa de permissões para registrar e cancelar o registro das instâncias de contêiner com o load balancer quando as tarefas são criadas e interrompidas.

Na maioria dos casos, a função de serviço do Amazon ECS é criada automaticamente para você na primeira execução do console do Amazon ECS. Você pode usar o procedimento a seguir para verificar e saber se a conta já tem uma função de serviço Amazon ECS.

Para verificar **`ecsServiceRole`** no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure `ecsServiceRole` na lista de funções. Se a função não existir, consulte Função do IAM programador de serviço do Amazon ECS (p. 232) para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
4. Escolha Permissions.
5. Na seção Políticas gerenciadas, certifique-se de que a política gerenciada AmazonEC2ContainerServiceRole esteja conectada à função. Se a política estiver anexada, sua função de serviço do Amazon ECS está configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para anexar a política.
   a. Escolha Attach Policy.
   b. Em Filter, digite **`AmazonEC2ContainerServiceRole`** para restringir as políticas disponíveis a serem anexadas.
   c. Selecione a caixa à esquerda da política AmazonEC2ContainerServiceRole e escolha Anexar política.
6. Escolha Trust Relationships (Relacionamentos de confiança), Edit Trust Relationship (Editar relacionamento de confiança).
7. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

Como criar um Balanceador de carga de aplicações

Esta seção orienta você em meio ao processo de criação de um Balanceador de carga de aplicações no Console de gerenciamento da AWS.

Defina o load balancer

Primeiro, forneça algumas informações sobre a configuração básica do load balancer, como nome, rede e escuta.

Escuta é um processo que verifica se há solicitações de conexão. Ele é configurado com um protocolo e uma porta para conexões front-end (cliente para load balancer), além de protocolo e uma porta para conexões back-end (load balancer para instância back-end). Neste exemplo, você configura uma escuta que aceita solicitações HTTP na porta 80 e as envia para os contêineres nas tarefas na porta 80 usando HTTP.

Para definir o load balancer

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Na barra de navegação, selecione uma região para seu load balancer. Selecione a mesma região selecionada para suas instâncias de contêiner do Amazon ECS.
3. No painel de navegação, em LOAD BALANCING, escolha Load balancers.
4. Selecione Criar load balancer.
5. Na página Select load balancer type (Selecionar tipo de load balancer), escolha Balanceador de carga de aplicações e Continue (Continuar).
6. Preencha a página Configurar Load Balancer conforme a seguir:

    a. Em Nome, insira um nome para o load balancer.
    b. Em Esquema, um load balancer voltado para a Internet roteará as solicitações de clientes pela Internet até os destinos. Um load balancer interno roteia solicitações a destinos usando endereços IP privados.
    c. No Tipo de endereço IP, selecione ipv4 para oferecer suporte somente aos endereços IPv4 ou dualstack para oferecer suporte aos endereços IPv4 e IPv6.
    d. Em Listeners, o padrão é um listener que aceite tráfego HTTP na porta 80. É possível manter as configurações do listener padrão, modificar o protocolo ou a porta do listener, ou ainda, selecionar Adicionar para inserir outro listener.

        Note

        Se você planeja rotear o tráfego para mais de um grupo de destino, consulte ListenerRules para obter detalhes sobre como adicionar regras baseadas em host ou caminho.

    e. Em VPC, selecione a mesma VPC que você usou para as instâncias de contêiner em que você pretende executar o serviço.
    f. Em Zonas de disponibilidade, marque a caixa de seleção para as zonas de disponibilidade a serem ativadas para seu load balancer. Se houver uma sub-rede para essa Zona de disponibilidade, ela estará selecionada. Se houver mais de uma sub-rede para essa Zona de disponibilidade, selecione uma delas. Você pode selecionar somente uma sub-rede por zona de disponibilidade. A configuração de sub-rede do load balancer deve incluir todas as zonas de disponibilidade nas quais as instâncias de contêiner residem.
    g. Selecione Próximo: Definir as configurações de segurança.

(Opcional) Configurar definições de segurança

Caso você tenha criado uma escuta segura na etapa anterior, preencha a página Configure Security Settings da maneira a seguir. Do contrário, escolha Next: Configure Security Groups.

Para definir as configurações de segurança

1. Se você tiver um certificado do AWS Certificate Manager, escolha Choose an existing certificate from AWS Certificate Manager (ACM) (Selecionar um certificado existente do AWS Certificate Manager (ACM) e o certificado em Certificate name (Nome do certificado).
2. Se você já tiver feito upload de um certificado usando o IAM, escolha Choose an existing certificate from AWS Identity and Access Management (IAM) (Selecionar um certificado existente do AWS Identity and Access Management [IAM]) e o seu certificado em Certificate name (Nome do certificado).
3. Se você tiver um certificado pronto para fazer upload, escolha Fazer upload de um novo certificado SSL para o AWS Identity and Access Management (IAM). Em Nome do certificado, digite um nome para o certificado. Na Chave privada, copie e cole o conteúdo do arquivo de chave privada (codificado por PEM). No Certificado de chave pública, copie e cole o conteúdo do arquivo do certificado de chave pública (codificado por PEM). Na Cadeia de certificados, copie e cole o conteúdo do arquivo da cadeia do certificado (codificado por PEM), exceto se estiver usando um certificado autoatribuído e se não for importante que os navegadores aceitem implicitamente o certificado.
4. Em Selecionar política, selecione uma política de segurança predefinida. Para obter detalhes sobre as políticas de segurança, consulte Políticas de segurança.
5. Selecione Próximo: Configurar security groups.

Configurar grupos de segurança

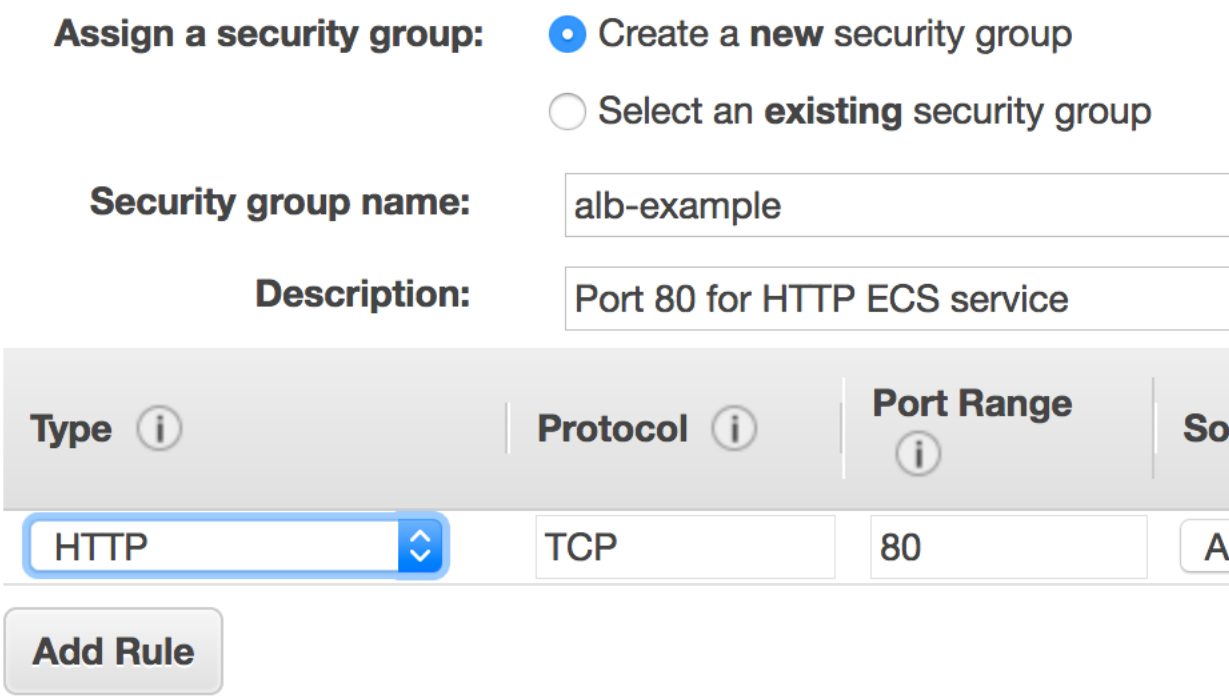
Você deve atribuir um grupo de segurança ao load balancer que permita tráfego de entrada às portas especificadas para os ouvintes. O Amazon ECS não atualiza automaticamente os grupos de segurança associados aos load balancers do Elastic Load Balancing ou às instâncias de contêiner do Amazon ECS.

Para atribuir um security group ao seu load balancer

1. Na página Atribuir security groups, selecione Criar novo security group.
2. Digite um nome e uma descrição para seu security group ou mantenha o nome e a descrição padrão. Esse novo security group contém uma regra que permite o tráfego para a porta que você configurou para que o listener usasse.

   Note

   Ainda neste tópico, você criará uma regra de security group para suas instâncias de contêiner que permita o tráfego em todas as portas provenientes do security group criado aqui, para que o Balanceador de carga de aplicações possa direcionar o tráfego para portas de host atribuídas dinamicamente em suas instâncias de contêiner.



3. Escolha Próximo: Configurar roteamento para ir para a próxima página do assistente.

Configurar roteamento

Nesta seção, você cria um grupo de destino para o load balancer e os critérios de verificação da integridade para destinos registrados dentro desse grupo.

Para criar um grupo de destino e configurar verificações de integridade

1. No Grupo de destino, mantenha o padrão, o Novo grupo de destino.
2. Em Nome, digite um nome para o novo grupo de destino.
3. Defina o Protocolo e a Porta conforme o necessário.
4. Para Target type, escolha se você quer registrar seus destinos com um ID de instância ou um endereço IP.

   Important

   Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), você precisará escolher `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.
5. Em Verificações de integridade, mantenha as configurações padrão para elas.
6. Selecione Próximo: registrar destinos.

Registrar destinos

O load balancer distribui tráfego entre os destinos registrados nos grupos de destino. Quando você associa um grupo de destino a um serviço do Amazon ECS, o Amazon ECS registra e cancela o registro

automaticamente dos contêineres com o grupo de destino. Como o Amazon ECS processa o registro de destino, você não adiciona destinos ao grupo de destino no momento.

Para ignorar o registro de destino

1. Na seção Instâncias registradas, garanta que nenhuma instância esteja selecionada para registro.
2. Escolha Próximo: Revisar para ir para a próxima página do assistente.

Revisar e criar

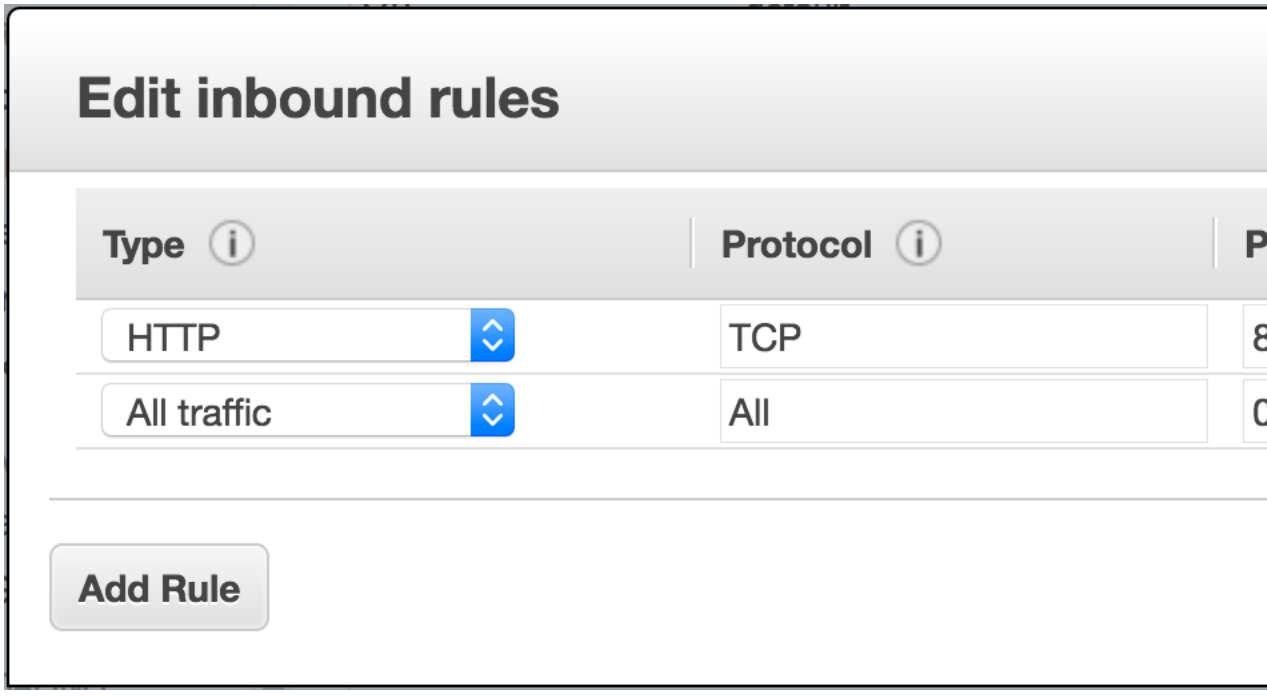
Revise a configuração do grupo de destino e do load balancer e escolha Create para criar o load balancer.

Criar uma regra do security group para as instâncias de contêiner

Depois que o Balanceador de carga de aplicações tiver sido criado, você deverá adicionar uma regra de entrada ao grupo de segurança da instância de contêiner que permite que o tráfego do load balancer alcance os contêineres.

Para permitir o tráfego de entrada de seu load balancer para suas instâncias de contêiner

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. No painel de navegação esquerdo, escolha Security Groups.
3. Escolha o security group que suas instâncias de contêiner usam. Se você criou suas instâncias de contêiner usando o assistente de primeira execução do Amazon ECS, esse grupo de segurança pode ter a descrição, ECS Allowed Ports (Portas permitidas pelo ECS).
4. Escolha a guia Inbound e depois Edit.
5. Para Tipo, escolha Todo o tráfego.
6. Em Source (Origem), escolha Custom (Personalizar) e digite o nome do grupo de segurança do Balanceador de carga de aplicações que você criou em Configurar grupos de segurança (p. 122). Essa regra permite que todo o tráfego do seu Balanceador de carga de aplicações atinja os contêineres em suas tarefas que estão registradas no seu load balancer.



7. Escolha Salvar para terminar.

Criar um serviço do Amazon ECS

Depois que o load balancer e o grupo de destino forem criados, você poderá especificar o grupo de destino em uma definição de serviço ao criar um serviço. Quando cada tarefa para o serviço é iniciada, a combinação de contêiner e porta especificada na definição de serviço é registrada com o grupo de destino e o tráfego é roteado do load balancer para esse contêiner. Para obter mais informações, consulte Criar um serviço (p. 148).

Criar um Load balancer de rede

Esta seção orienta você em meio ao processo de criação de um Load balancer de rede no Console de gerenciamento da AWS.

Defina o load balancer

Primeiro, forneça algumas informações sobre a configuração básica do load balancer, como nome, rede e escuta.

Escuta é um processo que verifica se há solicitações de conexão. Ele é configurado com um protocolo e uma porta para conexões front-end (cliente para load balancer), além de um protocolo e uma porta para conexões back-end (load balancer para instância back-end). Nesse exemplo, você configura um load balancer voltado para a internet na rede selecionada com um listener que recebe tráfego TCP na porta 80.

Para definir o load balancer

1. Abra o console do Amazon EC2 em https://console.aws.amazon.com/ec2/.
2. Na barra de navegação, selecione uma região para seu load balancer. Selecione a mesma região selecionada para suas instâncias de contêiner do Amazon ECS.
3. No painel de navegação, em LOAD BALANCING, escolha Load balancers.
4. Selecione Criar load balancer.
5. Na página Select load balancer type (Selecionar tipo de load balancer), escolha Create (Criar) em Load balancer de rede.
6. Preencha a página Configurar Load Balancer conforme a seguir:
   a. Em Nome, insira um nome para o load balancer.
   b. Em Esquema, escolha voltado para a internet ou interno. Um load balancer voltado para a Internet roteia solicitações de clientes da Internet até os destinos. Um load balancer interno roteia solicitações a destinos usando endereços IP privados.
   c. Em Listeners, o padrão é um listener que aceite tráfego TCP na porta 80. É possível manter as configurações do listener padrão, modificar o protocolo ou a porta do listener, ou ainda, selecionar Adicionar listener para inserir outro listener.

      Note

      Se você planeja rotear o tráfego para mais de um grupo de destino, consulte ListenerRules para obter detalhes sobre como adicionar regras baseadas em host ou caminho.
   d. Em Availability Zones (Zonas de disponibilidade), selecione a VPC usada nas instâncias do Amazon EC2. Para cada Zona de disponibilidade usada para iniciar suas instâncias do Amazon EC2, selecione uma Zona de disponibilidade e a sub-rede pública para ela. Para associar um endereço Elastic IP com a sub-rede, selecione-a no Elastic IP.
   e. Selecione Próximo: Configurar o roteamento.

Configurar roteamento

Você registra os destinos, como instâncias do Amazon EC2, com um grupo de destino. O grupo de destino que você configurar nesta etapa será usado como grupo de destino na regra do listener, que encaminha solicitações para o grupo de destino. Para obter mais informações, consulte Grupos de destino para seus load balancers de rede.

Para configurar seu grupo de destino

1. No Grupo de destino, mantenha o padrão, o Novo grupo de destino.
2. No Nome, digite um nome para o grupo de destino.
3. Defina o Protocolo e a Porta conforme o necessário.
4. Para Target type, escolha se você quer registrar seus destinos com um ID de instância ou um endereço IP.

   Important

   Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), você precisará escolher `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.
   Você não pode registrar instâncias por ID de instância se eles tiverem os seguintes tipos de instância: C1, CC1, CC2, CG1, CG2, CR1, G1, G2, HI1, HS1, M1, M2, M3 e T1. Você pode registrar instâncias desses tipos por endereço IP.
5. Em Verificações de integridade, mantenha as configurações padrão para elas.
6. Selecione Próximo: registrar destinos.

Registrar destinos com o grupo de destino

O load balancer distribui tráfego entre os destinos registrados nos grupos de destino. Quando você associa um grupo de destino a um serviço do Amazon ECS, o Amazon ECS registra e cancela o registro automaticamente dos contêineres com o grupo de destino. Como o Amazon ECS processa o registro de destino, você não adiciona destinos ao grupo de destino no momento.

Para ignorar o registro de destino

1. Na seção Instâncias registradas, garanta que nenhuma instância esteja selecionada para registro.
2. Escolha Próximo: Revisar para ir para a próxima página do assistente.

Revisar e criar

Revise a configuração do grupo de destino e do load balancer e escolha Create para criar o load balancer.

Criar um serviço do Amazon ECS

Depois que o load balancer e o grupo de destino forem criados, você poderá especificar o grupo de destino em uma definição de serviço ao criar um serviço. Quando cada tarefa para o serviço é iniciada, a combinação de contêiner e porta especificada na definição de serviço é registrada com o grupo de destino e o tráfego é roteado do load balancer para esse contêiner. Para obter mais informações, consulte Criar um serviço (p. 148).

Registro de vários grupos de destino com um serviço

Seu serviço do Amazon ECS pode atender ao tráfego de vários load balancers e expor várias portas com balanceamento de carga quando você especificar vários grupos de destino em uma definição de serviço.

Atualmente, se você desejar criar um serviço especificando vários grupos de destino, você deve criar o serviço usando a API do Amazon ECS o SDK, a AWS CLI ou um modelo do AWS CloudFormation. Depois que o serviço é criado, você pode visualizar o serviço e os grupos de destino registrados nele com o Console de gerenciamento da AWS.

Vários grupos de destino podem ser especificados em uma definição de serviço usando o seguinte formato.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"container_name",
      "containerPort":container_port
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"container_name",
      "containerPort":container_port
   }
]
```

Considerações sobre vários grupos de destino

Ao especificar vários grupos de destino em uma definição de serviço, considere o seguinte:

- Vários grupos de destino são compatíveis somente quando você usa os tipos de load balancer Balanceador de carga de aplicações ou Load balancer de rede.
- Vários grupos de destino são compatíveis somente quando o serviço usa o tipo de controlador de implantação de atualização contínua (`ECS`). Se você estiver usando o CodeDeploy ou um controlador de implantação externo, vários grupos de destino não serão compatíveis.
- Vários grupos de destino são compatíveis com serviços que contêm tarefas que usam os tipos de execução Fargate e EC2.
- Ao criar um serviço que especifica vários grupos de destino, a função vinculada ao serviço Amazon ECS deverá ser criada. A função é criada omitindo o parâmetro `role` nas solicitações de API ou a propriedade `Role` no AWS CloudFormation. Para obter mais informações, consulte Usar funções vinculadas ao serviço do Amazon ECS (p. 216).

Definições de serviço de exemplo

Veja a seguir alguns exemplos de casos de uso para especificar vários grupos de destino em uma definição de serviço.

Exemplo: ter load balancers separados para tráfego interno e externo

No seguinte caso de uso, um serviço usa dois load balancers, um para tráfego interno e um segundo para tráfego voltado para a Internet, para o mesmo contêiner e porta.

```
"loadBalancers":[
   //Internal ELB
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
```

```
      "containerName":"nginx",
      "containerPort":8080
   },
   //Internet-facing ELB
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"nginx",
      "containerPort":8080
   }
]
```

Exemplo: exposição de várias portas do mesmo contêiner

No seguinte caso de uso, um serviço usa um load balancer, mas expõe várias portas do mesmo contêiner. Por exemplo, um contêiner Jenkins pode expor a porta 8080 para a interface da Web do Jenkins e a porta 50000 para a API.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"jenkins",
      "containerPort":8080
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"jenkins",
      "containerPort":50000
   }
]
```

Exemplo: exposição de portas de vários contêineres

No seguinte caso de uso, um serviço usa um load balancer e dois grupos de destino para expor portas de contêineres separados.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"webserver",
      "containerPort":80
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"database",
      "containerPort":3306
   }
]
```

Serviço Auto Scaling

O serviço do Amazon ECS pode ser configurado para usar o Serviço Auto Scaling a fim de ajustar a contagem crescente ou decrescente automaticamente. O Serviço Auto Scaling. aproveita o serviço Aplicativo Auto Scaling para oferecer essa funcionalidade. O Serviço Auto Scaling está disponível em todas as regiões que oferecem suporte ao Amazon ECS. Para obter mais informações, consulte o Guia do usuário do Aplicativo Auto Scaling.

O Serviço Auto Scaling do Amazon ECS oferece suporte aos seguintes tipos de políticas de escalabilidade:

- Políticas de escalabilidade de rastreamento de destino (p. 130)—aumenta ou diminui o número de tarefas que o serviço executa com base em um valor de destino para uma métrica específica do CloudWatch. Isso é semelhante à forma como o termostato mantém a temperatura da casa. Você seleciona a temperatura, e o termostato faz o resto.
- Políticas de escalabilidade em etapas (p. 135)—aumenta ou diminui o número de tarefas que o serviço executa em resposta a alarmes do CloudWatch. A escalabilidade em etapas é baseada em um conjunto de ajustes de escalabilidade, conhecidos como ajustes em etapas, que variam com base no tamanho da ruptura do alarme.

Serviço Auto Scaling Permissões obrigatórias do IAM

Serviço Auto Scaling é possibilitado por uma combinação das APIs de Amazon ECS, CloudWatch e Application Auto Scaling. Os serviços são criados e atualizados com os alarmes do Amazon ECS, os alarmes são criados com o CloudWatch, e políticas de escalabilidade são criadas com o Application Auto Scaling. Os usuários do IAM devem ter as permissões apropriadas para esses serviços para que possam interagir com políticas de escalabilidade no Console de gerenciamento da AWS,na AWS CLI ou nos SDKs. Além das permissões do IAM padrão para criar e atualizar serviços, Serviço Auto Scaling exige as seguintes permissões:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:*",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Os exemplos de política Exemplo para criar serviço (p. 207) e Exemplo para atualizar serviço (p. 208) do IAM mostram as permissões obrigatórias para que usuários do IAM usem Serviço Auto Scaling no Console de gerenciamento da AWS.

O serviço do Application Auto Scaling precisa de permissão para descrever os serviços do Amazon ECS e os alarmes do CloudWatch, bem como permissões para modificar a contagem desejada do serviço em seu nome. Você deve criar uma função do IAM (`ecsAutoscaleRole`) para os serviços do ECS oferecerem essas permissões e acabarem associando essa função ao serviço para que possa usar Application Auto Scaling. Caso um usuário do IAM tenha as permissões obrigatórias para usar Serviço Auto Scaling no console do Amazon ECS, crie funções do IAM e anexe políticas de função do IAM a elas, e esse usuário poderá criar essa função automaticamente como parte dos fluxos de trabalho criar serviço (p.) ou

atualizar serviço (p. 160) do console do Amazon ECS e usar a função em qualquer outro serviço depois (no console ou com CLI ou SDKs). Você também pode criar a função seguindo os procedimentos em Função do IAM Serviço Auto Scaling do Amazon ECS (p. 237).

Como alternativa, o Application Auto Scaling também pode utilizar uma função vinculada ao serviço (`AWSServiceRoleForApplicationAutoScaling_ECSService`) que realiza ações de Auto Scaling em seu nome. Se você tiver as permissões necessárias, essa função será criada automaticamente para você, com as políticas do IAM necessárias associadas, quando começar a usar o serviço Application Auto Scaling. Para obter mais informações, consulte Funções vinculadas ao serviço para Auto Scaling de aplicativo no Guia do usuário do Aplicativo Auto Scaling. Você ou um administrador deve ter as permissões `iam:CreateServiceLinkedRole` ou `AdministratorAccess` para criar essa função vinculada ao serviço.

Políticas de escalabilidade de rastreamento de destino

Com políticas de escalabilidade de rastreamento de destino, você seleciona uma métrica do CloudWatch e define um valor de destino. O Amazon ECS cria e gerencia os alarmes do CloudWatch que acionam a política de escalabilidade e calculam o ajuste de escalabilidade com base na métrica e no valor de destino. A política de escalabilidade adiciona ou remove tarefas de serviço conforme necessário para manter a métrica no valor de destino especificado ou próxima a ele. Além de manter a métrica próxima ao valor de destino, uma política de escalabilidade de rastreamento de destino também se ajusta às flutuações na métrica, devido a um padrão de carga de flutuação, e minimiza as flutuações rápidas no número de tarefas que estão sendo executadas no serviço.

Você pode criar várias políticas de escalabilidade de rastreamento de destino para um serviço do Amazon ECS, desde que cada uma delas use uma métrica diferente. O serviço faz o dimensionamento com base na política que fornece a maior capacidade da tarefa. Com isso, é possível cobrir vários cenários e garantir que sempre haja capacidade suficiente para processar suas cargas de trabalho de aplicativos.

Para garantir a disponibilidade do aplicativo, o serviço se expande proporcionalmente à métrica o mais rápido possível, mas é reduzido gradualmente.

Não edite ou exclua os alarmes CloudWatch que o Amazon ECS gerencia para uma política de escalabilidade de rastreamento de destino. O Amazon ECS exclui os alarmes automaticamente quando você exclui a política de escalabilidade de rastreamento de destino.

Considerações

Ao criar uma política de escalabilidade de rastreamento de destino, tenha as seguintes considerações em mente:

- Uma política de escalabilidade de rastreamento de destino pressupõe que ela deve aumentar a escalabilidade quando a métrica especificada estiver acima do valor de destino. Você não pode usar uma política de escalabilidade de rastreamento de destino para expandir quando a métrica especificada estiver abaixo do valor de destino.
- Uma política de escalabilidade de rastreamento de destino não escala quando a métrica especificada tem dados insuficientes. Ela não aumenta a escalabilidade porque não interpreta dados insuficientes como baixa utilização.
- Você pode ver lacunas entre o valor de destino e os pontos de dados de métrica reais. Isso ocorre porque o Aplicativo Auto Scaling sempre funciona de maneira segura por arredondamento para cima ou para baixo, quando ele determina a capacidade a ser adicionada ou removida. Isso evita que ele adicione capacidade insuficiente ou remova muita capacidade. No entanto, para um destino dimensionável com capacidade pequena, os pontos de dados de métricas reais podem parecer distantes do valor de destino. Para um destino dimensionável com maior capacidade, a adição ou remoção de capacidade causa uma lacuna menor entre o valor de destino e os pontos de dados de métricas reais.
- Recomendamos que você defina a escalabilidade com base nas métricas com intervalos de 1 minuto, pois isso garante resposta mais rápida às mudanças de utilização. Aumentar a escalabilidade das

métricas com intervalos de cinco minutos pode resultar em tempo de resposta mais lento e aumentar a escalabilidade dos dados obsoletos.

- Para garantir a disponibilidade do aplicativo, o Aplicativo Auto Scaling se expande proporcionalmente à métrica o mais rápido possível, mas se retrai gradualmente.
- Não edite ou exclua os alarmes CloudWatch que o Aplicativo Auto Scaling gerencia para uma política de escalabilidade de rastreamento de destino. O Aplicativo Auto Scaling exclui os alarmes automaticamente quando você exclui a política de escalabilidade.

Tutorial: Auto Scaling de serviços com rastreamento de destino

Os procedimentos a seguir ajudam você a criar um cluster do Amazon ECS e um serviço que usa o Aplicativo Auto Scaling para expandir (e reduzir) usando o rastreamento de destino.

Neste tutorial, você cria um cluster e um serviço (que fica em execução por trás de um load balancer do Elastic Load Balancing) usando o assistente de primeira execução do Amazon ECS. Em seguida, você configura Serviço Auto Scaling no serviço com alarmes do CloudWatch que usam a métrica `ECSServiceAverageCPUUtilization` para expandir ou reduzir o serviço, dependendo da carga atual do aplicativo.

Quando a utilização da CPU do serviço aumenta acima de 75% (significando que mais de 75% da CPU reservada para o serviço está sendo usada), o alarme de expansão aciona o Serviço Auto Scaling para adicionar outra tarefa no serviço para ajudar com o aumento de carga. Do mesmo modo, quando a utilização da CPU de seu serviço cai abaixo de 75%, o alarme de redução aciona uma diminuição na contagem desejada do serviço para liberar esses recursos de cluster para outras tarefas e serviços.

Pré-requisitos

Este tutorial supõe que você possui uma conta da AWS, um administrador do IAM com permissões para executar todas as ações descritas nele e um par de chaves do Amazon EC2 na região atual. Se você não tem esses recursos, ou não tem certeza se os tem, você pode criá-los seguindo as etapas em Configuração com o Amazon ECS (p. 7).

Etapa 1: Criar um cluster e um serviço

Depois de habilitar as métricas do CloudWatch para os clusters e serviços, você pode criar um cluster e um serviço usando o assistente de primeira execução do Amazon ECS. O assistente de primeira execução cria as funções e políticas do IAM necessárias para este tutorial, um grupo do Auto Scaling para as instâncias de contêiner e um serviço que é executado por trás de um load balancer. O assistente também torna o processo de limpeza muito mais fácil, pois você pode excluir toda a pilha do AWS CloudFormation em uma etapa.

Para este tutorial, você cria um cluster chamado `service-autoscaling` e um serviço chamado `sample-webapp`.

Para criar seu cluster e serviço

1. Abra o assistente de primeira execução do console do Amazon ECS em https://console.aws.amazon.com/ecs/home#/firstRun.
2. Na barra de navegação, escolha a região US East (N. Virginia) (Leste dos EUA (Norte da Virgínia)).
3. Em Step 1: Container and Task (Etapa 1: contêiner e tarefa), para Container definition (Definição do contêiner), selecione sample-app.
4. Na página Task definition (Definição de tarefa), deixe todas as opções padrão e escolha Next (Próxima).
5. Em Step 2: Service (Etapa 2: serviço), para Load balancer type (Tipo de load balancer), escolha Application Load Balancer e, em seguida, escolha Next (Próxima).

Important

Os Balanceador de carga de aplicaçõess não geram custo enquanto existem em seus recursos da AWS. Para obter mais informações, consulte Definição de preço Elastic Load Balancing.

6. Em Step 3: Cluster (Etapa 3: cluster), para Cluster name (Nome do cluster), digite `service-autoscaling` e escolha Next (Próxima).
7. Verifique suas escolhas e selecione Create (Criar).

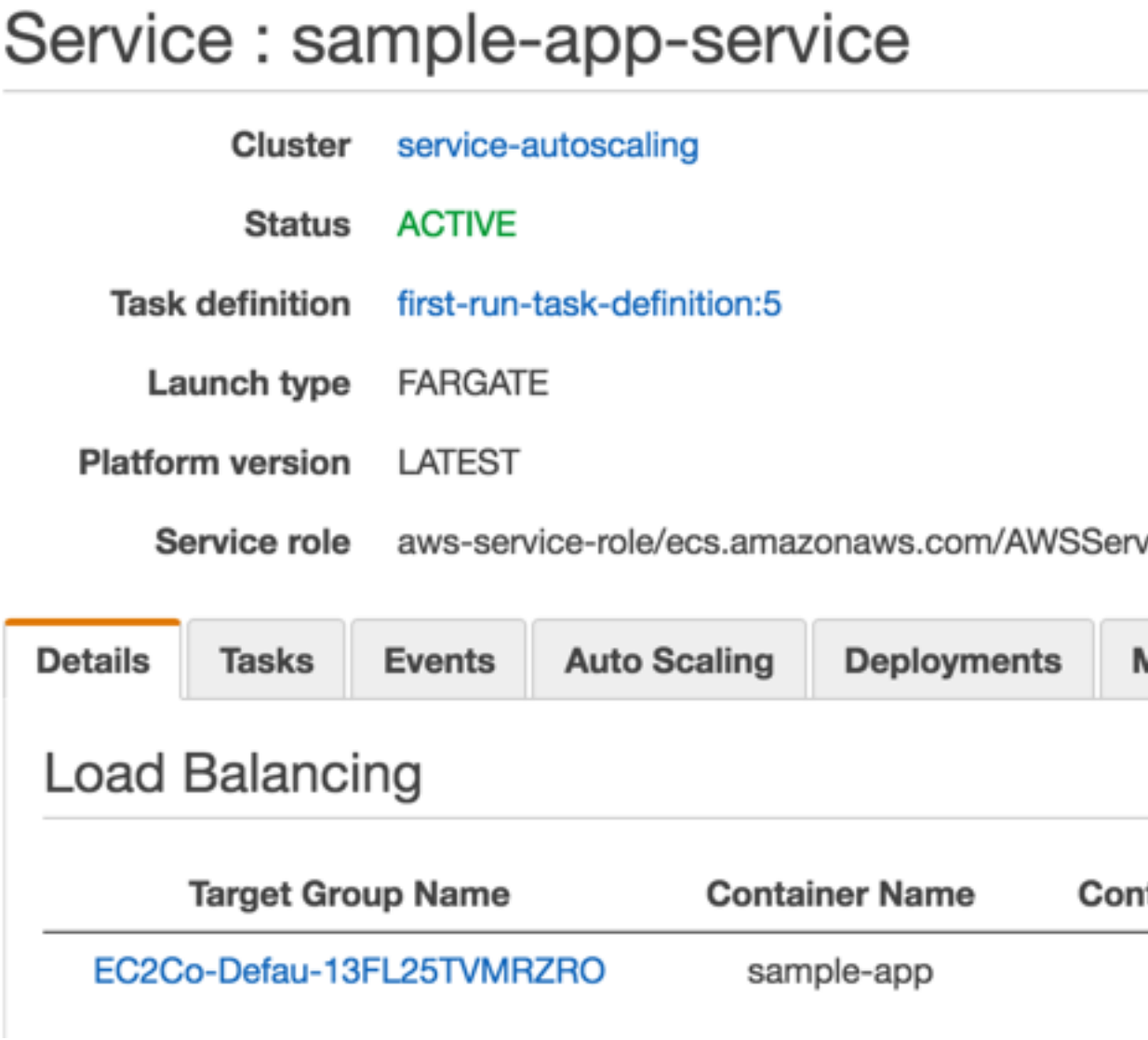
   Você será direcionado para uma página Launch Status (Status da inicialização) que mostra o status de sua inicialização e descreve cada etapa do processo (esse processo pode levar alguns minutos para concluir, enquanto os recursos do cluster são criados e preenchidos).
8. Quando seu cluster e serviço forem criados, selecione View service (Visualizar serviço).

Etapa 2: Configurar o Serviço Auto Scaling

Agora que você iniciou um cluster e criou um serviço no cluster que está em execução por trás de um load balancer, você pode configurar o Serviço Auto Scaling criando políticas de escalabilidade para expandir ou reduzir seu serviço em resposta aos alarmes do CloudWatch.

Para configurar os parâmetros básicos do Serviço Auto Scaling

1. Na página Service: sample-app-service (Serviço: sample-app-service), a sua configuração de serviço deve ser semelhante à imagem abaixo, embora a revisão da definição da tarefa e o nome do load balancer provavelmente sejam diferentes. Escolha Atualizar para atualizar o seu novo serviço.



2. Na página Update service (Atualizar serviço), escolha Next step (Próxima etapa) até chegar a Step 3: Set Auto Scaling (optional) (Etapa 3: definir Auto Scaling (opcional)).
3. Em Serviço Auto Scaling, escolha Configure Service Auto Scaling to adjust your service's desired count (Configurar Auto Scaling do serviço para ajustar a contagem desejada do serviço).
4. Em Minimum number of tasks (Número mínimo de tarefas), insira `1` para o limite inferior do número de tarefas para uso do Serviço Auto Scaling. A contagem desejada do seu serviço não será automaticamente ajustada para um valor abaixo desse.
5. Em Desired number of tasks (Número de tarefas desejadas), esse campo é preenchido antecipadamente com o valor já inserido. Esse valor deve ser um entre os números mínimo e máximo de tarefas especificado nessa página. Deixe esse valor em `1`.
6. Em Maximum number of tasks (Número máximo de tarefas), insira `2` para o limite superior do número de tarefas para uso do Serviço Auto Scaling. A contagem desejada do seu serviço não será automaticamente ajustada um valor acima desse.
7. Em IAM role for Service Auto Scaling (Função do IAM para Auto Scaling de serviço), escolha uma função do IAM a fim de autorizar o serviço Application Auto Scaling a ajustar a contagem desejada do seu serviço em seu nome. Se você não tiver criado essa função anteriormente, escolha Create new

role e a função será criada para você. Para referência futura, a função criada para você é chamada `ecsAutoscaleRole`. Para obter mais informações, consulte Função do IAM Serviço Auto Scaling do Amazon ECS (p. 237).

Para configurar políticas de escalabilidade para o serviço

Essas etapas o ajudarão a criar políticas de escalabilidade e alarmes do CloudWatch que podem ser usados para acionar ações de escalabilidade para o seu serviço. Você pode criar um alarme de expansão para aumentar a contagem desejada de serviços, e um alarme de redução para diminuir a contagem desejada de serviços.

1. Escolha Add scaling policy (Adicionar política de escalabilidade) para configurar a política de escalabilidade.
2. Na página Add policy (Adicionar política), atualize os seguintes campos:
   a. Em Scaling policy type (Tipo de política de escalabilidade), escolha Target tracking (Rastreamento de destino).
   b. Em Nome da política, insira `TargetTrackingPolicy`.
   c. Em ECS service metric (Métrica do serviço do ECS), escolha ECSServiceAverageCPUUtilization.
   d. Em Target value (Valor de destino), insira `75`.
   e. Em Scale-out cooldown period (Período de desaquecimento após expansão), insira `60`. Essa é a quantidade de tempo, em segundos, após a conclusão de uma ação de expansão e antes que outras atividades de expansão possam iniciar. Durante esse período, os recursos que foram iniciados não contribuem para a métrica de grupo do Auto Scaling.
   f. Em Scale-in cooldown period (Período de desaquecimento após redução), insira `60`. Essa é a quantidade de tempo, em segundos, após a conclusão de uma ação de redução e antes que outras atividades de redução possam iniciar. Durante esse período, os recursos que foram iniciados não contribuem para a métrica de grupo do Auto Scaling.
   g. Escolha Salvar.
3. Escolha Próxima etapa.
4. Revise todas as suas opções e, em seguida, escolha Update Service (Atualizar serviço).
5. Quando o status do serviço for atualização concluída, escolha Exibir serviço.

Etapa 3: Acionar uma atividade de escalabilidade

Depois de configurar seu serviço com Serviço Auto Scaling, você pode acionar uma ação de escalabilidade colocando a utilização de CPU do serviço no estado `ALARM`. Como o exemplo neste tutorial é de um aplicativo web em execução por trás de um load balancer, você poderá enviar milhares de solicitações HTTP para seu serviço (usando o utilitário ApacheBench) para determinar o pico de utilização de CPU do serviço acima da quantidade limite. Esse pico deve acionar o alarme, que acionará, por sua vez, uma atividade de escalabilidade para adicionar uma tarefa ao seu serviço.

Depois que o utilitário ApacheBench encerrar as solicitações, a utilização de CPU do serviço deve ficar abaixo do limite de 25%, acionando uma escala na atividade que retorna a contagem desejada do serviço para 1.

Para acionar uma ação de escalabilidade para o seu serviço

1. Na página de visualização principal do serviço no console, escolha o nome do load balancer para exibir seus detalhes no console do Amazon EC2. Você precisa do nome DNS do load balancer, que deve ser parecido com este `EC2Contai-EcsElast-SMAKV74U23PH-96652279.us-east-1.elb.amazonaws.com`.
2. Use o utilitário ApacheBench (ab) para fazer milhares de solicitações HTTP para o load balancer em um curto período.

Note

Esse comando é instalado por padrão no macOS e também está disponível para várias distribuições do Linux. Por exemplo, você pode instalar o ab no Amazon Linux com o seguinte comando:

```
$ sudo yum install -y httpd24-tools
```

Execute o comando a seguir, substituindo o nome DNS do load balancer.

```
$ ab -n 100000 -c 1000 http://EC2Contai-EcsElast-SMAKV74U23PH-96652279.us-east-1.elb.amazonaws.com/
```

3. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
4. No painel de navegação à esquerda, escolha Alarms (Alarmes).
5. Aguarde as suas solicitações HTTP ab para acionar o alarme de expansão no console do CloudWatch. Você deve ver o seu serviço do Amazon ECS se expandir e adicionar uma tarefa à sua contagem desejada do serviço.
6. Assim que suas solicitações HTTP ab forem concluídas (entre 1 e 2 minutos), seu alarme de expansão deverá ser acionado e a política de redução diminuirá a contagem desejada de seu serviço de volta para 1.

Etapa 4: Como limpar

Ao concluir este tutorial, você poderá optar por manter seu cluster, grupo de Auto Scaling, load balancer e alarmes do CloudWatch. Contudo, se não estiver usando ativamente esses recursos, você deve considerar a remoção deles para que sua conta não incorra em cobranças desnecessárias.

Para excluir seu cluster e alarmes do CloudWatch

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação à esquerda, escolha Clusters.
3. Na página Clusters, escolha o cluster service-autoscaling.
4. Escolha Delete Cluster (Excluir cluster), Delete (Excluir). Pode levar alguns minutos para que a pilha do AWS CloudFormation do cluster conclua a limpeza.
5. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
6. Escolha Alarms (Alarmes) e selecione os alarmes que começam com `TargetTracking-service`.
7. Escolha Delete (Excluir), Yes, Delete (Sim, excluir).

Políticas de escalabilidade em etapas

Com as políticas de escalabilidade em etapas, você especifica os alarmes do CloudWatch para acionamento do processo de escalabilidade. Por exemplo, se você deseja aumentar a escala quando a utilização de CPU atinge um determinado nível, crie um alarme usando a métrica `CPUUtilization` fornecida pelo Amazon ECS.

Conceitos de escalabilidade em etapas

- O programador de serviço do ECS respeita a contagem desejada sempre, mas desde que você tenha políticas de escalabilidade e alarmes ativos em um serviço, Serviço Auto Scaling pode alterar uma contagem desejada que foi definida manualmente por você.

- Caso a contagem desejada de um serviço esteja definida abaixo do valor mínimo de capacidade e um alarme dispare uma ação de expansão, o Application Auto Scaling expande a contagem desejada até o valor mínimo de capacidade e continua expandindo conforme necessário, com base na política de escalabilidade associada ao alarme. Porém, uma atividade de redução não ajusta a contagem desejada, pois ela já está abaixo do valor mínimo de capacidade.
- Caso a contagem desejada de um serviço esteja definida acima do valor de capacidade máximo e um alarme dispare uma atividade de aumento da escala, o Application Auto Scaling dimensiona a contagem desejada até o valor de capacidade máximo e continua dimensionando conforme necessário, com base na política de escalabilidade associada ao alarme. Porém, uma atividade de expansão não ajusta a contagem desejada, pois ela já está acima do valor máximo de capacidade.
- Durante atividades de escalabilidade, a contagem de tarefas em execução real em um serviço é o valor que Serviço Auto Scaling usa como o ponto de partida, ao contrário da contagem desejada, que é que capacidade de processamento que deve ser. Isso evita a escalabilidade excessiva (sem controle) que não pode ser atendida, por exemplo, caso não haja recursos de instância de contêiner suficientes para colocar as tarefas adicionais. Se a capacidade da instância de contêiner estiver disponível depois, a ação de escalabilidade pendente poderá continuar, e as ações de escalabilidade adicionais poderão continuar depois do período de desaquecimento.

Experiência de console do Amazon ECS

Os fluxos de trabalho de criação e atualização do serviço de console do Amazon ECS oferecem suporte a políticas de escalabilidade em etapas. O console do Amazon ECS lida com o `ecsAutoscaleRole` e a criação de política, desde que o usuário do IAM que está usando o console tenha as permissões descritas em Serviço Auto Scaling Permissões obrigatórias do IAM (p. 129) e elas possam criar funções do IAM e associar políticas a elas.

Quando você configura um serviço para usar Serviço Auto Scaling no console, o serviço é registrado automaticamente como um destino dimensionável com o Application Auto Scaling, de maneira que você possa configurar políticas de escalabilidade que dimensionem o serviço. Você também pode criar e atualizar as políticas de escalabilidade e os alarmes do CloudWatch que os dispara no console do Amazon ECS.

Para criar um novo serviço do ECS que use Serviço Auto Scaling, consulte Criar um serviço (p. 148).

Para atualizar um serviço existente a fim de usar Serviço Auto Scaling, consulte Atualizar um serviço (p. 160).

AWS CLI e experiência do SDK

Você pode configurar Serviço Auto Scaling usando a AWS CLI ou os SDKs da AWS, mas deve observar as considerações a seguir.

- Serviço Auto Scaling é possibilitado por uma combinação das APIs de Amazon ECS, CloudWatch e Application Auto Scaling. Os serviços são criados e atualizados com o Amazon ECS, os alarmes são criados com CloudWatch e as políticas de escalabilidade são criadas com o Application Auto Scaling. Para obter mais informações sobre essas operações de API específicas, consulte Amazon Elastic Container Service API Reference, Amazon CloudWatch API Reference e Referência da API do Aplicativo Auto Scaling. Para obter mais informações sobre os comandos da AWS CLI desses serviços, consulte as seções ecs, cloudwatch e application-autoscaling do AWS CLI Command Reference.
- Para o serviço usar Serviço Auto Scaling, você deve registrá-lo como um destino escalável com a operação da API Application Auto ScalingRegisterScalableTarget.
- Depois que o serviço do ECS for registrado como um destino escalável, você poderá criar políticas de escalabilidade com a operação da API PutScalingPolicy Application Auto Scaling para especificar o que deverá acontecer quando os alarmes do CloudWatch forem disparados.

- Depois que criar as políticas de escalabilidade para o serviço, você poderá criar os alarmes do CloudWatch que disparam os eventos de escalabilidade para o serviço com a operação de API PutMetricAlarm CloudWatch.

Descoberta de serviço

Seu serviço do Amazon ECS pode, opcionalmente, ser configurado para usar o Amazon ECS Descoberta de serviço. O Descoberta de serviço usa ações da API do AWS Cloud Map para gerenciar namespaces HTTP e DNS para seus serviços do Amazon ECS. Para obter mais informações, consulte O que é AWS Cloud Map? em Guia do desenvolvedor do AWS Cloud Map.

Tópicos

- Conceitos do Descoberta de serviço (p. 137)
- Considerações sobre o Descoberta de serviço (p. 138)
- Experiência de console do Amazon ECS (p. 139)
- Definição de preço do Descoberta de serviço (p. 139)
- Tutorial: como criar um serviço usando Descoberta de serviço (p. 139)

Conceitos do Descoberta de serviço

O Descoberta de serviço consiste nos seguintes componentes:

- Descoberta de serviçonamespace: um grupo lógico de serviços descoberta de serviço que compartilham o mesmo nome de domínio, como `example.com`.
- Serviço Descoberta de serviço: existe no namespace da descoberta de serviço e consiste no nome do serviço e na configuração do DNS para o namespace. Ele fornece os seguintes componentes principais:
  - Service registry (Registro do serviço): permite pesquisar um serviço por meio de DNS ou ações de API do AWS Cloud Map disponíveis que podem ser usados para se conectar ao serviço.
- Descoberta de serviço instância: existe no serviço descoberta de serviço e consiste nos atributos associados a cada serviço do Amazon ECS no diretório de serviços.
  - Atributos de instância: os metadados a seguir são adicionados como atributos personalizados para cada serviço do Amazon ECS configurado para usar o descoberta de serviço:
    - **`AWS_INSTANCE_IPV4`** – Para um registro `A`, o endereço IPv4 que o Route 53 retorna em resposta às consultas DNS e que o AWS Cloud Map retorna ao descobrir detalhes da instância, por exemplo, `192.0.2.44`.
    - **`AWS_INSTANCE_PORT`** – O valor da porta associado ao serviço descoberta de serviço.
    - **`AVAILABILITY_ZONE`** – A zona de disponibilidade na qual a tarefa foi executada. Para tarefas que usam o tipo de inicialização EC2, esta é a zona de disponibilidade na qual a instância de contêiner existe. Para tarefas que usam o tipo de inicialização Fargate, esta é a zona de disponibilidade na qual a interface de rede elástica existe.
    - **`REGION`** – A região em que a tarefa existe.
    - **`ECS_SERVICE_NAME`** – O nome do serviço do Amazon ECS ao qual a tarefa pertence.
    - **`ECS_CLUSTER_NAME`** – O nome do cluster do Amazon ECS ao qual a tarefa pertence.
    - **`EC2_INSTANCE_ID`** – O ID da instância de contêiner na qual a tarefa foi colocada. Este atributo personalizado não será adicionado se a tarefa estiver usando o tipo de inicialização Fargate.
    - **`ECS_TASK_DEFINITION_FAMILY`** – A família de definições de tarefas que a tarefa está usando.
    - **`ECS_TASK_SET_EXTERNAL_ID`** – se um conjunto de tarefas for criado para uma implantação externa e for associado a um registro de descoberta de serviço, o atributo `ECS_TASK_SET_EXTERNAL_ID` conterá o ID externo do conjunto de tarefas.

- Verificações de integridade do Amazon ECS: o Amazon ECS executa verificações de integridade periódicas em nível de contêiner. Se não passar na verificação de integridade, o endpoint é removido do roteamento de DNS e marcado como não íntegro.

Considerações sobre o Descoberta de serviço

As seguintes informações devem ser considerada ao usar a descoberta de serviço:

- O Descoberta de serviço é compatível com as tarefas que usam o tipo de inicialização Fargate se estiverem usando a versão v1.1.0 ou posterior da plataforma. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).
- O fluxo de trabalho "Criar Serviço" no console do Amazon ECS é compatível apenas com o registro de serviços em namespaces DNS privados. Quando um namespace DNS privado do AWS Cloud Map for criado, uma zona hospedada privada do Route 53 será criada automaticamente.
- O Amazon ECS não oferece suporte ao registro de serviços em namespaces DNS públicos.
- Os registros de DNS criados para um serviço descoberta de serviço serão sempre registrados com o endereço IP privado da tarefa, em vez do endereço IP público, mesmo quando forem usados namespaces públicos.
- O Descoberta de serviço exige que as tarefas especifiquem o modo de rede `awsvpc`, `bridge` ou `host` (`none` não é compatível).
- Se a definição de tarefa que a sua tarefa de serviço especifica usa o modo de rede `awsvpc`, você pode criar qualquer combinação de registros A ou SRV para cada tarefa de serviço. Se você usar registros SRV, uma porta é necessária.
- Se a definição de tarefa especificada por sua tarefa de serviço usa o modo de rede `bridge` ou `host`, o único tipo de registro DNS compatível é o registro SRV. Crie um registro SRV para cada tarefa de serviço. O registro SRV deve especificar o nome do contêiner e a combinação de portas do contêiner da definição de tarefa.
- Os registros DNS de um serviço de descoberta de serviço podem ser consultados em sua VPC. Eles usam o seguinte formato: `<descoberta de serviço service name>.<descoberta de serviço namespace>`. Para obter mais informações, consulte Etapa 3: Verificar Descoberta de serviço (p. 144).
- Ao fazer uma consulta de DNS no nome do serviço, os registros A retornam um conjunto de endereços IP correspondentes às suas tarefas. Os registros SRV retornam um conjunto de endereços IP e portas por tarefa.
- Se você tiver oito ou menos registros íntegros, o Route 53 responderá a todas as consultas DNS com todos os registros íntegros.
- Quando nenhum dos registros estiver íntegro, o Route 53 responderá às consultas DNS com até oito registros não íntegros.
- É possível configurar o descoberta de serviço para um serviço do ECS que está atrás de um load balancer, mas o tráfego do descoberta de serviço é sempre roteado para a tarefa, e não para o load balancer.
- Descoberta de serviço não é compatível com o uso de Classic Load Balancers.
- Recomenda-se o uso de verificações de integridade no nível do contêiner gerenciadas pelo Amazon ECS para seu serviço descoberta de serviço.
  - HealthCheckCustomConfig — O Amazon ECS gerencia verificações de integridade em seu nome. O Amazon ECS usa informações de contêiner e verificações de integridade, bem como o estado da tarefa, para atualizar a integridade com o AWS Cloud Map. Isso é especificado usando o parâmetro `--health-check-custom-config` ao criar seu serviço de descoberta de serviço. Para obter mais informações, consulte HealthCheckCustomConfig no Referência da API do AWS Cloud Map.
- Se você estiver usando o console do Amazon ECS, o fluxo de trabalho criará um serviço do descoberta de serviço por serviço do ECS. Ele mapeia todos os endereços IP da tarefa como registros A ou endereços IP e porta da tarefa como registros SRV.

- Descoberta de serviço só pode ser configurado durante a criação de um serviço. Atualização de serviços existentes para configurar descoberta de serviço pela primeira vez ou alterar a configuração atual não é suportada.
- Os recursos do AWS Cloud Map criados quando o descoberta de serviço é usado deve ser limpo manualmente. Para obter mais informações, consulte Etapa 4: Limpeza (p. 147) no tópico Tutorial: como criar um serviço usando Descoberta de serviço (p. 139).

Experiência de console do Amazon ECS

Os fluxos de trabalho de atualização e criação de serviços no console do Amazon ECS dão suporte a descoberta de serviço.

Para criar um novo serviço do Amazon ECS que use descoberta de serviço, consulte Criar um serviço (p. 148).

Definição de preço do Descoberta de serviço

Os clientes que usam a descoberta de serviço do Amazon ECS são cobrados pelos recursos do Route 53 e as operações da API de descoberta do AWS Cloud Map. Isso envolve custos de criação de zonas hospedadas do Route 53 e de consultas ao registro do serviço. Para obter mais informações, consulte Definição de preço do AWS Cloud Map no Guia do desenvolvedor do AWS Cloud Map.

O Amazon ECS executa verificações de integridade no nível do contêiner e as expõe às operações da API de verificação de integridade personalizada do AWS Cloud Map. Atualmente, isso é disponibilizado aos clientes sem nenhum custo extra. Se configurar verificações de integridade de rede adicionais para tarefas expostas publicamente, você será cobrado por essas verificações.

Tutorial: como criar um serviço usando Descoberta de serviço

O Descoberta de serviço foi integrado ao assistente Create Service (Criar serviço) no console do Amazon ECS. Para obter mais informações, consulte Criar um serviço (p. 148).

O tutorial a seguir mostra como criar um serviço do ECS contendo uma tarefa Fargate que usa a descoberta de serviço com a AWS CLI.

Note

As tarefas Fargate são compatíveis somente nas seguintes regiões:

Nome da região	Região
Leste dos EUA (Norte da Virgínia)	us-east-1
Leste dos EUA (Ohio)	us-east-2
Oeste dos EUA (Norte da Califórnia)	us-west-1
Oeste dos EUA (Oregon)	us-west-2
Ásia Pacífico (Mumbai)	ap-south-1
UE (Irlanda)	eu-west-1
UE (Londres)	eu-west-2
UE (Frankfurt)	eu-central-1

Nome da região	Região
Ásia-Pacífico (Tóquio)	ap-northeast-1
Canadá (Central)	ca-central-1
Ásia-Pacífico (Seul)	ap-northeast-2
Ásia-Pacífico (Cingapura)	ap-southeast-1
Ásia-Pacífico (Sydney)	ap-southeast-2

Tópicos

- Pré-requisitos (p. 140)
- Etapa 1: criar recursos da Descoberta de serviço (p. 140)
- Etapa 2: criar recursos do Amazon ECS (p. 141)
- Etapa 3: Verificar Descoberta de serviço (p. 144)
- Etapa 4: Limpeza (p. 147)

Pré-requisitos

Este tutorial pressupõe que os seguintes pré-requisitos foram concluídos:

- A versão mais recente da AWS CLI está instalada e configurada. Para obter mais informações, consulte Instalação da interface de linha de comando da AWS.
- As etapas em Configuração com o Amazon ECS (p. 7) foram concluídas.
- Seu usuário da AWS tem as permissões necessárias especificadas no exemplo de política Permissões do assistente de primeira execução do Amazon ECS (p. 202) do IAM.
- Você tem uma VPC e um grupo de segurança criados para uso. Para obter mais informações, consulte Tutorial: como criar uma VPC com sub-redes públicas e privadas para seus clusters.

Etapa 1: criar recursos da Descoberta de serviço

Use as etapas a seguir para criar o namespace de descoberta de serviço e o serviço de descoberta de serviço.

Para criar recursos da Descoberta de serviço

1. Crie um namespace descoberta de serviço privado denominado `tutorial` em uma das VPCs existentes:

```
aws servicediscovery create-private-dns-namespace --name tutorial --vpc vpc-abcd1234 --
region us-east-1
```

   Resultado:

```
{
    "OperationId": "h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-je6chs2e"
}
```

2. Usando o `OperationId` do resultado anterior, verifique se o namespace privado foi criado. Copie o ID do namespace como usado em comandos subsequentes.

```
aws servicediscovery get-operation --operation-id h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-
je6chs2e
```

Resultado:

```
{
    "Operation": {
        "Id": "h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-je6chs2e",
        "Type": "CREATE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1519777852.502,
        "UpdateDate": 1519777856.086,
        "Targets": {
            "NAMESPACE": "ns-uejictsjen2i4eeg"
        }
    }
}
```

3. Usando o ID `NAMESPACE` ID da saída anterior, crie um serviço de descoberta de serviço denominado `myapplication`. Copie o ID do serviço descoberta de serviço como usado em comandos subsequentes:

```
aws servicediscovery create-service --name myapplication --dns-config 'NamespaceId="ns-
uejictsjen2i4eeg",DnsRecords=[{Type="A",TTL="300"}]' --health-check-custom-config
 FailureThreshold=1 --region us-east-1
```

Resultado:

```
{
    "Service": {
        "Id": "srv-utcrh6wavdkggqtk",
        "Arn": "arn:aws:servicediscovery:region:aws_account_id:service/srv-
utcrh6wavdkggqtk",
        "Name": "myapplication",
        "DnsConfig": {
            "NamespaceId": "ns-uejictsjen2i4eeg",
            "DnsRecords": [
                {
                    "Type": "A",
                    "TTL": 300
                }
            ]
        },
        "HealthCheckCustomConfig": {
            "FailureThreshold": 1
        },
        "CreatorRequestId": "e49a8797-b735-481b-a657-b74d1d6734eb"
    }
}
```

Etapa 2: criar recursos do Amazon ECS

Use as etapas a seguir para criar o cluster, a definição de tarefa e os serviço do Amazon ECS.

Para criar recursos do Amazon ECS

1. Crie um cluster do Amazon ECS denominado `tutorial` a ser usado:

```
aws ecs create-cluster --cluster-name tutorial --region us-east-1
```

Resultado:

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/tutorial",
        "clusterName": "tutorial",
        "status": "ACTIVE",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": []
    }
}
```

2. Registre uma definição de tarefa compatível com Fargate. Ele exige o uso do modo de rede `awsvpc`. Veja a seguir o exemplo de definição de tarefa usado para este tutorial.

   Primeiro, crie um arquivo denominado `fargate-task.json` com o conteúdo da seguinte definição de tarefa:

```
{
    "family": "tutorial-task-def",
        "networkMode": "awsvpc",
        "containerDefinitions": [
            {
                "name": "sample-app",
                "image": "httpd:2.4",
                "portMappings": [
                    {
                        "containerPort": 80,
                        "hostPort": 80,
                        "protocol": "tcp"
                    }
                ],
                "essential": true,
                "entryPoint": [
                    "sh",
                    "-c"
                ],
                "command": [
                    "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
 ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html && httpd-
foreground\""
                ]
            }
        ],
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "cpu": "256",
        "memory": "512"
}
```

   Em seguida, registre a definição de tarefa usando o arquivo `fargate-task.json` criado:

```
aws ecs register-task-definition --cli-input-json file://fargate-task.json --region us-
east-1
```

3. Crie um arquivo denominado `ecs-service-discovery.json` com o conteúdo do serviço do ECS que você criará. Este exemplo usa a definição de tarefa criada na etapa anterior. Uma `awsvpcConfiguration` é necessária, pois o exemplo de definição de tarefa usa o modo de rede `awsvpc`.

```
{
    "cluster": "tutorial",
    "serviceName": "ecs-service-discovery",
    "taskDefinition": "tutorial-task-def",
    "serviceRegistries": [
       {
          "registryArn": "arn:aws:servicediscovery:region:aws_account_id:service/srv-
utcrh6wavdkggqtk"
       }
    ],
    "launchType": "FARGATE",
    "platformVersion": "LATEST",
    "networkConfiguration": {
       "awsvpcConfiguration": {
          "assignPublicIp": "ENABLED",
          "securityGroups": [ "sg-abcd1234" ],
          "subnets": [ "subnet-abcd1234" ]
       }
    },
    "desiredCount": 1
}
```

Crie seu serviço do ECS especificando o tipo de inicialização do Fargate e a versão `LATEST` da plataforma que oferece suporte à descoberta de serviço:

```
aws ecs create-service --cli-input-json file://ecs-service-discovery.json --region us-
east-1
```

Resultado:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:region:aws_account_id:service/ecs-service-
discovery",
        "serviceName": "ecs-service-discovery",
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/tutorial",
        "loadBalancers": [],
        "serviceRegistries": [
            {
                "registryArn": "arn:aws:servicediscovery:region:aws_account_id:service/
srv-utcrh6wavdkggqtk"
            }
        ],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/tutorial-
task-def:1",
        "deploymentConfiguration": {
```

```
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "id": "ecs-svc/9223370516993140842",
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
tutorial-task-def:1",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 0,
                "createdAt": 1519861634.965,
                "updatedAt": 1519861634.965,
                "launchType": "FARGATE",
                "platformVersion": "1.1.0",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-abcd1234"
                        ],
                        "securityGroups": [
                            "sg-abcd1234"
                        ],
                        "assignPublicIp": "ENABLED"
                    }
                }
            }
        ],
        "roleArn": "arn:aws:iam::aws_account_id:role/ECSServiceLinkedRole",
        "events": [],
        "createdAt": 1519861634.965,
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-abcd1234"
                ],
                "securityGroups": [
                    "sg-abcd1234"
                ],
                "assignPublicIp": "ENABLED"
            }
        }
    }
}
```

Etapa 3: Verificar Descoberta de serviço

Verifique se tudo foi criado corretamente consultando suas informações da descoberta de serviço. Após a configuração da descoberta de serviço, é possível consultá-la usando operações de API do AWS Cloud Map ou usando `dig` de dentro de sua VPC, como descrito abaixo.

Para verificar a configuração de descoberta de serviço

1. Usando o ID de serviço de descoberta de serviço, relacione as instâncias da descoberta de serviço:

```
aws servicediscovery list-instances --service-id srv-utcrh6wavdkggqtk --region us-
east-1
```

Resultado:

```
{
    "Instances": [
        {
            "Id": "16becc26-8558-4af1-9fbd-f81be062a266",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.31.87.2"
                "AWS_INSTANCE_PORT": "80",
                "AVAILABILITY_ZONE": "us-east-1a",
                "REGION": "us-east-1",
                "ECS_SERVICE_NAME": "ecs-service-discovery",
                "ECS_CLUSTER_NAME": "tutorial",
                "ECS_TASK_DEFINITION_FAMILY": "tutorial-task-def"
            }
        }
    ]
}
```

2. Usando o namespace e o serviço de descoberta de serviço, use parâmetros adicionais para consultar os detalhes sobre as instâncias de descoberta de serviço:

```
aws servicediscovery discover-instances --namespace-name tutorial --service-
name myapplication --query-parameters ECS_CLUSTER_NAME=tutorial --region us-east-1
```

Resultado:

```
{
    "Instances": [
        {
            "InstanceId": "16becc26-8558-4af1-9fbd-f81be062a266",
            "NamespaceName": "tutorial",
            "ServiceName": "ecs-service-discovery",
            "HealthStatus": "HEALTHY",
            "Attributes": {
                "AWS_INSTANCE_IPV4": "172.31.87.2"
                "AWS_INSTANCE_PORT": "80",
                "AVAILABILITY_ZONE": "us-east-1a",
                "REGION": "us-east-1",
                "ECS_SERVICE_NAME": "ecs-service-discovery",
                "ECS_CLUSTER_NAME": "tutorial",
                "ECS_TASK_DEFINITION_FAMILY": "tutorial-task-def"
            }
        }
    ]
}
```

3. Os registros DNS criados na zona hospedada do Route 53 para seu serviço de descoberta de serviço podem ser consultados com os comandos da AWS CLI a seguir.

   Usando o ID do namespace, obtenha informações sobre o namespace, o que inclui o ID da zona hospedada do Route 53:

```
aws servicediscovery get-namespace --id ns-uejictsjen2i4eeg --region us-east-1
```

Resultado:

```
{
    "Namespace": {
```

```
        "Id": "ns-uejictsjen2i4eeg",
        "Arn": "arn:aws:servicediscovery:region:aws_account_id:namespace/ns-
uejictsjen2i4eeg",
        "Name": "tutorial",
        "Type": "DNS_PRIVATE",
        "Properties": {
            "DnsProperties": {
                "HostedZoneId": "Z35JQ4ZFDRYPLV"
            }
        },
        "CreateDate": 1519777852.502,
        "CreatorRequestId": "9049a1d5-25e4-4115-8625-96dbda9a6093"
    }
}
```

4. Usando o ID da zona hospedada do Route 53, obtenha o conjunto de registros de recursos para a zona hospedada:

```
aws route53 list-resource-record-sets --hosted-zone-id Z35JQ4ZFDRYPLV --region us-
east-1
```

Resultado:

```
{
    "ResourceRecordSets": [
        {
            "Name": "tutorial.",
            "Type": "NS",
            "TTL": 172800,
            "ResourceRecords": [
                {
                    "Value": "ns-1536.awsdns-00.co.uk."
                },
                {
                    "Value": "ns-0.awsdns-00.com."
                },
                {
                    "Value": "ns-1024.awsdns-00.org."
                },
                {
                    "Value": "ns-512.awsdns-00.net."
                }
            ]
        },
        {
            "Name": "tutorial.",
            "Type": "SOA",
            "TTL": 900,
            "ResourceRecords": [
                {
                    "Value": "ns-1536.awsdns-00.co.uk. awsdns-hostmaster.amazon.com. 1
 7200 900 1209600 86400"
                }
            ]
        },
        {
            "Name": "myapplication.tutorial.",
            "Type": "A",
            "SetIdentifier": "16becc26-8558-4af1-9fbd-f81be062a266",
            "MultiValueAnswer": true,
            "TTL": 300,
            "ResourceRecords": [
                {
```

```
                    "Value": "172.31.87.2"
                }
            ]
        }
    ]
}
```

5. Também é possível consultar o DNS usando `dig` de uma instância na sua VPC com o seguinte comando:

```
dig +short myapplication.tutorial
```

Resultado:

```
172.31.87.2
```

Etapa 4: Limpeza

Ao concluir este tutorial, você deve limpar os recursos associados para evitar cobranças por recursos não utilizados.

Para limpar as instâncias de descoberta de serviço e os recursos do Amazon ECS

1. Cancele o registro das instâncias do serviço de descoberta de serviço:

```
aws servicediscovery deregister-instance --service-id srv-utcrh6wavdkggqtk --instance-id 16becc26-8558-4af1-9fbd-f81be062a266 --region us-east-1
```

Resultado:

```
{
    "OperationId": "xhu73bsertlyffhm3faqi7kumsmx274n-jh0zimzv"
}
```

2. Usando o `OperationId` do resultado anterior, verifique se o registro das instâncias de serviço de descoberta de serviço foi cancelado com êxito:

```
aws servicediscovery get-operation --operation-id xhu73bsertlyffhm3faqi7kumsmx274n-jh0zimzv --region us-east-1
```

Resultado:

```
{
    "Operation": {
        "Id": "xhu73bsertlyffhm3faqi7kumsmx274n-jh0zimzv",
        "Type": "DEREGISTER_INSTANCE",
        "Status": "SUCCESS",
        "CreateDate": 1525984073.707,
        "UpdateDate": 1525984076.426,
        "Targets": {
            "INSTANCE": "16becc26-8558-4af1-9fbd-f81be062a266",
            "ROUTE_53_CHANGE_ID": "C5NSRG1J4I1FH",
            "SERVICE": "srv-utcrh6wavdkggqtk"
        }
    }
}
```

3. Exclua o serviço de descoberta de serviço:

```
aws servicediscovery delete-service --id srv-utcrh6wavdkggqtk --region us-east-1
```

4. Exclua o namespace de descoberta de serviço:

```
aws servicediscovery delete-namespace --id ns-uejictsjen2i4eeg --region us-east-1
```

   Resultado:

```
{
    "OperationId": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj"
}
```

5. Usando o `OperationId` do resultado anterior, verifique se o namespace de descoberta de serviço foi excluído com êxito:

```
aws servicediscovery get-operation --operation-id c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj --region us-east-1
```

   Resultado:

```
{
    "Operation": {
        "Id": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj",
        "Type": "DELETE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1525984602.211,
        "UpdateDate": 1525984602.558,
        "Targets": {
            "NAMESPACE": "ns-rymlehshst7hhukh",
            "ROUTE_53_CHANGE_ID": "CJP2A2M86XW3O"
        }
    }
}
```

6. Atualize o serviço do Amazon ECS para que a contagem desejada seja `0`, que permite a exclusão:

```
aws ecs update-service --cluster tutorial --service ecs-service-discovery --desired-count 0 --force-new-deployment --region us-east-1
```

7. Exclua o serviço de Amazon ECS:

```
aws ecs delete-service --cluster tutorial --service ecs-service-discovery --region us-east-1
```

8. Exclua o cluster do Amazon ECS:

```
aws ecs delete-cluster --cluster tutorial --region us-east-1
```

Criar um serviço

Quando você cria um serviço Amazon ECS, você especifica os parâmetros básicos que definem o que compõe o serviço e como ele deve se comportar. Esses parâmetros criam uma definição de serviço.

Outra opção é configurar recursos adicionais, como load balancer do Elastic Load Balancing para distribuir o tráfego através dos contêineres no seu serviço. Para obter mais informações. Para obter mais informações, consulte Balanceamento de carga do serviço (p. 115). Você deve verificar se as instâncias de contêiner podem receber tráfego dos load balancers. É possível permitir o tráfego para todas as portas nas instâncias de contêiner a partir do security group do load balancer para garantir que o tráfego possa alcançar todos os contêineres que usam portas dinamicamente atribuídas.

Os documentos a seguir apresentarão cada passo do assistente de criação de serviços no Console de gerenciamento da AWS.

Tópicos

- Etapa 1: configuração de parâmetros básicos de serviço (p. 149)
- Etapa 2: configuração de uma rede (p. 151)
- Etapa 3: (opcional) configurar o serviço para usar um load balancer (p. 151)
- Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço (p. 156)
- Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling (p. 157)
- Etapa 6: consultar e criar o serviço (p. 160)

Etapa 1: configuração de parâmetros básicos de serviço

Todos os serviços exigem alguns parâmetros básicos de configuração que definem o serviço, como a definição de tarefa a ser usada, qual cluster o serviço deve executar, quantas tarefas devem ser atribuídas para o serviço etc. Isso é chamado de definição de serviço. Para obter mais informações sobre os parâmetros definidos em uma definição de serviço, consulte Parâmetros de definição de serviço (p. 101).

Este procedimento abrange a criação de um serviço com os parâmetros básicos de definição de serviço necessários. Após configurar esses parâmetros, será possível criar o serviço ou passar para os procedimentos de configuração opcional de definição de serviço, como a configuração do serviço para usar um load balancer.

Para configurar os parâmetros básicos de definição de serviço

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a Região em que seu cluster está localizado.
3. No painel de navegação, selecione Task Definitions e selecione a definição da tarefa da qual criar seu serviço.
4. Na página Task Definition name, selecione a revisão da definição da tarefa da qual criar seu serviço.
5. Revise a definição de tarefa e selecione Actions, Create Service.
6. Na página Configure service, preencha os seguintes parâmetros de acordo:

   - Em Launch type (Tipo de inicialização): escolha se o serviço deve executar tarefas na infraestrutura Fargate ou nas instâncias de contêiner do Amazon EC2 que você mantém. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).
   - Platform version (Versão da plataforma): se você escolheu o tipo de inicialização Fargate, selecione a versão da plataforma a ser usada.
   - Cluster: selecione o cluster no qual criar o serviço.
   - Service name: digite um nome exclusivo para o serviço.
   - Service type (Tipo de serviço): Selecione uma estratégia de agendamento para seu serviço. Para obter mais informações, consulte Conceitos do programador de serviço (p. 99).
   - Em Number of tasks (Número de tarefas): se você escolheu o tipo de serviço `REPLICA`, digite o número de tarefas para executar e manter em seu cluster.

Note

Se o tipo de inicialização for `EC2` e a sua definição de tarefa usa os mapeamentos de porta de host estático em suas instâncias de contêiner, você precisa pelo menos de uma instância de contêiner com a porta especificada disponível no cluster para cada tarefa em seu serviço. Essa restrição não se aplicará se a sua definição de tarefa usar os mapeamentos de porta de host dinâmico com o modo de rede `bridge`. Para obter mais informações, consulte portMappings (p. 37).

- Se você estiver usando o tipo de implantação Rolling update (Atualização contínua), preencha os seguintes parâmetros:
  - Minimum healthy percent: especifique um limite menor no número de tarefas do serviço que devem permanecer no estado `RUNNING` durante uma implantação, como uma porcentagem do número desejado do serviço (arredondado para o número inteiro mais próximo). Por exemplo, se o serviço tiver um número desejado de 4 tarefas e uma porcentagem íntegra mínima de 50%, o programador poderá interromper 2 tarefas existentes para liberar a capacidade do cluster antes de iniciar 2 novas tarefas. As tarefas para serviços que não usam um load balancer serão consideradas íntegras se estiverem no estado `RUNNING`. As tarefas para serviços que usam um load balancer são consideradas íntegras se tiverem o status `RUNNING` e quando a instância de contêiner na qual o load balancer está hospedado for relatada como íntegra pelo load balancer. O valor padrão para a porcentagem de integridade mínima é de 50% no console e 100% com a AWS CLI ou SDKs.
  - Maximum percent: especifique um limite inferior no número de tarefas do serviço que são permitidas no estado `RUNNING` ou `PENDING` durante uma implantação, como uma porcentagem de número de tarefas desejadas do serviço (arredondado para o número inteiro mais próximo). Por exemplo, se o serviço tiver um número desejado de 4 tarefas e um valor máximo de porcentagem de 200%, o programador poderá iniciar 4 tarefas novas antes de interromper as 4 tarefas mais antigas. Isso é fornecido desde que os recursos de cluster necessários para isso estejam disponíveis. O valor padrão para a porcentagem máxima é 200%.

7. Na página Deployments (Implantações), preencha os seguintes parâmetros de acordo:
   - Em Deployment type (Tipo de implantação), escolha se o serviço deve usar uma implantação de atualização contínua ou uma implantação azul/verde usando o AWS CodeDeploy. Para obter mais informações, consulte Tipos de implantação do Amazon ECS (p. 107).
   - Se você selecionou o tipo de implantação azul/verde, em Service role for CodeDeploy (Função de serviço para AWS CodeDeploy), escolha a função de serviço do IAM para o AWS CodeDeploy. Para obter mais informações, consulte Função do IAM CodeDeploy do Amazon ECS (p. 234).
8. In the Task tagging configuration section, complete the following steps:
   a. Select Enable ECS managed tags if you want Amazon ECS to automatically tag the tasks in the service with the Amazon ECS managed tags. For more information, see Tagging Your Amazon ECS Resources.
   b. For Propagate tags from, select one of the following:
      - Do not propagate – This option will not propagate any tags to the tasks in the service.
      - Service – This option will propagate the tags specified on your service to each of the tasks in the service.
      - Task Definitions – This option will propagate the tags specified in the task definition of a task to the tasks in the service.

      Note

      If you specify a tag with the same `key` in the Tags section, it will override the tag propagated from either the service or the task definition.

9. Na seção Tags, especifique a chave e o valor de cada tag para associá-la à tarefa. Para obter mais informações, consulte Como marcar seus recursos do Amazon ECS.
10. Escolha Next step (Próxima etapa) e vá até Etapa 2: configuração de uma rede (p. 151).

Etapa 2: configuração de uma rede

Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc`, você deve fazer as configurações de VPC, sub-rede e do security group para seu serviço.

Se a definição da tarefa do seu serviço não usar o modo de rede `awsvpc`, você poderá passar para a próxima etapa, Etapa 3: (opcional) configurar o serviço para usar um load balancer (p. 151).

Para fazer as configurações de VPC e security group para o seu serviço

1. Caso ainda não tenha feito, siga os procedimentos de configuração de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Em Cluster VPC (VPC de cluster), se tiver selecionado o tipo de inicialização EC2, escolha a VPC na qual as instâncias de contêineres residem. Se tiver selecionado o tipo de inicialização Fargate, selecione a VPC que as tarefas de Fargate deveriam usar. Certifique-se de que a VPC escolhida não está configurada para exigir locação de hardware dedicada, porque isso não é compatível com tarefas de Fargate.
3. Para Subnets, escolha as sub-redes disponíveis para posicionamento da sua tarefa de serviço.

   Important

   Somente as sub-redes privadas são compatíveis com o modo de rede `awsvpc`. Como as tarefas não recebem endereços IP públicos, um gateway NAT é necessário para o acesso à Internet de saída. O tráfego de entrada de Internet deve ser roteado por meio de um load balancer.
4. Para Security groups, foi criado um security group para as tarefas do seu serviço, que permitem tráfego HTTP da Internet (0.0.0.0/0). Para editar o nome ou as regras deste security group, ou para escolher um security group existente, escolha Edit e modifique as configurações do seu security group.
5. Para Auto-assign Public IP (Atribuir IP publico automaticamente), defina se suas tarefas devem receber um endereço IP público. Se você estiver usando as tarefas Fargate, um endereço IP público precisará ser atribuído à interface de rede elástica da tarefa. A interface de rede deve ter uma rota para a Internet ou um gateway NAT que possa encaminhar solicitações para a Internet, para que a tarefa extraia imagens de contêiner.
6. Se você estiver configurando seu serviço para usar um load balancer ou se estiver usando o tipo de implantação verde/azul, prossiga para Etapa 3: (opcional) configurar o serviço para usar um load balancer (p. 151). Se você não estiver configurando o serviço para usar um load balancer, poderá escolher None como o tipo de load balancer e passar para a próxima seção, Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling (p. 157).

Etapa 3: (opcional) configurar o serviço para usar um load balancer

Os serviços podem ser configurados para usar um load balancer para distribuir o tráfego de entrada para as tarefas no seu serviço. Se o seu serviço estiver usando o tipo de implantação de atualização contínua, isso é opcional. Se o seu serviço estiver usando o tipo de implantação azul/verde, será necessário usar um Balanceador de carga de aplicações ou um Load balancer de rede.

Se você não estiver configurando o serviço para usar um load balancer, poderá escolher None como o tipo de load balancer e passar para a próxima seção, Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço (p. 156).

Se você tiver um load balancer Elastic Load Balancing disponível e configurado, será possível anexá-lo ao serviço por meio dos procedimentos a seguir, ou configurar um novo load balancer. Para obter mais informações, consulte Como criar um balanceador de carga (p. 119).

> **Important**
>
> Antes de seguir esses procedimentos, você deve criar seus recursos do load balancer do Elastic Load Balancing.

Tópicos

- Configurar um load balancer para o tipo de implantação de atualização contínua (p. 152)
- Configurar um load balancer para o tipo de implantação azul/verde (p. 154)

Configurar um load balancer para o tipo de implantação de atualização contínua

Se as tarefas do serviço demorarem para ser iniciadas e responder às verificações de integridade do Elastic Load Balancing, você poderá especificar um período de carência de verificação de integridade de até 2.147.483.647 segundos. Durante esse tempo, o programador de serviços ignora o status da verificação de integridade. Esse período de carência pode evitar que o programador do serviço marque tarefas como não íntegras e as interrompa antes de terem tempo de surgir. Isso será válido somente se o serviço estiver configurado para usar um load balancer.

Para configurar um período de carência da verificação de integridade

1. Caso ainda não tenha feito, siga os procedimentos de configuração de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Em Health check grace period (Período de carência da verificação de integridade): informe o período, em segundos, que o programador de serviços do Amazon ECS deve ignorar verificações de destino do Elastic Load Balancing não íntegras depois que uma tarefa tiver sido iniciada pela primeira vez.

Para configurar seu serviço para usar um load balancer, você deve escolher o tipo de load balancer a ser usado com seu serviço.

Para escolher um tipo de load balancer

1. Caso ainda não tenha feito, siga os procedimentos de criação de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Em Load balancer type, escolha o tipo de load balancer para uso com o seu serviço:

   Balanceador de carga de aplicações

   Permite que os contêineres usem o mapeamento de porta de host dinâmico, o que permite que você coloque várias tarefas usando a mesma porta em uma única instância de contêiner. Vários serviços podem usar a mesma porta de escuta em um único load balancer com caminhos e roteamento baseados em regra.

   Load balancer de rede

   Permite que os contêineres usem o mapeamento de porta de host dinâmico, o que permite que você coloque várias tarefas usando a mesma porta em uma única instância de contêiner. Vários serviços podem usar a mesma porta de escuta em um único load balancer com roteamento baseado em regra.

Classic Load Balancer

Requer mapeamentos de porta de host estático (apenas uma tarefa permitida por instância de contêiner); roteamento baseado em regra e caminhos não são suportados.

Recomendamos que você use Balanceador de carga de aplicaçõess para os serviços do Amazon ECS, de maneira que possa aproveitar os recursos avançados disponíveis para eles.

3. Em Select IAM role for service (Selecionar função do IAM para o serviço), escolha Create new role (Criar nova função) para criar uma nova função para o seu serviço, ou selecione uma função do IAM existente a ser usada para o seu serviço (por padrão, `ecsServiceRole`).

   Important

   Se você optar por usar uma função `ecsServiceRole` do IAM existente, deverá verificar se a função tem as permissões adequadas para usar Balanceador de carga de aplicaçõess e Classic Load Balancers. Para obter mais informações, consulte Função do IAM programador de serviço do Amazon ECS (p. 232).

4. Em ELB Name, escolha o nome do load balancer para uso com o seu serviço. Somente load balancers que correspondam ao tipo selecionado anteriormente estarão visíveis aqui.
5. A próxima etapa depende do tipo do load balancer para o seu serviço. Se você tiver escolhido um Balanceador de carga de aplicações, siga as etapas em Para configurar um Balanceador de carga de aplicações (p. 153). Se você tiver escolhido um Load balancer de rede, siga as etapas em Para configurar um Load balancer de rede (p. 154).

Para configurar um Balanceador de carga de aplicações

1. Em Container to load balance, escolha o contêiner e a combinação de portas a partir de sua definição de tarefa que seu load balancer deve distribuir tráfego e escolha Add to load balancer.
2. Em Listener port (Porta do listener), escolha a porta e o protocolo do listener criado em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável) ou escolha create new (criar novo) para criar um novo listener e, em seguida, insira um número de porta e escolha um protocolo de porta em Listener protocol (Protocolo do listener).
3. Em Target group name (Nome do grupo de destino), escolha o grupo de destino que você criou em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável), ou escolha create new (criar novo) para criar um novo grupo de destino.

   Important

   Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), seu grupo de destino precisará usar `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.

4. (Opcional) Se você escolher criar um novo grupo de destino, preencha os seguintes campos da seguinte forma:

   - Em Target group name (Nome do grupo de destino), um nome padrão é fornecido a você.
   - Em Target group protocol, insira o protocolo a ser usado para rotear tráfico para suas tarefas.
   - Em Path pattern, se a sua escuta não tiver regras, o caminho padrão (`/`) será usado. Se a escuta já tiver uma regra padrão, você deverá inserir um caminho padrão que corresponda ao tráfego que você deseja enviar ao grupo de destino do serviço. Por exemplo, se o seu serviço for um aplicativo web denominado `web-app`, e você deseja um tráfego que corresponda a `http://my-elb-url/web-app` para rotear para o seu serviço, você deverá inserir `/web-app*` como seu padrão de caminho. Para obter mais informações, consulte ListenerRules no Guia do usuário para Application Load Balancers.

- Em Health check path (Caminho da verificação de integridade), insira o caminho para o qual o load balancer deve enviar pings de verificação de integridade.

5. Ao concluir a configuração do Balanceador de carga de aplicações, selecione Next step (Próxima etapa).

Para configurar um Load balancer de rede

1. Em Container to load balance, escolha o contêiner e a combinação de portas a partir de sua definição de tarefa que seu load balancer deve distribuir tráfego e escolha Add to load balancer.
2. Em Listener port (Porta do listener), escolha a porta e o protocolo do listener criado em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável) ou escolha create new (criar novo) para criar um novo listener e, em seguida, insira um número de porta e escolha um protocolo de porta em Listener protocol (Protocolo do listener).
3. Em Target group name (Nome do grupo de destino), escolha o grupo de destino que você criou em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável), ou escolha create new (criar novo) para criar um novo grupo de destino.

   Important

   Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), seu grupo de destino precisará usar `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.
4. (Opcional) Se você escolher criar um novo grupo de destino, preencha os seguintes campos da seguinte forma:

   - Em Target group name (Nome do grupo de destino), um nome padrão é fornecido a você.
   - Em Target group protocol, insira o protocolo a ser usado para rotear tráfico para suas tarefas.
   - Em Health check path (Caminho da verificação de integridade), insira o caminho para o qual o load balancer deve enviar pings de verificação de integridade.
5. Ao concluir a configuração do Load balancer de rede, selecione Next Step (Próxima etapa).

Configurar um load balancer para o tipo de implantação azul/verde

Para configurar o serviço que usa o tipo de implantação azul/verde para usar um load balancer, você deve usar um Balanceador de carga de aplicações ou um Load balancer de rede.

Para escolher um tipo de load balancer

1. Caso ainda não tenha feito, siga os procedimentos de criação de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Em Load balancer type, escolha o tipo de load balancer para uso com o seu serviço:

   Balanceador de carga de aplicações

   Permite que os contêineres usem o mapeamento de porta de host dinâmico, o que permite que você coloque várias tarefas usando a mesma porta em uma única instância de contêiner. Vários serviços podem usar a mesma porta de escuta em um único load balancer com caminhos e roteamento baseados em regra.

   Load balancer de rede

   Permite que os contêineres usem o mapeamento de porta de host dinâmico, o que permite que você coloque várias tarefas usando a mesma porta em uma única instância de contêiner. Vários

serviços podem usar a mesma porta de escuta em um único load balancer com roteamento baseado em regra.

Recomendamos que você use Balanceador de carga de aplicaçõess para os serviços do Amazon ECS, de maneira que possa aproveitar os recursos avançados disponíveis para eles.

3. Em Load balancer name (Nome do load balancer), escolha o nome do load balancer para uso com o seu serviço. Somente load balancers que correspondam ao tipo selecionado anteriormente estarão visíveis aqui.
4. A próxima etapa depende do tipo do load balancer para o seu serviço. Se você tiver escolhido um Balanceador de carga de aplicações, siga as etapas em Para configurar um Balanceador de carga de aplicações (p. 153). Se você tiver escolhido um Load balancer de rede, siga as etapas em Para configurar um Load balancer de rede (p. 154).

Para configurar um Balanceador de carga de aplicações para o tipo de implantação azul/verde

1. Em Container to load balance, escolha o contêiner e a combinação de portas a partir de sua definição de tarefa que seu load balancer deve distribuir tráfego e escolha Add to load balancer.
2. Em Production listener port (Porta do listener de produção), escolha a porta e o protocolo do listener criado em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável) ou escolha create new (criar novo) para criar um novo listener e, em seguida, insira um número de porta e escolha um protocolo de porta em Protocolo do listener de produção.
3. (Opcional) Selecione Test listener (Listener de teste) se você deseja configurar a porta e o protocolo de um listener no seu load balancer para testar atualizações para o serviço antes de rotear o tráfego para seu novo conjunto de tarefas. Execute a etapa a seguir:

   - Em Test listener port (Porta do listener de teste), escolha a porta e o protocolo do listener pelo qual você deseja testar o tráfego ou escolha create new (criar novo) para criar um novo listener de teste e, em seguida, insira um número de porta e escolha um protocolo de porta em Test listener protocol (Protocolo do listener de teste).
4. Para implantações azul/verde, dois grupos de destino são necessários. Cada grupo de destino se vincula a um conjunto de tarefas separado na implantação. Execute as etapas a seguir:

   a. Em Target group 1 name (Nome do grupo de destino 1), escolha o grupo de destino que você criou em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável), ou escolha create new (criar novo) para criar um novo grupo de destino.

      Important

      Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), seu grupo de destino precisará usar `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.

   b. (Opcional) Se você escolher criar um novo grupo de destino, preencha os seguintes campos da seguinte forma:

      - Em Target group name, digite um nome para o grupo de destino.
      - Em Target group protocol, insira o protocolo a ser usado para rotear tráfico para suas tarefas.
      - Em Path pattern, se a sua escuta não tiver regras, o caminho padrão (`/`) será usado. Se a escuta já tiver uma regra padrão, você deverá inserir um caminho padrão que corresponda ao tráfego que você deseja enviar ao grupo de destino do serviço. Por exemplo, se o seu serviço for um aplicativo web denominado `web-app`, e você deseja um tráfego que corresponda a `http://my-elb-url/web-app` para rotear para o seu serviço, você deverá inserir `/web-app*` como seu padrão de caminho. Para obter mais informações, consulte ListenerRules no Guia do usuário para Application Load Balancers.

- Em Health check path (Caminho da verificação de integridade), insira o caminho para o qual o load balancer deve enviar pings de verificação de integridade.

c. Repita as etapas para o grupo de destino 2.

d. Ao concluir a configuração do Balanceador de carga de aplicações, selecione Next step (Próxima etapa). Navegue até Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço (p. 156).

Para configurar um Load balancer de rede para o tipo de implantação azul/verde

1. Em Container to load balance, escolha o contêiner e a combinação de portas a partir de sua definição de tarefa que seu load balancer deve distribuir tráfego e escolha Add to load balancer.
2. Em Listener port (Porta do listener), escolha a porta e o protocolo do listener criado em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável) ou escolha create new (criar novo) para criar um novo listener e, em seguida, insira um número de porta e escolha um protocolo de porta em Listener protocol (Protocolo do listener).
3. Em Target group name (Nome do grupo de destino), escolha o grupo de destino que você criou em Como criar um Balanceador de carga de aplicações (p. 121) (se aplicável), ou escolha create new (criar novo) para criar um novo grupo de destino.

   Important

   Se a definição de tarefa do seu serviço usar o modo de rede `awsvpc` (exigido para o tipo de execução Fargate), seu grupo de destino precisará usar `ip` como tipo de destino, e não `instance`. Isso ocorre porque as tarefas que usam o modo de rede `awsvpc` estão associadas a uma interface de rede elástica, e não a uma instância do Amazon EC2.
4. (Opcional) Se você escolher criar um novo grupo de destino, preencha os seguintes campos da seguinte forma:

   - Em Target group name, digite um nome para o grupo de destino.
   - Em Target group protocol, insira o protocolo a ser usado para rotear tráfico para suas tarefas.
   - Em Health check path (Caminho da verificação de integridade), insira o caminho para o qual o load balancer deve enviar pings de verificação de integridade.
5. Ao concluir a configuração do Load balancer de rede, selecione Next Step (Próxima etapa). Navegue até Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço (p. 156).

Etapa 4: (opcional) configurar o serviço para usar o Descoberta de serviço

O serviço do Amazon ECS também pode habilitar a integração da descoberta de serviço, o que permite que o serviço seja detectável por meio de DNS. Para obter mais informações, consulte Descoberta de serviço (p. 137).

Se você não estiver configurando o serviço para usar uma descoberta de serviço, poderá passar para a próxima seção, Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling (p. 157).

Para configurar o descoberta de serviço

1. Caso ainda não tenha feito, siga os procedimentos de configuração de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Na página Configure network (Configurar rede), selecione Enable descoberta de serviço integration (Habilitar integração).
3. Em Namespace, selecione um namespace do Amazon Route 53 existente se você tiver um. Caso contrário, selecione create new private namespace (criar novo namespace privado).

4. Se você criar um namespace, em Namespace name (Nome de namespace), insira um nome descritivo para o namespace. Este é o nome usado para a zona hospedada de Amazon Route 53.
5. Em Configure descoberta de serviço service (Configurar serviço), opte por criar um serviço de descoberta de serviço ou selecionar um existente.
6. Se você criar um serviço de descoberta de serviço, em Service discovery name (Nome da descoberta de serviço), insira um nome descritivo para o serviço descoberta de serviço. Ele é usado como o prefixo dos registros DNS a serem criados.
7. Selecione Enable ECS task health propagation (Habilitar propagação de integridade de tarefa do ECS) se você quiser que as verificações de integridade sejam habilitadas para seu serviço de descoberta de serviço.
8. Em DNS record type (Tipo de registro DNS), selecione o tipo de registro DNS para criar para o seu serviço. Amazon ECS descoberta de serviço só oferece suporte a registros A e SRV, dependendo do modo de rede que especifica a sua definição de tarefa. Para obter mais informações sobre esses tipos de registro, consulte DnsRecord.

    - Se a definição de tarefa que a sua tarefa de serviço especifica usa o modo de rede do `bridge` ou `host`, somente os registros do tipo SRV são suportados. Escolha o nome do contêiner e a combinação de portas a serem associadas ao registro.
    - Se a definição de tarefa que a sua tarefa de serviço especifica usa o modo de rede do `awsvpc`, selecione o tipo de registro A ou SRV. Se o registro DNS do tipo A estiver selecionado, vá para a próxima etapa. Se o tipo SRV estiver selecionado, especifique a porta na qual o serviço pode ser encontrado ou o nome do contêiner e a combinação de portas a serem associadas ao registro.
9. Em TTL, insira o tempo de vida (TTL), em segundos, do cache do registro de recurso. Esse valor determina por quanto tempo um conjunto de registros é armazenado em cache pelos resolvedores de DNS e navegadores.
10. Escolha Next step (Próxima etapa) para prosseguir e vá até Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling (p. 157).

Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling

O serviço do Amazon ECS pode ser configurado opcionalmente para usar Auto Scaling para ajustar a contagem desejada para cima ou para baixo em resposta aos alarmes do CloudWatch.

O Serviço Auto Scaling do Amazon ECS oferece suporte aos seguintes tipos de políticas de escalabilidade:

- Políticas de escalabilidade de rastreamento de destino (p. 130)—Aumenta ou diminui o número de tarefas que o serviço executa com base em um valor de destino para uma métrica específica. Isso é semelhante à forma como o termostato mantém a temperatura da casa. Você seleciona a temperatura, e o termostato faz o resto.
- Políticas de escalabilidade em etapas (p. 135)—Aumenta ou diminui o número de tarefas que o serviço executa com base em um conjunto de ajustes de escalabilidade, conhecidos como ajustes em etapas, que variam com base no tamanho da violação do alarme.

Para obter mais informações, consulte Serviço Auto Scaling (p. 129).

Para configurar os parâmetros básicos do Serviço Auto Scaling

1. Caso ainda não tenha feito, siga os procedimentos de configuração de serviço básicos em Etapa 1: configuração de parâmetros básicos de serviço (p. 149).
2. Na página Set Auto Scaling, selecione Configure Service Auto Scaling to adjust your service's desired count.

3. Em Minimum number of tasks (Número mínimo de tarefas), insira o limite inferior do número de tarefas para uso do Serviço Auto Scaling. A contagem desejada do seu serviço não será automaticamente ajustada para um valor abaixo desse.
4. Em Desired number of tasks (Número de tarefas desejadas), esse campo é preenchido antecipadamente com o valor já inserido. Você pode alterar a contagem desejada do serviço no momento, mas esse valor deve ser entre o número mínimo e máximo de tarefas especificado nessa página.
5. Em Maximum number of tasks (Número máximo de tarefas), insira o limite superior do número de tarefas para uso do Serviço Auto Scaling. A contagem desejada do seu serviço não será automaticamente ajustada um valor acima desse.
6. Em IAM role for Service Auto Scaling (Função do IAM para Auto Scaling de serviço), escolha uma função do IAM a fim de autorizar o serviço Application Auto Scaling a ajustar a contagem desejada do seu serviço em seu nome. Se você não tiver criado essa função anteriormente, escolha Create new role e a função será criada para você. Para referência futura, a função criada para você é chamada `ecsAutoscaleRole`. Para obter mais informações, consulte Função do IAM Serviço Auto Scaling do Amazon ECS (p. 237).
7. Os procedimentos a seguir fornecem as etapas para a criação de políticas de rastreamento de destino ou de escalabilidade de etapas para o seu serviço. Escolha o tipo de política de escalabilidade desejado.

Estas etapas o ajudarão a criar políticas de escalabilidade de rastreamento de destino e alarmes do CloudWatch que podem ser usados para disparar ações de escalabilidade para o seu serviço. Você pode criar um alarme de expansão para aumentar a contagem desejada de serviços, e um alarme de redução para diminuir a contagem desejada de serviços.

Para configurar políticas de escalabilidade de rastreamento de destino para o serviço

1. Em Scaling policy type (Tipo de política de escalabilidade), escolha Target tracking (Rastreamento de destino).
2. Em Policy name (Nome da política), insira um nome descritivo para a política.
3. Em ECS service metric (Métrica do serviço do ECS), escolha a métrica a ser rastreada.
4. Em Target value (Valor de destino), insira o valor da métrica que a política deve manter.
5. Em Scale-out cooldown period (Período de desaquecimento após expansão), insira a quantidade de tempo, em segundos, após a conclusão de uma ação de expansão antes que uma outra atividade de expansão possa iniciar. Durante esse período, os recursos que foram iniciados não contribuem para a métrica de grupo do Auto Scaling.
6. Em Scale-in cooldown period (Período de desaquecimento após redução), insira a quantidade de tempo, em segundos, após a conclusão de uma ação de redução antes que uma outra atividade de redução possa iniciar. Durante esse período, os recursos que foram iniciados não contribuem para a métrica de grupo do Auto Scaling.
7. (Opcional) Para desativar as ações de redução para esta política, escolha Disable scale-in (Desabilitar a redução). Isso permite que você crie uma política de escalabilidade distinta para a redução mais tarde.
8. Escolha Próxima etapa.

Essas etapas o ajudarão a criar políticas de escalabilidade em etapas e alarmes do CloudWatch que podem ser usados para disparar ações de escalabilidade para o seu serviço. Você pode criar um alarme de Scale out para aumentar a contagem desejada de seu serviço, e um alarme de Scale in para diminuir a contagem desejada de seu serviço.

Para configurar políticas de escalabilidade em etapas para o serviço

1. Em Scaling policy type (Tipo de política de escalabilidade), escolha Step scaling (Escalabilidade em etapas).

2. Em Policy name (Nome da política), insira um nome descritivo para a política.
3. Em Execute policy when (Executar a política quando), selecione o alarme do CloudWatch para aumentar ou reduzir o serviço.

   Você pode usar um alarme do CloudWatch existente que você criou anteriormente ou optar por criar um novo alarme. O fluxo de trabalho Create new alarm (Criar novo alarme) permite criar alarmes do CloudWatch com base no `CPUUtilization` e no `MemoryUtilization` do serviço que está sendo criado. Para usar outras métricas, você pode criar seu alarme no console do CloudWatch e, em seguida, retornar para este assistente para escolher esse alarme.
4. (Opcional) Se você escolher criar um novo alarme, conclua as seguintes etapas.

   a. Em Alarm name, insira um nome descritivo para o alarme. Por exemplo, se o seu alarme precisar ser acionado quando a sua utilização de CPU exceder 75%, você poderá chamar o alarme *`service_name`*`-cpu-gt-75`.

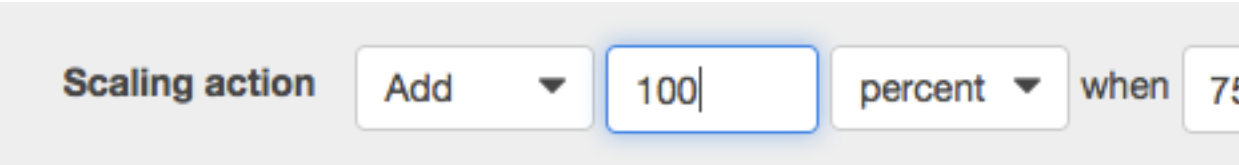
   b.

   c. Em Alarm threshold, digite as seguintes informações para configurar seu alarme:

      - Escolha a estatística do CloudWatch para seu alarme (o valor padrão de Average (Média) funciona em muitos casos). Para obter mais informações, consulte Estatísticas no Guia do usuário do Amazon CloudWatch.
      - Escolha o operador de comparação para o seu alarme e insira o valor que o operador de comparação usará na verificação (por exemplo, `>` e `75`).
      - Insira o número de períodos consecutivos antes do alarme ser acionado e a duração do período. Por exemplo, dois períodos consecutivos de 5 minutos precisariam de 10 minutos antes de o alarme ser acionado. Como as tarefas do Amazon ECS podem ser reduzidas ou aumentadas rapidamente, considere a utilização de um número pequeno de períodos consecutivos e uma duração de período curta para reagir a alarmes assim que possível.

   d. Escolha Salvar.
5. Em Scaling action, insira as seguintes informações para configurar como o seu serviço responde ao alarme:

   - Escolha se deseja adicionar, subtrair ou definir uma contagem desejada para o seu serviço.
   - Se você decidir adicionar ou subtrair tarefas, insira o número de tarefas (ou porcentagem de tarefas existentes) para adicionar ou subtrair quando a ação de escalabilidade é acionada. Se você optar por definir a contagem desejada, insira a contagem desejada para o seu serviço quando a ação de escalabilidade é acionada.
   - (Opcional) Se você decidir adicionar ou subtrair tarefas, escolha se o valor anterior é usado como um número inteiro ou um valor percentual da contagem desejada existente.
   - Insira o limite inferior do seu ajuste de escalabilidade incremental. Por padrão, para a sua primeira ação de escalabilidade, esse valor é o valor de métrica em que o alarme é acionado. Por exemplo, a seguinte ação de escalabilidade adiciona 100% da contagem desejada existente quando a utilização da CPU é maior do que 75%.

6. (Opcional) Você pode repetir Step 5 (p. 159) para configurar várias ações de escalabilidade para um único alarme (por exemplo, para adicionar uma tarefa se a utilização da CPU ficar entre 75% e 85%, e para adicionar duas tarefas se a utilização da CPU for maior do que 85%).
7. (Opcional) Se você decidir adicionar ou subtrair uma porcentagem da contagem desejada existente, insira um valor incremental mínimo em Add tasks in increments of *`N`* task(s).
8. Em Cooldown period, insira o número de segundos entre ações de escalabilidade.

9. Repita Step 1 (p. 158) a Step 8 (p. 159) para a política de Scale in (Escalar em) e escolha Save (Salvar).
10. Escolha Next step (Próxima etapa) para prosseguir e vá até Etapa 6: consultar e criar o serviço (p. 160).

Etapa 6: consultar e criar o serviço

Após configurar os parâmetros básicos de definição de serviço e configurar opcionalmente a rede, o load balancer, a descoberta de serviço e a escalabilidade automática do serviço, você poderá revisar sua configuração. Em seguida, escolha Create Service (Criar serviço) para terminar de criar o seu serviço.

> Note
>
> Depois que você cria um serviço, o Nome de região da Amazon (ARN) ou o nome do load balancer, o nome do contêiner e porta do contêiner especificados na definição do serviço não podem mais ser alterados. Você não pode adicionar, remover ou alterar a configuração do load balancer de um serviço já existente. Se você atualizar a definição de tarefa para o serviço, o contêiner e a porta de contêiner especificados quando o serviço foi criado deverão permanecer na definição da tarefa.

Atualizar um serviço

É possível atualizar um serviço em execução para alterar o número de tarefas mantidas por um serviço ou a definição de tarefa usada pelas tarefas. Se suas tarefas estiverem usando o tipo de inicialização Fargate, é possível alterar a versão da plataforma usada pelo seu serviço. Se você tiver um aplicativo que precisa de mais capacidade, é possível expandir seu serviço. Se você tiver capacidade não utilizada para reduzir, é possível reduzir o número de tarefas desejadas no serviço e liberar recursos.

Se você atualizou a imagem de Docker do aplicativo, será possível criar uma nova definição de tarefa com essa imagem e implantá-la no serviço.

> Note
>
> Se a imagem do Docker atualizada usar a mesma tag que a existente na definição de tarefa existente para seu serviço (por exemplo, `my_image:latest`), não é necessário criar outra revisão da sua definição de tarefa. É possível atualizar o serviço usando o procedimento a seguir. Mantenha as configurações atuais do seu serviço e selecione Force new deployment (Forçar nova implantação). As novas tarefas executadas pela implantação obtêm a combinação atual de imagem/tag do seu repositório ao iniciarem. A opção Force new deployment (Forçar nova implantação) também é usada ao atualizar uma tarefa Fargate para usar uma versão mais atual da plataforma ao especificar `LATEST`. Por exemplo, se você especificar `LATEST`, suas tarefas em execução estiverem usando a versão `1.0.0` da plataforma e quiser que elas sejam reiniciadas usando uma versão mais recente da plataforma.

O programador de serviço usa os parâmetros de porcentagem íntegros mínimos e máximos (na configuração de implantação para o serviço) para determinar a estratégia de implantação.

Se um serviço estiver usando o tipo de implantação de atualização contínua (`ECS`), a porcentagem de integridade mínima representa um limite menor no número de tarefas em um serviço que devem permanecer no estado `RUNNING` durante uma implantação, como uma porcentagem do número desejado de tarefas (arredondado para cima, para o número inteiro mais próximo). O parâmetro também se aplica enquanto qualquer instância de contêiner estiver no estado `DRAINING` se o serviço contiver tarefas usando o tipo de inicialização EC2. Esse parâmetro permite implantar sem usar a capacidade adicional de cluster.

Por exemplo, se o serviço tiver um número desejado de 4 tarefas e uma porcentagem íntegra mínima de 50%, o programador poderá interromper 2 tarefas existentes para liberar a capacidade do cluster antes de iniciar 2 novas tarefas. As tarefas para serviços que não usam um load balancer serão consideradas íntegras se estiverem no estado `RUNNING`. As tarefas para serviços que usam um load balancer são consideradas íntegras se estiverem com o status `RUNNING` e forem relatadas como íntegras pelo load balancer. O valor padrão para a porcentagem mínima de integridade é 100%.

Se um serviço estiver usando o tipo de implantação de atualização contínua (`ECS`), o parâmetro porcentagem máxima representa um limite superior no número de tarefas em um serviço que é permitido no estado `RUNNING` ou `PENDING` durante uma implantação, como uma porcentagem do número desejado de tarefas (arredondado para baixo, para o menor número inteiro mais próximo). O parâmetro também se aplica enquanto qualquer instância de contêiner estiver no estado `DRAINING` se o serviço contiver tarefas usando o tipo de inicialização EC2. Esse parâmetro permite que você defina o tamanho dos lotes de implantação. Por exemplo, se o serviço tiver um número desejado de 4 tarefas e um valor máximo de porcentagem de 200%, o programador poderá iniciar 4 tarefas novas antes de interromper as 4 tarefas mais antigas. Isso é fornecido desde que os recursos de cluster necessários para isso estejam disponíveis. O valor padrão para a porcentagem máxima é 200%.

Se um serviço estiver usando o tipo de implantação azul/verde (`CODE_DEPLOY`) e as tarefas que usam o tipo de inicialização EC2, os valores de porcentagem de integridade mínima e porcentagem máxima serão definidos para os valores padrão. Elas são usadas apenas para definir os limites inferior e superior do número de tarefas no serviço que permanecem no `RUNNING` estado enquanto as instâncias de contêiner estão no `DRAINING` estado. Se as tarefas no serviço usarem o tipo de inicialização Fargate, os valores de porcentagem de integridade mínima e de porcentagem máxima não serão usados. Atualmente, eles ficam visíveis ao descrever o seu serviço.

Quando o programador de serviço substitui uma tarefa durante uma atualização, o serviço primeiro eliminará a tarefa do load balancer (se usado) e esperará as conexões se dissiparem. Em seguida, o equivalente de docker stop será enviado para os contêineres executados na tarefa. Isso resulta em um sinal `SIGTERM` e um tempo limite de 30 segundos, após o qual `SIGKILL` é enviado, e os contêineres são interrompidos à força. Se o contêiner administra o sinal `SIGTERM` com tranquilidade e sai dentro de 30 segundos após recebê-lo, nenhum sinal `SIGKILL` é enviado. O programador de serviço inicia e interrompe tarefas conforme definidas por suas configurações de porcentagem íntegras mínimas e máximas.

> **Important**
>
> Se você alterar as portas usadas por contêineres em uma definição de tarefa, talvez seja necessário atualizar os security groups da instância de contêiner para que funcionem com as portas atualizadas.
> Caso o serviço use um load balancer, a configuração do load balancer definido para o serviço quando criado não pode ser alterada. Se você atualizar a definição de tarefa para o serviço, o contêiner e a porta de contêiner especificados quando o serviço foi criado deverão permanecer na definição da tarefa.
> Para alterar o load balancer, o nome do contêiner, ou a porta do contêiner associada a uma configuração do load balancer de serviço, você deve criar um novo serviço.
> O Amazon ECS não dá atualiza automaticamente os grupos de segurança associados a load balancers do Elastic Load Balancing ou instâncias de contêiner do Amazon ECS.

Para atualizar um serviço em execução

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a Região em que seu cluster está localizado.
3. No painel de navegação, escolha Clusters.
4. Na página Clusters, selecione o nome do cluster no qual seu serviço reside.
5. Na página Cluster: ***name***, escolha Services.
6. Marque a caixa à esquerda do serviço para atualizar e escolha Update.

7. Na página Configure service, as informações do serviço são preenchidas antecipadamente. Altere a definição de tarefa, a versão da plataforma, a configuração de implantação ou o número de tarefas desejadas (ou qualquer combinação desses itens) e escolha Next step (Próxima etapa).

   Note

   Para que seu serviço use uma imagem de docker recém-atualizada com a mesma tag da definição de tarefa existente (por exemplo, `my_image:latest`) ou mantenha as configurações atuais do seu serviço, selecione Force new deployment (Forçar nova implantação). As novas tarefas executadas pela implantação obtêm a combinação atual de imagem/tag do seu repositório ao iniciarem. A opção Force new deployment (Forçar nova implantação) também é usada ao atualizar uma tarefa Fargate para usar uma versão mais atual da plataforma ao especificar `LATEST`. Por exemplo, se você especificar `LATEST`, suas tarefas em execução estiverem usando a versão `1.0.0` da plataforma e quiser que elas sejam reiniciadas usando uma versão mais recente da plataforma.

8. Na página Configure deployments (Configurar implantações), se o seu serviço estiver usando o tipo de implantação azul/verde, os componentes de sua implantação de serviço serão pré-preenchidos. Confirme as configurações a seguir.

   a. Em Application name (Nome do aplicativo), escolha o aplicativo do CodeDeploy do qual o serviço faz parte.

   b. Em Deployment group name (Nome do grupo de implantação), escolha o grupo de implantação do CodeDeploy do qual o serviço faz parte.

   c. Selecione os ganchos de evento do ciclo de vida da implantação e as funções do Lambda associadas para executar como parte da nova revisão da implantação do serviço. Os ganchos de ciclo de vida disponíveis são:

      - BeforeInstall – Use este gancho de evento do ciclo de vida da implantação para invocar uma função do Lambda antes da criação do conjunto de tarefas de substituição. O resultado da função Lambda nesse evento de ciclo de vida não aciona uma reversão.
      - AfterInstall – Use este gancho de evento do ciclo de vida da implantação para invocar uma função do Lambda após a criação do conjunto de tarefas de substituição. O resultado da função Lambda nesse evento de ciclo de vida pode acionar uma reversão.
      - BeforeAllowTraffic – Use esse gancho de evento de ciclo de vida de implantação para invocar uma função do Lambda antes que o tráfego de produção seja reencaminhado para o conjunto de tarefas de substituição. O resultado da função Lambda nesse evento de ciclo de vida pode acionar uma reversão.
      - AfterAllowTraffic – Use esse gancho de evento de ciclo de vida de implantação para invocar uma função do Lambda após o tráfego de produção ser reencaminhado para o conjunto de tarefas de substituição. O resultado da função Lambda nesse evento de ciclo de vida pode acionar uma reversão.

      Para obter mais informações sobre os ganchos de eventos de ciclo de vida, consulte Seção 'hooks' de AppSpec no AWS CodeDeploy User Guide.

9. Escolha Próxima etapa.

10. Na página Configure network, as informações da rede são preenchidas antecipadamente. Na seção Load balancing (Balanceamento de carga), se o serviço estiver usando o tipo de implantação azul/verde, selecione os listeners a serem associados aos grupos de destino. Altere o período de carência de verificação da integridade (se desejado) e escolha Next step.

11. (Opcional) Você pode usar o Serviço Auto Scaling para aumentar ou reduzir seu serviço automaticamente em resposta a alarmes do CloudWatch.

    a. Em Optional configurations (Configurações opcionais), escolha Configure Serviço Auto Scaling (Configurar).

    b. Vá para Etapa 5: (opcional) configurar o serviço para usar o Serviço Auto Scaling (p. 157).

c. Execute as etapas na seção e, em seguida, retorne.

12. Escolha Update Service para terminar e atualizar o seu serviço.

Excluir um serviço

É possível excluir um serviço do Amazon ECS usando o console. Antes da exclusão, o serviço é automaticamente reduzido a zero. Se você tiver um load balancer ou recursos de descoberta de serviço associados ao serviço, eles não são afetadas pela exclusão do serviço. Para excluir os recursos do Elastic Load Balancing, consulte um dos seguintes tópicos, de acordo com o tipo de load balancer: Excluir um Application Load Balancer ou Excluir um Network Load Balancer. Para excluir os recursos de descoberta de serviço, siga o procedimento abaixo.

Para excluir uma serviço do Amazon ECS

Use o procedimento a seguir para excluir um serviço do Amazon ECS.

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a Região em que seu cluster está localizado.
3. No painel de navegação, escolha Clusters e selecione o nome do cluster no qual o serviço reside.
4. Na página Cluster: ***name***, escolha Services.
5. Marque a caixa à esquerda do serviço para atualizar e escolha Delete.
6. Confirme a exclusão do serviço inserindo a frase de texto e escolha Delete (Excluir).

Para excluir os recursos do descoberta de serviço (AWS CLI)

Para excluir os recursos da descoberta de serviço restantes, você pode usar a AWS CLI para excluir o serviço da descoberta de serviço e o namespace da descoberta de serviço.

1. Certifique-se de que a versão mais recente da AWS CLI esteja instalada e configurada. Para obter mais informações sobre como instalar ou fazer o upgrade do seu AWS CLI, consulte Instalar a interface de linha do comando da AWS.
2. Recupere o ID do serviço de descoberta de serviço para exclusão.

```
aws servicediscovery list-services --region <region_name>
```

   Note

   Se nenhum serviço de descoberta de serviço for retornado, prossiga para a etapa 4.

3. Usando o ID do serviço de descoberta de serviço da saída anterior, exclua o serviço.

```
aws servicediscovery delete-service --id <service_discovery_service_id> --
region <region_name>
```

4. Recupere o ID do namespace de descoberta de serviço para exclusão.

```
aws servicediscovery list-namespaces --region <region_name>
```

5. Usando o ID do namespace de descoberta de serviço da saída anterior, exclua o namespace.

```
aws servicediscovery delete-namespace --id <service_discovery_namespace_id> --
region <region_name>
```

Lógica de controle de serviço

O programador de serviço do Amazon ECS inclui uma lógica que regula a frequência com que as tarefas de serviço são inicializadas caso elas falhem repetidamente ao tentar iniciar.

Se as tarefas de um serviço do ECS falharem repetidamente ao tentar entrar no estado `RUNNING` (mudando diretamente do estado `PENDING` para `STOPPED`), o tempo entre as tentativas subsequentes aumentará incrementalmente até chegar, no máximo, a 15 minutos. Esse período máximo está sujeito a alterações no futuro e não deve ser considerado permanente. Esse comportamento reduz o efeito que tarefas que não podem ser iniciadas têm sobre os recursos de cluster do Amazon ECS ou os custos de infraestrutura do Fargate. Se o seu serviço acionar a lógica de controle, você receberá a seguinte mensagem de evento do serviço (p. 313):

```
(service service-name) is unable to consistently start tasks successfully.
```

O Amazon ECS nunca impede novas tentativas de um serviço com falha nem tenta modificá-lo de outra forma que não seja aumentando o tempo entre as reinicializações. A lógica de controle de serviço não fornece parâmetros ajustáveis pelo usuário.

Se você atualizar o serviço para usar uma nova definição de tarefa, ele retornará imediatamente ao estado normal, não controlado. Para obter mais informações, consulte Atualizar um serviço (p. 160).

Veja a seguir algumas causas comuns que acionam essa lógica:

- O agente de contêiner do Amazon ECS não consegue extrair a imagem do Docker da tarefa. Isso pode ser devido a problemas com a imagem, tag ou nome da imagem do contêiner ou a falta de autenticação ou de permissões de registro privado. Nesse caso, você também verá um `CannotPullContainerError` entre os erros da tarefa interrompida (p. 310).

  Important

  Tarefas que são interrompidas depois que alcançam o estado `RUNNING` não acionam a lógica de controle ou a mensagem de evento do serviço associada. Por exemplo, se as verificações de integridade do Elastic Load Balancing que apresentam falha em um serviço resultam na sinalização da tarefa como não íntegra e no cancelamento do registro e eliminação da tarefa pelo Amazon ECS, isso não aciona o controle. Mesmo que um comando de contêiner da tarefa seja encerrado imediatamente com um código de saída diferente de zero, o estado da tarefa já terá sido mudado para `RUNNING`. Tarefas que falham imediatamente devido a erros de comando não acionam o controle ou a mensagem de evento do serviço.

Recursos e tags

Os recursos do Amazon ECS, incluindo definições de tarefa, clusters, tarefas, serviços e instâncias de contêiner, recebem um nome de recurso da Amazon (ARN) e um identificador de recurso (ID) exclusivo. Esses recursos podem ser marcados com valores que você define, para ajudá-lo a organizá-los e identificá-los.

Os seguintes tópicos descrevem recursos e tags e como você pode trabalhar com eles.

Tópicos

- Marcação dos seus recursos do Amazon ECS (p. 165)
- Relatórios de uso do Amazon ECS (p. 170)

Marcação dos seus recursos do Amazon ECS

Para ajudá-lo a gerenciar tarefas, serviços, definições de tarefas, clusters e instâncias de contêiner do Amazon ECS, é possível atribuir seus próprios metadados a cada recurso na forma de tags. Este tópico descreve tags e mostra a você como criá-los.

> **Important**
>
> Para usar esse recurso, é necessário que você aceite os novos formatos do nome de recurso da Amazon (ARN) e do identificador de recurso (ID). Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).

Tópicos

- Conceitos básicos de tags (p. 165)
- Marcação dos seus recursos (p. 166)
- Restrições de tag (p. 167)
- Marcação dos seus recursos para faturamento (p. 167)
- Trabalho com tags usando o console (p. 167)
- Trabalho com tags usando a CLI ou a API (p. 168)

Conceitos básicos de tags

Uma tag é um rótulo atribuído a um recurso da AWS. Cada tag consiste de uma chave e um valor opcional, ambos definidos por você.

As tags permitem categorizar seus recursos da AWS de diferentes formas (como por finalidade, por proprietário ou por ambiente). Isso é útil quando você tem muitos recursos do mesmo tipo; é possível identificar rapidamente um recurso específico baseado nas tags que você atribuiu a ele. Por exemplo, você pode definir um conjunto de tags para as instâncias de contêiner do Amazon ECS da sua conta que lhe ajudem a rastrear o proprietário e o nível da pilha de cada instância de contêiner.

Recomendamos que você desenvolva um conjunto de chave de tags que atenda suas necessidades para cada tipo de recurso. Usar um conjunto consistente de chaves de tags facilita para você gerenciar seus recursos. Você pode pesquisar e filtrar os recursos de acordo com as tags que adicionar.

As tags não têm significado semântico no Amazon ECS e são interpretadas estritamente como uma sequência dos caracteres. Além disso, as tags não são automaticamente atribuídas aos seus recursos. Você pode editar chaves de tags e valores, e você pode remover as tags de um recurso a qualquer

momento. Você pode definir o valor de uma tag a uma string vazia, mas não pode configurar o valor de um tag como nula. Se você adicionar uma tag que tenha a mesma chave de uma tag existente nesse recurso, o novo valor substituirá o antigo. Se você excluir um recurso, todas as tags do recurso também serão excluídas.

Você pode trabalhar com tags usando o Console de gerenciamento da AWS, a AWS CLI e a API do Amazon ECS.

Se você estiver usando o AWS Identity and Access Management (IAM), pode controlar quais usuários na sua conta da AWS têm permissão para criar, editar ou excluir tags.

Marcação dos seus recursos

Você pode marcar tarefas, serviços, definições de tarefa e clusters novos ou existentes do Amazon ECS.

Se você estiver usando o console do Amazon ECS, poderá aplicar tags a novos recursos quando eles forem criados ou a recursos existentes usando a guia Tags na página de recursos relevante a qualquer momento. A opção Propagate tags from (Propagar tags de) pode ser usada ao executar uma tarefa para copiar as tags da definição de tarefa para a tarefa ou ao criar um serviço para copiar as tags do serviço ou da definição de tarefa para as tarefas no serviço.

Se você estiver usando a API do Amazon ECS, a AWS CLI ou um SDK da AWS, poderá aplicar tags a novos recursos usando o parâmetro `tags` na ação da API relevante ou usar a ação da API `TagResource` para aplicar tags a recursos existentes. Para obter mais informações, consulte TagResource. O parâmetro propagateTags pode ser usado ao executar uma tarefa para copiar as tags da definição de tarefa para a tarefa ou ao criar um serviço para copiar as tags do serviço ou da definição de tarefa para as tarefas no serviço. Para obter mais informações, consulte RunTask e CreateService.

Além disso, algumas ações de criação de recursos permitem que você especifique tags para um recurso quando ele é criado. Se as tags não puderem ser aplicadas durante a criação dos recursos, nós reverteremos o processo de criação de recursos. Isso garante que os recursos sejam criados com tags ou, então, não criados, e que nenhum recurso seja deixado sem tags. Ao marcar com tags os recursos no momento da criação, você elimina a necessidade de executar scripts personalizados de uso de tags após a criação do recurso.

A tabela a seguir descreve os recursos do Amazon ECS que podem ser marcados com tags e os recursos que podem ser marcados na criação.

Suporte à marcação para os recursos do Amazon ECS

Recurso	Compatível com tags	Oferece suporte à propagação de tags	Compatível com a marcação na criação (API do Amazon ECS, AWS CLI, SDK da AWS)
Tarefas do Amazon ECS	Sim	Sim, a partir da definição de tarefa.	Sim
Serviços da Amazon ECS	Sim	Sim, da definição de tarefa ou do serviço para as tarefas no serviço.	Sim
Definições de tarefa do Amazon ECS	Sim	Não	Sim
Clusters do Amazon ECS	Sim	Não	Sim

Restrições de tag

As restrições básicas a seguir se aplicam às tags:

- Número máximo de tags por recurso – 50
- Em todos os recursos, cada chave de tag deve ser exclusiva e pode ter apenas um valor.
- Comprimento máximo da chave – 128 caracteres Unicode em UTF-8
- Valor máximo da chave: 256 caracteres Unicode em UTF-8
- Se seu esquema de tags é usado em vários serviços e recursos, lembre-se de que outros serviços podem ter restrições nos caracteres permitidos. Em geral, os caracteres permitidos são: letras, números e espaços representáveis em UTF-8 e os seguintes caracteres: + - = . _ : / @.
- As chaves e os valores de tags diferenciam maiúsculas de minúsculas.
- Não use `aws:`, `AWS:` nem qualquer combinação de letras maiúsculas e minúsculas deles como um prefixo para chaves ou valores, pois são reservados para uso pela AWS. Você não pode editar nem excluir chaves nem valores de tag com esse prefixo. As tags com esse prefixo não contam para as tags por limite de recurso.

Marcação dos seus recursos para faturamento

Ao ativar as tags gerenciadas do Amazon ECS, o Amazon ECS marcará automaticamente todas as tarefas recém-iniciadas com o nome do cluster. Para tarefas que pertencem a um serviço, elas também serão marcadas com o nome do serviço. Essas tags gerenciadas são úteis ao analisar a alocação de custos depois de ativá-las em seu Relatório de custo e uso. Para obter mais informações, consulte Relatórios de uso do Amazon ECS (p. 170).

Para ver o custo dos recursos combinados, você pode organizar as informações de faturamento com base nos recursos com os mesmos valores da chave da tag. Por exemplo, você pode etiquetar vários recursos com um nome de aplicação específico, e depois organizar suas informações de faturamento para ver o custo total daquela aplicação em vários serviços. Para obter mais informações sobre como configurar um relatório de alocação de custos com tags, consulte Relatório de alocação de custos mensal no Guia do usuário do AWS Billing and Cost Management.

> Important
>
> Para usar esse recurso, é necessário que você aceite os novos formatos do nome de recurso da Amazon (ARN) e do identificador de recurso (ID). Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).

> Note
>
> Se você tiver acabado de habilitar a criação de relatórios, os dados do mês atual estarão disponíveis para visualização após 24 horas.

Trabalho com tags usando o console

Usando o console do Amazon ECS, você pode gerenciar as tags associadas a tarefas, serviços, definições de tarefas, clusters ou instâncias de contêiner novas ou existentes.

Quando você selecionar uma página específica do recurso no console do Amazon ECS, ela exibirá uma lista desses recursos. Por exemplo, se você selecionar Clusters no painel de navegação, o console exibirá uma lista dos clusters do Amazon ECS. Ao selecionar um recurso de uma dessas listas (por exemplo, um cluster específico), se o recurso for compatível com tags, você pode vê-las e gerenciá-las na guia Tags.

Tópicos

- Adição de tags em um recurso individual ao iniciar (p. 168)

- Adição e exclusão de tags em um recurso individual (p. 168)

Adição de tags em um recurso individual ao iniciar

Os recursos a seguir permitem que você especifique tags ao criar o recurso.

Tarefa	Console
Execute uma ou mais tarefas.	Tarefas em execução (p. 91)
Crie um serviço.	Criar um serviço (p. 148)
Registre uma definição de tarefa.	Como criar uma definição de tarefa (p. 29)
Crie um cluster.	Criação de um cluster (p. 22)

Adição e exclusão de tags em um recurso individual

O Amazon ECS permite adicionar ou excluir tags associadas aos clusters, serviços, tarefas e definições de tarefas diretamente na página do recurso.

Para adicionar uma tag a um recurso individual

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região a ser usada.
3. No painel de navegação, selecione um tipo de recurso (por exemplo, Clusters).
4. Selecione o recurso da lista de recursos e selecione Tags, Edit (Editar).
5. Na caixa de diálogo Edit Tags (Editar tags), especifique a chave e o valor de cada tag e selecione Save (Salvar).

Para excluir uma tag de um recurso individual

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região a ser usada.
3. No painel de navegação, escolha um tipo de recurso (por exemplo, Clusters).
4. Selecione o recurso da lista de recursos e selecione Tags, Edit (Editar).
5. Na página Edit Tags (Editar tags), selecione o ícone Delete (Excluir) para cada tag que você deseja excluir e escolha Save (Salvar).

Trabalho com tags usando a CLI ou a API

Use o seguinte para adicionar, atualizar, listar e excluir as tags para seus recursos. A documentação correspondente traz exemplos.

Suporte à marcação para os recursos do Amazon ECS

Tarefa	CLI da AWS	Ação API
Adicione ou sobrescreva uma ou mais tags.	tag-resource	TagResource

Tarefa	CLI da AWS	Ação API
Exclua uma ou mais tags.	untag-resource	UntagResource

Os exemplos a seguir mostram como marcar ou desmarcar recursos usando a AWS CLI.

Exemplo 1: marcar um cluster existente

O comando a seguir marca um cluster existente.

```
aws ecs tag-resource --resource-arn resource_ARN --tags key=stack,value=dev
```

Exemplo 2: desmarcar um cluster existente

O comando a seguir exclui uma tag de um cluster existente.

```
aws ecs untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Exemplo 3: listar tags de um recurso

O comando a seguir lista as tags associadas a um recurso existente.

```
aws ecs list-tags-for-resource --resource-arn resource_ARN
```

Algumas ações de criação de recursos permitem especificar as tags ao criar o recurso. As ações a seguir são compatíveis com o uso de tags na criação.

Tarefa	AWS CLI	AWS Tools para Windows PowerShell	Ação API
Execute uma ou mais tarefas.	run-task	Start-ECSTask	RunTask
Crie um serviço.	create-service	New-ECSService	CreateService
Registre uma definição de tarefa.	register-task-definition	Register-ECSTaskDefinition	RegisterTaskDefinition
Crie um cluster.	create-cluster	New-ECSCluster	CreateCluster

Os exemplos a seguir demonstram como aplicar tags ao criar recursos.

Exemplo 1: criar um cluster e aplicar uma tag

O comando a seguir cria um cluster chamado `devcluster` e adiciona uma tag com a chave `team` e o valor `devs`.

```
aws ecs create-cluster --cluster-name devcluster --tags key=team,value=devs
```

Exemplo 2: criar um serviço e aplicar uma tag

O comando a seguir cria um serviço chamado `application` e adiciona uma tag com a chave `stack` e o valor `dev`.

```
aws ecs create-service --service-name application --task-definition task-def-app --tags
 key=stack,value=dev
```

Exemplo 3: criar um serviço com tags e propagar as tags para as tarefas no serviço

O parâmetro `--propagateTags` pode ser usado para copiar as tags de uma definição de tarefa ou de um serviço para as tarefas em um serviço. O comando a seguir cria um serviço com tags e as propaga para as tarefas nesse serviço.

```
aws ecs create-service --service-name application --task-definition task-def-app --tags
 key=stack,value=dev --propagateTags Service
```

Relatórios de uso do Amazon ECS

A AWS fornece uma ferramenta de geração de relatório gratuita, chamada Cost Explorer, que permite analisar o custo e o uso dos recursos do Amazon ECS.

O Cost Explorer é uma ferramenta gratuita que você pode usar para exibir gráficos de uso e custos. É possível visualizar dados dos últimos 13 meses e prever o provável valor que você gastará nos próximos três meses. É possível usar o Cost Explorer para ver padrões de gastos de recursos da AWS ao longo do tempo, identificar áreas que precisam de uma investigação mais profunda e ver tendências que você pode usar para entender seus custos. Também é possível especificar os períodos dos dados e visualizar os dados de tempo por dia ou mês.

Os dados de medição no seu Relatório de custo e uso mostram o uso em todas as suas tarefas do Amazon ECS. Os dados de medição incluem `vCPU-Hours` e `GB-Hours` de memória para cada tarefa que foi executada. A forma como os dados são apresentados depende do tipo de execução da tarefa, conforme descrito abaixo.

Para tarefas que usam o tipo de execução Fargate, você também verá o custo associado às suas tarefas.

Você também pode usar as tags gerenciadas do Amazon ECS para identificar o serviço ou o cluster ao qual cada tarefa pertence. Para obter mais informações, consulte Marcação dos seus recursos para faturamento (p. 167).

> **Important**
>
> Os dados de medição só podem ser visualizados para tarefas iniciadas em 16 de novembro de 2018 ou após essa data. Tarefas executadas antes desta data não mostrarão dados de medição.

Veja um exemplo de alguns campos para os quais você pode classificar dados de alocação de custos ao usar o Cost Explorer:

- Nome do cluster
- Nome do serviço
- Tags de recursos
- Tipo de inicialização
- Região
- Tipo de uso

Para obter mais informações sobre como criar um Relatório de custo e uso do AWS, consulte Relatório de custo e uso da AWS no Guia do usuário do AWS Billing and Cost Management.

Monitoramento do Amazon ECS

Você pode monitorar os recursos do Amazon ECS usando o Amazon CloudWatch, que coleta e processa dados brutos do Amazon ECS em métricas legíveis, quase em tempo real. Essas estatísticas são registradas por um período de duas semanas para que você possa acessar informações históricas e obter uma perspectiva melhor sobre o desempenho dos clusters ou serviços. Os dados de métricas do Amazon ECS são enviados automaticamente para o CloudWatch em períodos de 1 minuto. Para obter mais informações sobre o CloudWatch, consulte o Guia do usuário do Amazon CloudWatch.

O monitoramento é uma parte importante da manutenção da confiabilidade, da disponibilidade e do desempenho do Amazon ECS e das suas soluções da AWS. Você deve coletar dados de monitoramento de todas as partes de sua solução da AWS, para facilitar a depuração de uma falha multipontos, caso ocorra. Porém, para começar a monitorar o Amazon ECS, é necessário criar um plano de monitoramento que inclua respostas às seguintes perguntas:

- Quais são seus objetivos de monitoramento?
- Quais recursos você vai monitorar?
- Com que frequência você vai monitorar esses recursos?
- Quais ferramentas de monitoramento você usará?
- Quem realizará o monitoramento das tarefas?
- Quem deve ser notificado quando algo der errado?

Quando você usa o tipo de inicialização Fargate, obtém métricas de utilização da CPU e da memória para cada um dos seus serviços para auxiliar no monitoramento do seu ambiente.

A próxima etapa é estabelecer uma linha de base para desempenho do Amazon ECS normal no ambiente medindo o desempenho em vários momentos e em diferentes condições de carga. À medida que você monitora o Amazon ECS, armazene dados de monitoramento históricos para compará-los com os dados de performance atuais, identificar padrões de performance normais e anomalias de performance e elaborar métodos para resolver problemas.

Tópicos

- Ferramentas de monitoramento (p. 171)
- Amazon ECS CloudWatch Métricas (p. 173)
- Streaming de eventos do Amazon ECS para o Eventos do CloudWatch (p. 179)
- CloudWatch Container Insights do Amazon ECS (p. 186)
- Registro em log de chamadas à API do Amazon ECS com o AWS CloudTrail (p. 188)

Ferramentas de monitoramento

A AWS fornece várias ferramentas que você pode usar para monitorar o Amazon ECS. É possível configurar algumas dessas ferramentas para fazer o monitoramento em seu lugar, e, ao mesmo tempo, algumas das ferramentas exigem intervenção manual. Recomendamos que as tarefas de monitoramento sejam automatizadas ao máximo possível.

Ferramentas de monitoramento automatizadas

Você pode usar as seguintes ferramentas de monitoramento automatizadas para observar o Amazon ECS e gerar relatórios quando algo estiver errado:

- Alarmes do Amazon CloudWatch – observe uma única métrica ao longo de um período que você especificar e realize uma ou mais ações com base no valor da métrica em relação a um determinado limite ao longo de vários períodos. A ação é uma notificação enviada a um tópico do Amazon Simple Notification Service (Amazon SNS) ou a uma política do Amazon EC2 Auto Scaling. Os alarmes do CloudWatch não invocam ações simplesmente porque estão em um estado específico. O estado deve ter sido alterado e mantido por um número específico de períodos. Para obter mais informações, consulte Amazon ECS CloudWatch Métricas (p. 173).

  Para serviços com tarefas que usam o tipo de inicialização Fargate, você pode usar os alarmes do CloudWatch para expandir e reduzir as tarefas no serviço com base nas métricas do CloudWatch, como CPU e utilização da memória. Para obter mais informações, consulte Serviço Auto Scaling (p. 129).
- Amazon CloudWatch Logs – Monitore, armazene e acesse os arquivos de log nos contêineres nas tarefas do Amazon ECS especificando o driver de log `awslogs` nas definições de tarefa. Esse é o único método compatível com o acesso a logs para tarefas que usem o tipo de inicialização Fargate Para obter mais informações, consulte Como usar o driver de log awslogs (p. 64).
- Eventos do Amazon CloudWatch – Faça correspondência de eventos e direcione-os a uma ou mais funções ou fluxos de destino para fazer alterações, capturar informações de estado e realizar ações corretivas. Para obter mais informações, consulte Streaming de eventos do Amazon ECS para o Eventos do CloudWatch (p. 179) neste guia e O que é o Amazon CloudWatch Events? no Guia do usuário do Eventos do Amazon CloudWatch.
- Monitoramento do log do AWS CloudTrail – Compartilhe arquivos de log entre contas, monitore os arquivos de log do CloudTrail em tempo real enviando-os para o CloudWatch Logs, grave aplicativos de processamento de logs em Java e confirme se os arquivos de log não foram alterados após a distribuição pelo CloudTrail. Para obter mais informações, consulte Registro em log de chamadas à API do Amazon ECS com o AWS CloudTrail (p. 188) neste guia e Trabalhar com arquivos de log do CloudTrail no AWS CloudTrail User Guide.

Ferramentas de monitoramento manual

Outra parte importante do monitoramento do Amazon ECS envolve o monitoramento manual desses itens que os alarmes do CloudWatch não abrangem. Os painéis dos consoles do CloudWatch, Trusted Advisor e outros da AWS fornecem uma visão rápida do estado do ambiente da AWS. Recomendamos que você também verifique os arquivos de log nas instâncias de contêiner e os contêineres nas tarefas.

- Página inicial do CloudWatch:
  - Alertas e status atual
  - Gráficos de alertas e recursos
  - Estado de integridade do serviço

  Além disso, você pode usar o CloudWatch para fazer o seguinte:
  - Criar painéis personalizados para monitorar os serviços com os quais você se preocupa.
  - Colocar em gráfico dados de métrica para solucionar problemas e descobrir tendências.
  - Pesquise e procure todas as métricas de recursos da AWS.
  - Criar e editar alertas para ser notificado sobre problemas.
- O AWS Trusted Advisor pode ajudar a monitorar os recursos da AWSpara melhorar a performance, a confiabilidade, a segurança e a economia. Quatro verificações do Trusted Advisor estão disponíveis para todos os usuários. Mais de 50 verificações estão disponíveis para os usuários que têm um plano de suporte Business ou empresarial. Para obter mais informações, consulte AWS Trusted Advisor.

Amazon ECS CloudWatch Métricas

Você pode monitorar os recursos do Amazon ECS usando o Amazon CloudWatch, que coleta e processa dados brutos do Amazon ECS em métricas legíveis, quase em tempo real. Essas estatísticas são registradas por um período de duas semanas para que você possa acessar informações históricas e obter uma perspectiva melhor sobre o desempenho dos clusters ou serviços. Os dados de métricas do Amazon ECS são enviados automaticamente para o CloudWatch em períodos de 1 minuto. Para obter mais informações sobre o CloudWatch, consulte o Guia do usuário do Amazon CloudWatch.

Tópicos

- Como habilitar as métricas do CloudWatch (p. 173)
- Métricas e dimensões disponíveis (p. 173)
- Utilização de serviço (p. 176)
- Contagem de tarefas RUNNING de serviço (p. 177)
- Visualizar métricas do Amazon ECS (p. 178)

Como habilitar as métricas do CloudWatch

Qualquer serviço do Amazon ECS que use o tipo de inicialização do Fargate será automaticamente habilitada para métricas de uso de CPU e de memória do CloudWatch. Portanto, você não precisa executar etapas manuais.

Métricas e dimensões disponíveis

As seções a seguir listam as métricas e as dimensões que o Amazon ECS envia para o Amazon CloudWatch.

Métricas do Amazon ECS

O Amazon ECS fornece métricas para você monitorar seus recursos. Você pode medir a reserva e a utilização de CPU e memória em seu cluster como um todo e a utilização de CPU e memória dos serviços em seus clusters. Para as cargas de trabalho de GPU, você pode medir a reserva de GPU no cluster.

As métricas disponibilizadas dependerão do tipo de inicialização das tarefas e dos serviços em seus clusters. Se você estiver usando o tipo de inicialização Fargate para seus serviços, serão fornecidas métricas de utilização de CPU e memória para auxiliar no monitoramento de seus serviços. Para o tipo de inicialização do EC2, você possuirá e precisará monitorar as instâncias do Amazon EC2 que formam a infraestrutura subjacente. Dessa forma, CPU, memória e reserva de GPU adicionais, bem como métricas adicionais de utilização e reserva de memória e CPU são disponibilizadas no nível de cluster, serviço e tarefa.

Amazon ECS envia as seguintes métricas para CloudWatch por minuto. Ao coletar as métricas, Amazon ECS coleta vários pontos de dados a cada minuto. Em seguida, agrega-os a um ponto de dados antes de enviar os dados para CloudWatch. Logo, em CloudWatch, uma contagem de amostras é na verdade o agregado de vários pontos de dados durante um minuto.

O namespace `AWS/ECS` inclui as métricas a seguir.

Métrica	Descrição
`CPUReservation`	O percentual de unidades de CPU reservadas ao executar tarefas no cluster.

Métrica	Descrição
	A reserva de CPU no cluster (essa métrica só pode ser filtrada por `ClusterName`) é medida como o total de unidades de CPU reservadas por tarefas do Amazon ECS no cluster, dividido pelo total de unidades de CPU registradas para todas as instâncias de contêiner no cluster. Somente instâncias de contêiner no status `ACTIVE` ou `DRAINING` afetarão as métricas de reserva de CPU. Essa métrica é usada somente para tarefas usando o tipo de execução EC2 . Dimensões válidas: `ClusterName`. Estatísticas válidas: média, mínima, máxima, soma, contagem de exemplo. A estatística mais útil é Média. Unidade: percentual.
`CPUUtilization`	O percentual de unidades de CPU usadas no cluster ou serviço. A utilização de CPU no cluster (essa métrica só pode ser filtrada por `ClusterName` sem `ServiceName`) é medida como o total de unidades de CPU em uso por tarefas do Amazon ECS no cluster, dividido pelo total de unidades de CPU registradas para todas as instâncias de contêiner no cluster. Somente instâncias de contêiner no status `ACTIVE` ou `DRAINING` afetarão as métricas de utilização de CPU. As métricas de utilização de CPU do cluster são usadas somente para tarefas que usem o tipo de inicialização do EC2. A utilização da CPU de serviço (métricas que são filtradas por `ClusterName` e `ServiceName`) é medida como o total de unidades de CPU em uso pelas tarefas que pertencem ao serviço, dividido pelo número total de unidades de CPU reservadas para as tarefas que pertencem ao serviço. As métricas de utilização do CPU de serviço são usadas para tarefas usando o Fargate e o tipo de inicialização do EC2. Dimensões válidas: `ClusterName`, `ServiceName`. Estatísticas válidas: média, mínima, máxima, soma, contagem de exemplo. A estatística mais útil é Média. Unidade: percentual.

Métrica	Descrição
`MemoryReservation`	O percentual de memória reservada ao executar tarefas no cluster. A reserva de memória no cluster (essa métrica só pode ser filtrada por `ClusterName`) é medida como o total de memória reservada por tarefas do Amazon ECS no cluster, dividido pelo total de memória registrada para todas as instâncias de contêiner no cluster. Somente instâncias de contêiner no status `ACTIVE` ou `DRAINING` afetarão as métricas de reserva de memória de CPU. Essa métrica é usada somente para tarefas usando o tipo de inicialização do EC2. Dimensões válidas: `ClusterName`. Estatísticas válidas: média, mínima, máxima, soma, contagem de exemplo. A estatística mais útil é Média. Unidade: percentual.
`MemoryUtilization`	O percentual de memória usada no cluster ou serviço. A utilização de memória no cluster (essa métrica só pode ser filtrada por `ClusterName` sem `ServiceName`) é medida como o total de memória em uso por tarefas do Amazon ECS no cluster, dividido pelo total de memória registrada para todas as instâncias de contêiner no cluster. Somente instâncias de contêiner no status `ACTIVE` ou `DRAINING` afetarão as métricas de utilização de memória. As métricas de utilização de memória do cluster são usadas somente para tarefas que usem o tipo de inicialização do EC2. A utilização da memória de serviço (métricas que são filtradas por `ClusterName` e `ServiceName`) é medida como o total de memória em uso pelas tarefas que pertencem ao serviço, dividido pelo número total de memória reservada para as tarefas que pertencem ao serviço. As métricas de utilização da memória de serviço são usadas para tarefas usando os tipos de inicialização do Fargate e EC2. Dimensões válidas: `ClusterName`, `ServiceName`. Estatísticas válidas: média, mínima, máxima, soma, contagem de exemplo. A estatística mais útil é Média. Unidade: percentual.

Métrica	Descrição
`GPUReservation`	O percentual total de GPUs disponíveis reservadas ao executar tarefas no cluster. A reserva de GPU de cluster é medida pelo número de GPUs reservadas por tarefas do Amazon ECS no cluster, dividido pelo número total de GPUs disponíveis em todas as instâncias de contêiner habilitadas para GPU no cluster. Somente instâncias de contêiner no status `ACTIVE` ou `DRAINING` afetarão as métricas de reserva de GPU. Dimensões válidas: `ClusterName`. Estatísticas válidas: média, mínima, máxima, soma, contagem de exemplo. A estatística mais útil é Média. Unidade: percentual.

Note

Se você estiver usando tarefas com o tipo de inicialização do EC2 tiver instâncias de contêiner do Linux, o agente de contêiner do Amazon ECS contará com métricas `stats` do Docker para coletar dados de CPU e memória para cada contêiner em execução na instância. Para instâncias de desempenho com capacidade de intermitência (instâncias T3, T3a e T2), a métrica de utilização da CPU pode refletir dados diferentes em comparação com as métricas de CPU no nível de instância.

Dimensões para métricas do Amazon ECS

As métricas do Amazon ECS usam `AWS/ECS` namespace e fornecem métricas para as seguintes dimensões:

Dimensão	Descrição
`ClusterName`	Essa dimensão filtra os dados que você solicitar para todos os recursos em um cluster especificado. Todas as métricas do Amazon ECS são filtradas por `ClusterName`.
`ServiceName`	Essa dimensão filtra os dados que você solicitar para todos os recursos em um serviço especificado em um determinado cluster.

Utilização de serviço

A utilização de serviço é medida como a porcentagem de CPU e de memória utilizada pelas tarefas do Amazon ECS que pertencem a um serviço em um cluster em comparação com a CPU e a memória especificadas na definição de tarefa do serviço. Essa métrica oferece suporte somente a serviços com tarefas que usam o tipo de inicialização Fargate.

```
                    (Total CPU units used by tasks in service) x 100
```

```
Service CPU utilization =
 ------------------------------------------------------------------------------
                          (Total CPU units specified in task definition) x (number of
 tasks in service)
```

```
                                        (Total MiB of memory used by tasks in service) x
 100
Service memory utilization =
 ------------------------------------------------------------------------------
                            (Total MiB of memory specified in task definition) x (number
 of tasks in service)
```

A cada minuto, o agente de contêiner do Amazon ECS associado a cada tarefa calcula o número de unidades de CPU e de MiBs de memória em uso atualmente para cada tarefa pertencente ao serviço, e essas informações são relatadas ao Amazon ECS. A quantidade total de CPU e memória utilizada para todas as tarefas de propriedade do serviço em execução no cluster é calculada, e esses números são informados ao CloudWatch como uma porcentagem do total de recursos especificados para o serviço na definição de tarefa de serviço. Se você especificar um limite flexível (`memoryReservation`), ele será usado para calcular a quantidade de memória reservada. Caso contrário, o limite rígido (`memory`) será usado. Para obter mais informações sobre os limites rígidos e flexíveis, consulte Parâmetros de definição de tarefas.

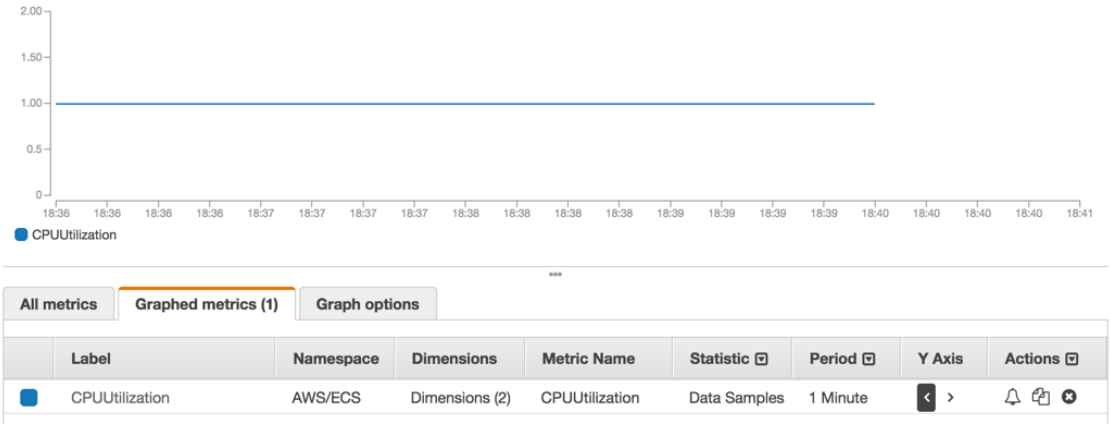
Contagem de tarefas `RUNNING` de serviço

Você pode usar as métricas do CloudWatch para saber o número de tarefas nos serviços que estão no estado `RUNNING`. Por exemplo, você pode definir um alarme do CloudWatch para essa métrica para alertá-lo se o número de tarefas em execução em seu serviço ficar abaixo de um valor especificado.

Para visualizar o número de tarefas em execução em um serviço

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, selecione Metrics (Métricas).
3. Na guia All metrics (Todas as métricas), escolha ECS.
4. Escolha ClusterName, ServiceName e selecione qualquer métrica (`CPUUtilization` ou `MemoryUtilization`) que corresponda ao serviço no qual deseja visualizar as tarefas em execução.
5. Na guia Graphed metrics (Métricas em gráfico), altere Period (Período) para 1 Minute (1 minuto) e a Statistic (Estatística) para Sample Count (Contagem de amostras).

   O valor exibido no gráfico indica o número de tarefas `RUNNING` no serviço.



Visualizar métricas do Amazon ECS

Depois de habilitar as métricas do CloudWatch para o Amazon ECS, você verá essas métricas nos consoles do Amazon ECS e do CloudWatch. O console do Amazon ECS fornece um máximo de 24 horas, no mínimo, e uma visualização média de seu métricas de serviço. O console do CloudWatch fornece uma exibição refinada e personalizável de seus recursos, bem como o número de tarefas em execução em um serviço.

Tópicos

- Como visualizar métricas de serviço no console do Amazon ECS (p. 178)
- Visualização das métricas do Amazon ECS no console do CloudWatch (p. 178)

Como visualizar métricas de serviço no console do Amazon ECS

As métricas de utilização de CPU e memória do serviço Amazon ECS estão disponíveis no console do Amazon ECS. A visualização fornecida para as métricas de serviço mostra os valores médios, mínimos e máximos referentes ao período de 24 horas anterior, com pontos de dados disponíveis em intervalos de 5 minutos. Para obter mais informações, consulte Utilização de serviço (p. 176).

Para visualizar métricas de serviço no console

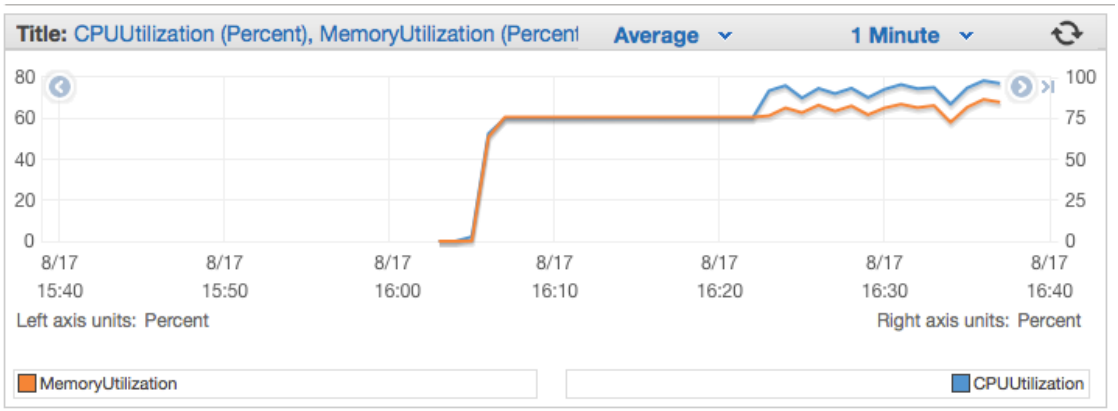
1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Selecione o cluster que contém o serviço do qual você deseja visualizar as métricas.
3. Na página Cluster: ***`cluster-name`***, escolha Services (Serviços).
4. Escolha o serviço do qual você deseja visualizar as métricas.
5. Na página Service: ***`service-name`***, selecione a guia Metrics (Métricas).

Visualização das métricas do Amazon ECS no console do CloudWatch

As métricas de serviço do Amazon ECS também podem ser visualizadas no console do CloudWatch. O console fornece a visualização mais detalhada das métricas do Amazon ECS, e você pode personalizar as visualizações para atender a suas necessidades. Você pode visualizar a Utilização de serviço (p. 176) e a Contagem de tarefas `RUNNING` de serviço (p. 177). Para obter mais informações sobre o CloudWatch, consulte o Guia do usuário do Amazon CloudWatch.

Para visualizar métricas no console do CloudWatch

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. Na seção Metrics (Métricas) do painel de navegação, escolha ECS.
3. Escolha as métricas a serem exibidas. As métricas de cluster são delimitadas como ECS > ClusterName e as métricas de utilização de serviço são delimitadas como ECS > ClusterName, ServiceName. O exemplo a seguir mostra a utilização de CPU e memória do cluster.



Streaming de eventos do Amazon ECS para o Eventos do CloudWatch

Você pode usar o streaming de eventos do Amazon ECS para o Eventos do CloudWatch receber notificações quase em tempo real quanto ao estado atual dos seus clusters do Amazon ECS. Ao usar o tipo de execução Fargate, você poderá ver o estado das tarefas.

Usando o Eventos do CloudWatch, você pode criar programadores personalizados além do Amazon ECS que são responsáveis por orquestrar tarefas entre clusters e monitorar o estado de clusters praticamente em tempo real. Você pode eliminar a programação e o monitoramento de código que consulta continuamente o serviço do Amazon ECS em busca de alterações feitas no status e, em vez disso, lidar com alterações feitas no estado do Amazon ECS de maneira assíncrona usando qualquer destino do Eventos do CloudWatch, como AWS Lambda, Amazon Simple Queue Service, Amazon Simple Notification Service e Amazon Kinesis Data Streams.

Um fluxo de eventos do Amazon ECS garante que cada evento seja entregue pelo menos uma vez. Se eventos duplicados forem enviados, o evento fornecerá informações suficientes para identificar as duplicatas. Para obter mais informações, consulte Como processar eventos (p. 181).

Os eventos são ordenados de maneira relativa, de maneira que você possa identificar facilmente quando um evento ocorreu em relação a outros eventos.

Tópicos

- Eventos do Amazon ECS (p. 179)
- Como processar eventos (p. 181)
- Tutorial: como escutar Eventos do CloudWatch do Amazon ECS (p. 183)
- Tutorial: como enviar alertas do Amazon Simple Notification Service para eventos de tarefa parada (p. 185)

Eventos do Amazon ECS

O Amazon ECS rastreia o estado de cada uma de suas tarefas. Se o estado de uma tarefa for alterado, um evento será acionado e enviado ao Eventos do CloudWatch. Esses eventos são classificados como eventos de alteração de estado de tarefa. Esses eventos e as causas possíveis serão descritos com mais detalhes nas seções a seguir.

Note

O Amazon ECS pode adicionar outros tipos de evento, origens e detalhes no futuro. Caso você esteja desserializando programaticamente dados JSON do evento, certifique-se de que o aplicativo esteja preparado para lidar com propriedades desconhecidas a fim de evitar problemas caso e quando essas propriedades adicionais sejam adicionadas.

Os eventos contêm dois campos `version`; um no corpo principal do evento e um no objeto `detail` objeto do evento.

- A versão no corpo principal do evento é definida como `0` em todos os eventos. Para obter mais informações sobre parâmetros do Eventos do CloudWatch, consulte Eventos e padrões de evento no Guia do usuário do Eventos do Amazon CloudWatch.
- A versão no objeto `detail` do evento descreve a versão do recurso associado. Sempre que um recurso muda de estado, essa versão é incrementada. Como os eventos podem ser enviados várias vezes, esse campo permite identificar eventos duplicados (eles terão a mesma versão no objeto `detail`). Se estiver replicando a o estado da tarefa com o Eventos do CloudWatch, você poderá comparar a versão de um recurso relatado pelas APIs do Amazon ECS com a versão relatada no Eventos do CloudWatch para o recurso (dentro do objeto `detail`) para verificar se a versão no fluxo de eventos é atual.

Tópicos

- Eventos de alteração feita no estado da tarefa (p. 180)

Eventos de alteração feita no estado da tarefa

Os seguintes cenários disparam eventos de alteração feita no estado da tarefa:

Você chama as operações da API `StartTask`, `RunTask` ou `StopTask` (diretamente ou com o Console de gerenciamento da AWS, a AWS CLI ou os SDKs).

A inicialização ou a parada de tarefas cria novos recursos de tarefa ou modifica o estado de recursos da tarefa existente.

O programador de serviços do Amazon ECS inicia ou para uma tarefa.

A inicialização ou a parada de tarefas cria novos recursos de tarefa ou modifica o estado de recursos da tarefa existente.

O agente de contêiner do Amazon ECS chama a operação da API `SubmitTaskStateChange`.

O agente de contêiner do Amazon ECS monitora o estado de suas tarefas e relata todas as alterações feitas de estado (por exemplo, de `PENDING` para `RUNNING` ou de `RUNNING` para `STOPPED`).

Um contêiner na tarefa muda de estado.

O agente de contêiner do Amazon ECS monitora o estado de contêineres dentro das tarefas. Por exemplo, caso um contêiner em execução dentro de uma tarefa pare, essa alteração no estado do contêiner dispara um evento.

Example Evento de alteração feita no estado da tarefa

Os eventos de alteração feitos no estado da tarefa são distribuídos no formato a seguir (a seção `detail` abaixo lembra o objeto Task retornado de uma operação de API DescribeTasks no Amazon Elastic Container Service API Reference). Para obter mais informações sobre parâmetros do Eventos do CloudWatch, consulte Eventos e padrões de evento no Guia do usuário do Eventos do Amazon CloudWatch.

```
{
  "version": "0",
  "id": "9bcdac79-b31f-4d3d-9410-fbd727c29fab",
  "detail-type": "ECS Task State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2016-12-06T16:41:06Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ecs:us-east-1:111122223333:task/b99d40b3-5176-4f71-9a52-9dbd6f1cebef"
  ],
  "detail": {
    "clusterArn": "arn:aws:ecs:us-east-1:111122223333:cluster/default",
    "containerInstanceArn": "arn:aws:ecs:us-east-1:111122223333:container-instance/
b54a2a04-046f-4331-9d74-3f6d7f6ca315",
    "containers": [
      {
        "containerArn": "arn:aws:ecs:us-east-1:111122223333:container/3305bea1-
bd16-4217-803d-3e0482170a17",
        "exitCode": 0,
        "lastStatus": "STOPPED",
        "name": "xray",
        "taskArn": "arn:aws:ecs:us-east-1:111122223333:task/
b99d40b3-5176-4f71-9a52-9dbd6f1cebef"
      }
    ],
    "createdAt": "2016-12-06T16:41:05.702Z",
    "desiredStatus": "RUNNING",
    "group": "task-group",
    "lastStatus": "RUNNING",
    "overrides": {
      "containerOverrides": [
        {
          "name": "xray"
        }
      ]
    },
    "startedAt": "2016-12-06T16:41:06.8Z",
    "startedBy": "ecs-svc/9223370556150183303",
    "updatedAt": "2016-12-06T16:41:06.975Z",
    "taskArn": "arn:aws:ecs:us-east-1:111122223333:task/
b99d40b3-5176-4f71-9a52-9dbd6f1cebef",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:111122223333:task-definition/xray:2",
    "version": 4
  }
}
```

Como processar eventos

O Amazon ECS envia eventos "pelo menos uma vez". Isso significa que você pode receber mais de uma única cópia de um determinado evento. Além disso, os eventos não podem ser distribuídos para os ouvintes de evento na ordem em que os eventos ocorreram.

Para habilitar a ordem apropriada de eventos, a seção `detail` de cada evento contém uma propriedade `version`. Os eventos com um número de propriedade da versão mais alto devem ser tratados como ocorridos depois de eventos com números da versão mais baixos. Os eventos com números de versão correspondentes podem ser tratados como cópias.

Exemplo: como processar eventos em uma função do AWS Lambda

O exemplo a seguir mostra uma função do Lambda escrita em Python 2.7 que captura os eventos de alteração de estado da tarefa e salva-os na seguinte tabela do Amazon DynamoDB:

- ECSTaskState: armazena o estado mais recente de uma tarefa. A ID da tabela é o valor `taskArn` da tarefa.

```
import json
import boto3

def lambda_handler(event, context):
    id_name = ""
    new_record = {}

    # For debugging so you can see raw event format.
    print('Here is the event:')
    print(json.dumps(event))

    if event["source"] != "aws.ecs":
       raise ValueError("Function only supports input from events with a source type of:
 aws.ecs")

    # Switch on task/container events.
    table_name = ""
    if event["detail-type"] == "ECS Task State Change":
        table_name = "ECSTaskState"
        id_name = "taskArn"
        event_id = event["detail"]["taskArn"]
    elif event["detail-type"] == "ECS Container Instance State Change":
        table_name = "ECSCtrInstanceState"
        id_name =  "containerInstanceArn"
        event_id = event["detail"]["containerInstanceArn"]
    else:
        raise ValueError("detail-type for event is not a supported type. Exiting without
 saving event.")

    new_record["cw_version"] = event["version"]
    new_record.update(event["detail"])

    # "status" is a reserved word in DDB, but it appears in containerPort
    # state change messages.
    if "status" in event:
        new_record["current_status"] = event["status"]
        new_record.pop("status")


    # Look first to see if you have received a newer version of an event ID.
    # If the version is OLDER than what you have on file, do not process it.
    # Otherwise, update the associated record with this latest information.
    print("Looking for recent event with same ID...")
    dynamodb = boto3.resource("dynamodb", region_name="us-east-1")
    table = dynamodb.Table(table_name)
    saved_event = table.get_item(
        Key={
            id_name : event_id
        }
    )
    if "Item" in saved_event:
        # Compare events and reconcile.
        print("EXISTING EVENT DETECTED: Id " + event_id + " - reconciling")
```

```
        if saved_event["Item"]["version"] < event["detail"]["version"]:
            print("Received event is a more recent version than the stored event -
updating")
            table.put_item(
                Item=new_record
            )
        else:
            print("Received event is an older version than the stored event - ignoring")
    else:
        print("Saving new event - ID " + event_id)

        table.put_item(
            Item=new_record
        )
```

Tutorial: como escutar Eventos do CloudWatch do Amazon ECS

Neste tutorial, você configura uma função do AWS Lambda simples que escuta eventos de tarefa do Amazon ECS e os grava em um fluxo de log do CloudWatch Logs.

Pré-requisito: configurar um cluster de teste

Caso você não tenha um cluster em execução para capturar eventos, siga as etapas em Conceitos básicos do Amazon ECS (p. 16) para criar um. Ao final deste tutorial, você executa uma tarefa nesse cluster para testar se configurou a função do Lambda corretamente.

Etapa 1: criar a função Lambda

Neste procedimento, você cria uma função simples do Lambda para funcionar como um destino para mensagens do fluxo de eventos do Amazon ECS.

1. Abra o console do AWS Lambda em https://console.aws.amazon.com/lambda/.
2. Escolha Create function.
3. Na tela Author from scratch, faça o seguinte:
   a. Em Name (Nome), insira um valor.
   b. Para Runtime, escolha Python 2.7.
   c. Em Role (Função), escolha Create a new role with basic Lambda permissions (Criar uma nova função com permissões básicas do Lambda).
4. Escolha Create function.
5. Na seção Function code, edite o código de exemplo de acordo com o exemplo a seguir:

```
import json

def lambda_handler(event, context):
    if event["source"] != "aws.ecs":
       raise ValueError("Function only supports input from events with a source type
 of: aws.ecs")

    print('Here is the event:')
    print(json.dumps(event))
```

   Essa é uma função simples do Python 2.7 que imprime o evento enviado pelo Amazon ECS. Se tudo estiver configurado corretamente, no final deste tutorial, você verá que os detalhes do evento aparecerão no stream de logs do CloudWatch Logs associado a essa função do Lambda.

6. Escolha Salvar.

Etapa 2: Registrar regra de evento

Em seguida, você cria uma regra de evento do Eventos do CloudWatch que captura eventos de tarefa vindos dos clusters do Amazon ECS. Esta regra captura todos os eventos vindos de todos os clusters dentro da conta em que está definida. As próprias mensagens de tarefa contêm informações sobre a origem do evento, inclusive o cluster no qual reside, que você pode usar para filtrar e classificar eventos programaticamente.

> Note
>
> Quando você usa Console de gerenciamento da AWS para criar uma regra de evento, o console adiciona automaticamente as permissões do IAM necessárias para conceder ao Eventos do CloudWatch permissão para chamar a função do Lambda. Caso esteja criando uma regra de evento usando a AWS CLI, você precisa conceder essa permissão explicitamente. Para obter mais informações, consulte Eventos e padrões de evento no Guia do usuário do Eventos do Amazon CloudWatch.

Para rotear eventos para sua função Lambda

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Events (Eventos), Rules (Regras), Create rule (Criar regra).
3. Em Event Source, escolha ECS como origem do evento. Por padrão, a regra se aplica a todos os eventos do Amazon ECS de todos os grupos do Amazon ECS. Como alternativa, você pode selecionar eventos específicos ou um grupo do Amazon ECS específico.
4. Em Targets (Alvos), escolha Add target (Adicionar alvo), em Target type (Tipo de alvo), escolha Lambda function (Função Lambda) e selecione a função do Lambda.
5. Escolha Configure details (Configurar detalhes).
6. Em Rule definition, digite um nome e uma descrição para a regra e escolha Create rule.

Etapa 3: Testar a regra

Por fim, você cria uma regra de evento do Eventos do CloudWatch que captura eventos de tarefa vindos dos clusters do Amazon ECS. Esta regra captura todos os eventos vindos de todos os clusters dentro da conta em que está definida. As próprias mensagens de tarefa contêm informações sobre a origem do evento, inclusive o cluster no qual reside, que você pode usar para filtrar e classificar eventos programaticamente.

Para testar sua regra

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Escolha Clusters, padrão.
3. Na tela Cluster: default (Cluster: padrão), escolha Tasks (Tarefas), Run new Task (Executar nova tarefa).
4. Em Task Definition (Definição da tarefa), selecione a versão mais recente de console-sample-app-static e escolha Run Task (Executar tarefa).
5. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
6. No painel de navegação, escolha Logs e selecione o grupo de logs para sua função do Lambda (por exemplo, /aws/lambda/*my-function*).
7. Selecione um fluxo de log para visualizar os dados do evento.

Tutorial: como enviar alertas do Amazon Simple Notification Service para eventos de tarefa parada

Neste tutorial, você configura uma regra de evento do Eventos do CloudWatch que só captura eventos de tarefa nos quais a tarefa deixou de ser executada porque um dos contêineres essenciais foi encerrado. O evento só envia eventos de tarefa com uma propriedade `stoppedReason` específica para o tópico do Amazon SNS designado.

Pré-requisito: configurar um cluster de teste

Caso você não tenha um cluster em execução para capturar eventos, siga as etapas em Conceitos básicos do Amazon ECS (p. 16) para criar um. Ao final deste tutorial, você executa uma tarefa nesse cluster para testar se configurou o tópico do Amazon SNS e a regra de evento do Eventos do CloudWatch corretamente.

Etapa 1: Criar e se inscrever em um tópico do Amazon SNS

Neste tutorial, você configura um tópico do Amazon SNS para funcionar como um destino de evento para a nova regra de evento.

Para criar um tópico do Amazon SNS

1. Abra o console do Amazon SNS em https://console.aws.amazon.com/sns/v3/home.
2. Escolha Topics, Create new topic.
3. Na janela Create new topic (Criar novo tópico), em Topic name (Nome do tópico), insira TaskStoppedAlert e escolha Create topic (Criar tópico).
4. Na janela Topics, selecione o tópico que você acabou de criar. Na tela Topic details: TaskStoppedAlert, escolha Create subscription.
5. Na janela Criar assinatura, em Protocolo, selecione Email. Em Endpoint, insira um endereço de e-mail ao qual você atualmente tem acesso e escolha Criar assinatura.
6. Verifique sua conta de e-mail e espere para receber uma mensagem de e-mail de confirmação de assinatura. Quando você recebê-la, escolha Confirmar assinatura.

Etapa 2: Registrar regra de evento

Em seguida, você registra uma regra de evento que captura apenas eventos de tarefa parada para tarefas com contêineres parados.

Para criar uma regra de evento

1. Abra o console do CloudWatch em https://console.aws.amazon.com/cloudwatch/.
2. No painel de navegação, escolha Events (Eventos), Create rule (Criar regra).
3. Escolha Mostrar opções avançadas, editar.
4. Em Criar um padrão que seleciona os eventos para processamento por seus destinos, substitua o texto existente com o seguinte texto:

```
{
"source": [
  "aws.ecs"
],
"detail-type": [
```

```
  "ECS Task State Change"
],
"detail": {
  "lastStatus": [
    "STOPPED"
  ],
  "stoppedReason" : [
    "Essential container in task exited"
  ]
}
}
```

Esse código define uma regra de evento do Eventos do CloudWatch que corresponda a qualquer evento no qual os campos `lastStatus` e `stoppedReason` correspondem aos valores indicados. Para obter mais informações sobre padrões de evento, consulte Eventos e padrões de eventos no Guia do usuário do Amazon CloudWatch.

5. Em Targets, escolha Add target. Em Target type (Tipo de alvo), escolha SNS topic (Tópico SNS) e, em seguida, escolha TaskStoppedAlert.
6. Escolha Configure details (Configurar detalhes).
7. Em Definição de regra, digite um nome e uma descrição para a regra e escolha Criar regra.

Etapa 3: Testar a regra

Para testar a regra, você tenta executar uma tarefa fechada logo depois da inicialização. Caso a regra de evento esteja configurada corretamente, você recebe uma mensagem de e-mail em alguns minutos com o texto do evento.

Para testar uma regra

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Escolha Task Definitions (Definições de tarefas), Create new Task Definition (Criar nova definição de tarefa).
3. Em Task Definition Name (Nome da definição de tarefa), digite WordPressFailure e escolha Add Container (Adicionar contêiner).
4. Em Container name, digite Wordpress, em Image, digite wordpress e em Maximum memory (MB), digite 128.
5. Escolha Add, Create.
6. Na tela Task Definition, escolha Actions, Run Task.
7. Em Clusters, escolha default (padrão). Escolha Run Task (Executar tarefa).
8. Na guia Tasks para seu cluster, periodicamente, escolha o ícone de atualização até que você deixe de ver sua tarefa em execução. Para verificar se sua tarefa parou, em Desired task status (Status desejado da tarefa), escolha Stopped (Interrompido) para verificar se a tarefa foi interrompida.
9. Verifique seu e-mail para confirmar que você recebeu um e-mail de alerta para a notificação de interrupção.

CloudWatch Container Insights do Amazon ECS

O CloudWatch Container Insights coleta, agrega e resume métricas e logs de seus aplicativos conteinerizados e microsserviços. As métricas incluem a utilização de recursos, como CPU, memória, disco e rede. As métricas de rede só estão disponíveis para tarefas que usam o modo de rede `bridge`. As métricas estão disponíveis em painéis automáticos do CloudWatch. Para obter uma lista completa de

métricas do Amazon ECS Container Insights, consulte Métricas do Amazon ECS Container Insights no Guia do usuário do Amazon CloudWatch.

Os dados operacionais são coletados como eventos do log de desempenho. Essas são entradas que usam um esquema JSON estruturado que permite dados de alta cardinalidade para serem ingeridos e armazenados em escala. Com base nesses dados, o CloudWatch cria métricas agregadas de alto nível no cluster e no nível de serviço como métricas do CloudWatch. Para obter mais informações, consulte Logs estruturados do Container Insights para Amazon ECS no Guia do usuário do Amazon CloudWatch.

> **Important**
>
> O CloudWatch Container Insights é fornecido a um custo adicional. Para obter informações sobre as métricas de monitoramento padrão que são fornecidas sem custo adicional, consulte Amazon ECS CloudWatch Métricas (p. 173).

Como trabalhar com clusters habilitados para Container Insights

O Container Insights pode ser habilitado para todos os novos clusters criados ao escolher a configuração da conta `containerInsights`, em clusters individuais, habilitando-o usando as configurações de cluster durante a criação do cluster ou em clusters existentes usando a API UpdateClusterSettings.

A escolha da configuração da conta do `containerInsights` pode ser feita com o console do Amazon ECS e a AWS CLI. Para obter mais informações sobre como criar clusters do Amazon ECS, consulte Criação de um cluster (p. 22).

> **Important**

Para aceitar os clusters habilitados para Container Insights para todos os usuários ou funções do IAM na sua conta usando o console

1. Como o usuário raiz da conta, abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação na parte superior da tela, selecione a região na qual deseja aceitar clusters habilitados para Container Insights.
3. No painel, escolha Account Settings (Configurações da conta).
4. Em IAM user or role (Usuário ou função do IAM), verifique se o usuário raiz ou a função do IAM de instância de contêiner está selecionada.
5. Em Container Insights, selecione a caixa de seleção. Escolha Save (Salvar) quando terminar.

   > **Important**
   >
   > Os usuários e as funções do IAM precisam da permissão `ecs:PutAccountSetting` para executar essa ação.
6. Na tela de confirmação, escolha Confirm (Confirmar) para salvar a seleção.

Para aceitar os clusters habilitados Container Insights para todos os usuários ou funções do IAM na sua conta usando a linha de comando

Qualquer usuário em uma conta pode usar um dos comando a seguir para modificar a configuração de conta padrão para todos os usuários ou funções do IAM na sua conta. Essas alterações se aplicarão a toda a conta da AWS, a menos que um usuário ou uma função do IAM cancele explicitamente essas configurações por conta própria.

- put-account-setting-default (AWS CLI)

```
aws ecs put-account-setting-default --name containerInsights --value enabled --region us-
east-1
```

- Write-ECSAccountSettingDefault (AWS Tools para Windows PowerShell)

```
Write-ECSAccountSettingDefault -Name containerInsights -Value enabled -Region us-east-1 -
Force
```

Para aceitar os clusters habilitados para Container Insights como usuário raiz para um usuário do IAM ou uma função do IAM de instância de contêiner usando a linha de comando

O usuário raiz em uma conta pode usar um dos seguintes comandos e especificar o ARN do usuário do IAM ou a função do IAM de instância de contêiner principal na solicitação para modificar as configurações da conta.

- put-account-setting (AWS CLI)

  O exemplo a seguir modifica a configuração da conta de um usuário do IAM específico:

```
aws ecs put-account-setting --name containerInsights --value enabled --principal-arn
 arn:aws:iam::aws_account_id:user/userName --region us-east-1
```

- Write-ECSAccountSetting (AWS Tools para Windows PowerShell)

  O exemplo a seguir modifica a configuração da conta de um usuário do IAM específico:

```
Write-ECSAccountSetting -Name containerInsights -Value enabled -PrincipalArn
 arn:aws:iam::aws_account_id:user/userName -Region us-east-1 -Force
```

Como atualizar as configurações de um cluster existente usando a linha de comando

Use um dos seguintes comandos para atualizar a configuração de um cluster.

- update-cluster-settings (AWS CLI)

```
aws ecs update-cluster-settings --cluster cluster_name_or_arn --settings
 name=containerInsights,value=enabled|disabled --region us-east-1
```

Registro em log de chamadas à API do Amazon ECS com o AWS CloudTrail

O Amazon ECS é integrado ao AWS CloudTrail, um serviço que fornece um registro de ações tomadas por um usuário, uma função ou um serviço da AWS no Amazon ECS. O CloudTrail captura todas as chamadas à API do Amazon ECS como eventos, incluindo as chamadas do console do Amazon ECS e as chamadas de código para as operações da API do Amazon ECS.

Se você criar uma trilha, você poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para o Amazon ECS. Se não configurar uma trilha, você ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history. Com as informações coletadas pelo CloudTrail, você pode determinar a solicitação feita para o Amazon ECS, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para obter mais informações, consulte o AWS CloudTrail User Guide.

Informações sobre o Amazon ECS no CloudTrail

O CloudTrail está habilitado na sua conta da AWS ao criá-la. Quando uma atividade ocorrer no Amazon ECS, ela será registrada em um evento do CloudTrail com outros eventos de serviços da AWS em Event history (Histórico de eventos). Você pode visualizar, pesquisar e fazer download de eventos recentes em sua conta da AWS. Para obter mais informações, consulte Visualizar eventos com o histórico de eventos do CloudTrail.

Para obter um registro contínuo de eventos em sua conta da AWS, incluindo eventos para o Amazon ECS, crie uma trilha. Uma trilha permite CloudTrail para fornecer arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões. A trilha registra eventos de todas as regiões na partição da AWS e fornece os arquivos de log para o bucket do Amazon S3 que você especificar. Além disso, é possível configurar outros serviços da AWS para analisar mais profundamente e agir sobre os dados de evento coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- CloudTrail Serviços compatíveis e integrações do
- Configuração de notificações do Amazon SNS para o CloudTrail
- Receber arquivos de log do CloudTrail de várias regiões e receber arquivos de log do CloudTrail de várias contas

Todas as ações do Amazon ECS são registradas em log pelo CloudTrail e documentadas no Amazon Elastic Container Service API Reference. Por exemplo, as chamadas para as seções `CreateService`, `RunTask` e `DeleteCluster` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário da raiz ou do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas das entradas dos arquivos de log do Amazon ECS

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log em um bucket do Amazon S3 que você especificar. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer origem e inclui informações sobre a ação solicitada, a data e hora da ação, os parâmetros da solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas da API pública. Assim, eles não são exibidos em nenhuma ordem específica.

> Note
>
> Estes exemplos foram formatados para obter melhor legibilidade. Em um arquivo de log do CloudTrail, todas as entradas e eventos são concatenados em uma única linha. Além disso, este exemplo foi limitado a uma única entrada do Amazon ECS. Em um arquivo de log real do CloudTrail, você vê entradas e eventos de vários serviços da AWS.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateCluster`:


```
{
    "eventVersion": "1.04",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
        "arn": "arn:aws:sts::123456789012:user/Mary_Major",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-06-20T18:32:25Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Mary_Major"
            }
        }
    },
    "eventTime": "2018-06-20T19:04:36Z",
    "eventSource": "ecs.amazonaws.com",
    "eventName": "CreateCluster",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "203.0.113.12",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "clusterName": "default"
    },
    "responseElements": {
        "cluster": {
            "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",
            "pendingTasksCount": 0,
            "registeredContainerInstancesCount": 0,
            "status": "ACTIVE",
            "runningTasksCount": 0,
            "statistics": [],
            "clusterName": "default",
            "activeServicesCount": 0
        }
    },
    "requestID": "cb8c167e-EXAMPLE",
    "eventID": "e3c6f4ce-EXAMPLE",
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789012"
}
```

Segurança no Amazon Elastic Container Service

A segurança da nuvem na AWS é a nossa maior prioridade. Como cliente da AWS, você se beneficiará de um datacenter e de uma arquitetura de rede criados para atender aos requisitos das empresas com as maiores exigências de segurança.

A segurança é uma responsabilidade compartilhada entre a AWS e você. O modelo de responsabilidade compartilhada descreve isto como segurança da nuvem e segurança na nuvem:

- Segurança da nuvem – A AWS é responsável pela proteção da infraestrutura que executa serviços da AWS na nuvem da AWS. A AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia da nossa segurança como parte dos programas de conformidade da AWS. Para saber mais sobre os programas de conformidade que se aplicam ao Amazon Elastic Container Service, consulte Serviços da AWS no escopo pelo programa de conformidade.
- Segurança na nuvem – Sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da sua empresa e as leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar o Amazon ECS. Os tópicos a seguir mostram como configurar o Amazon ECS para atender às suas metas de segurança e conformidade. Veja também como usar os serviços da AWS que ajudam a monitorar e proteger seus recursos do Amazon ECS.

Tópicos

- Gerenciamento de identidade e acesso do Amazon Elastic Container Service (p. 191)

Gerenciamento de identidade e acesso do Amazon Elastic Container Service

O AWS Identity and Access Management (IAM) é um serviço da AWS que ajuda um administrador a controlar com segurança o acesso aos recursos da AWS. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (ter permissões) para usar os recursos do Amazon ECS. O IAM é um serviço da AWS que pode ser usado sem custo adicional.

Tópicos

- Público (p. 192)
- Autenticação com identidades (p. 192)
- Gerenciamento do acesso usando políticas (p. 194)
- Como o Amazon Elastic Container Service funciona com o IAM (p. 196)
- Exemplos de políticas baseadas em identidade do Amazon Elastic Container Service (p. 200)
- Permissões no nível do recurso com suporte para ações de API do Amazon ECS (p. 209)
- Usar funções vinculadas ao serviço do Amazon ECS (p. 216)
- Políticas gerenciadas e relacionamentos de confiança (p. 221)
- Função do IAM da execução de tarefas do Amazon ECS (p. 228)
- Função do IAM programador de serviço do Amazon ECS (p. 232)

- Função do IAM CodeDeploy do Amazon ECS (p. 234)
- Função do IAM Serviço Auto Scaling do Amazon ECS (p. 237)
- Função do IAM Eventos do CloudWatch (p. 239)
- Amazon ECS Task Role (Função da tarefa do Amazon ECS) (p. 241)
- Funções do IAM para tarefas (p. 242)
- Solução de problemas de identidade e acesso do Amazon Elastic Container Service (p. 245)

Público

O uso do AWS Identity and Access Management (IAM) varia, dependendo do trabalho que você realiza no Amazon ECS.

Usuário do serviço – se você usar o Amazon ECS para fazer sua tarefa, o administrador fornecerá as credenciais e as permissões de que você precisa. À medida que usar mais recursos do Amazon ECS para fazer seu trabalho, você poderá precisar de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se não for possível acessar um recurso no Amazon ECS, consulte Solução de problemas de identidade e acesso do Amazon Elastic Container Service (p. 245).

Administrador do serviço – se você for o responsável pelos recursos do Amazon ECS em sua empresa, você provavelmente terá acesso total ao Amazon ECS. Seu trabalho é determinar quais recursos do Amazon ECS seus funcionários devem acessar. Assim, é necessário enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os conceitos básicos do IAM. Para saber mais sobre como sua empresa pode usar o IAM com o Amazon ECS, consulte Como o Amazon Elastic Container Service funciona com o IAM (p. 196).

Administrador do IAM – se você é um administrador do IAM, talvez queira saber detalhes sobre como pode escrever políticas para gerenciar o acesso ao Amazon ECS. Para visualizar exemplos de políticas baseadas em identidade do Amazon ECS que podem ser usadas no IAM, consulte Exemplos de políticas baseadas em identidade do Amazon Elastic Container Service (p. 200).

Autenticação com identidades

A autenticação é a forma como você faz login na AWS usando suas credenciais de identidade. Para obter mais informações sobre como fazer login usando o Console de gerenciamento da AWS, consulte A página de login e do console do IAM no Guia do usuário do IAM.

Você deve ser autenticado (fazer login na AWS) como o Usuário raiz da conta da AWS, um usuário do IAM, ou assumindo uma função do IAM. Também é possível usar a autenticação de logon único da sua empresa, ou até mesmo fazer login usando o Google ou o Facebook. Nesses casos, seu administrador configurou anteriormente a federação de identidades usando funções do IAM. Ao acessar a AWS usando credenciais de outra empresa, você estará assumindo uma função indiretamente.

Para fazer login diretamente no Console de gerenciamento da AWS, use sua senha com o e-mail do usuário raiz ou seu nome de usuário do IAM. É possível acessar a AWS de maneira programática usando seu usuário raiz ou as chaves de acesso do usuário do IAM. A AWS fornece ferramentas do SDK ou da linha de comando para assinar sua solicitação de forma criptográfica usando suas credenciais. Se você não utilizar ferramentas da AWS, cadastre a solicitação você mesmo. Faça isso usando o Signature versão 4, um protocolo para autenticação de solicitações de API de entrada. Para obter mais informações sobre a autenticação de solicitações, consulte Processo de cadastramento do Signature versão 4 na AWS General Reference.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, a AWS recomenda o uso da autenticação multifator (MFA) para aumentar a segurança de sua conta. Para saber mais, consulte Usar o Multi-Factor Authentication (MFA) na AWS no Guia do usuário do IAM.

Usuário raiz da conta da AWS

Ao criar uma conta da AWS, você começa com uma única identidade de login que tenha acesso total a todos os recursos e serviços da AWS na conta. Essa identidade é chamada de AWS da conta da usuário raiz e é acessada pelo login com o endereço de e-mail e a senha que você usou para criar a conta. Recomendamos que não use o usuário raiz para suas tarefas diárias, nem mesmo as administrativas. Em vez disso, siga as melhores práticas de uso do usuário raiz somente para criar seu primeiro usuário do IAM. Depois, armazene as credenciais usuário raiz com segurança e use-as para executar apenas algumas tarefas de gerenciamento de contas e de serviços.

Grupos e usuários do IAM

Um usuário do IAM é uma identidade em sua conta da AWS que tem permissões específicas para uma única pessoa ou um único aplicativo. Um usuário do IAM pode ter credenciais de longo prazo, como um nome de usuário e uma senha ou um conjunto de chaves de acesso. Para saber como gerar chaves de acesso, consulte Gerenciar chaves de acesso para usuários do IAM no Guia do usuário do IAM. Ao gerar chaves de acesso para um usuário do IAM, visualize e salve o par de chaves de maneira segura. Não será possível recuperar a chave de acesso secreta futuramente. Em vez disso, você deverá gerar outro par de chaves de acesso.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado Administradores do IAM e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de funções. Um usuário é exclusivamente associado a uma pessoa ou a um aplicativo, mas uma função pode ser assumida por qualquer pessoa que precisar dela. Os usuários têm credenciais permanentes de longo prazo, mas as funções fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de uma função) no Guia do usuário do IAM.

Funções do IAM

Uma função do IAM é uma identidade dentro de sua conta da AWS que tem permissões específicas. Ela é semelhante a um usuário do IAM, mas não está associada a uma pessoa específica. É possível assumir temporariamente uma função do IAM no Console de gerenciamento da AWS alternando funções. É possível assumir uma função chamando uma operação de API da AWS CLI ou da AWS, ou usando um URL personalizado. Para obter mais informações sobre os métodos para o uso de funções, consulte Usar funções do IAM no Guia do usuário do IAM.

As funções do IAM com credenciais temporária são úteis nas seguintes situações:

- Permissões temporárias para usuários do IAM – um usuário do IAM pode assumir uma função do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso de usuário federado – Em vez de criar um usuário do IAM, você pode usar identidades existentes do AWS Directory Service, do diretório de usuário da sua empresa ou de um provedor de identidades da Web. Estes são conhecidos como usuários federados. A AWS atribui uma função a um usuário federado quando o acesso é solicitado por meio de um provedor de identidades. Para obter mais informações sobre usuários federados, consulte Usuários federados e funções no Guia do usuário do IAM.
- Acesso entre contas – é possível usar uma função do IAM para permitir que alguém (um principal confiável) em outra conta acesse recursos em sua conta. As funções são a principal forma de conceder acesso entre contas. No entanto, alguns serviços da AWS permitem que você anexe uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como as funções do IAM diferem das políticas baseadas em recurso no Guia do usuário do IAM.

- Acesso a serviços da AWS – Uma função de serviço é uma função do IAM que um serviço assume para realizar ações em seu nome na sua conta. Ao configurar alguns ambientes de serviço da AWS, você deve definir uma função a ser assumida pelo serviço. Essa função de serviço deve incluir todas as permissões necessárias para o serviço acessar os recursos da AWS de que precisa. As funções de serviço variam de acordo com o serviço, mas muitas permitem que você escolha as permissões, desde que atenda aos requisitos documentados para esse serviço. As funções de serviço fornecem acesso apenas dentro de sua conta e não podem ser usadas para conceder acesso a serviços em outras contas. Você pode criar, modificar e excluir uma função de serviço no IAM. Por exemplo, você pode criar uma função que permita que Amazon Redshift acesse um bucket do Amazon S3 em seu nome e carregue dados desse bucket em um cluster Amazon Redshift. Para obter mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS no Guia do usuário do IAM.
- Aplicativos em execução no Amazon EC2 –Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e que fazem solicitações de API da AWS CLI ou AWS. É preferível fazer isso do que armazenar chaves de acesso na instância do EC2. Para atribuir uma função da AWS a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, crie um perfil de instância que esteja anexado à instância. Um perfil de instância contém a função e permite que programas que estão em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte Uso de uma função do IAM para conceder permissões aos aplicativos em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se você deve usar funções do IAM, consulte Quando criar uma função do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso na AWS criando políticas e anexando-as às identidades do IAM ou aos recursos da AWS. Uma política é um objeto na AWS que, quando associado a uma identidade ou a um recurso, define suas permissões. A AWS avalia essas políticas quando uma entidade (usuário raiz, usuário do IAM ou função do IAM) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas são armazenadas na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral de políticas JSON no Guia do usuário do IAM.

Um administrador do IAM pode usar políticas para especificar quem tem acesso aos recursos da AWS e quais ações essas pessoas podem executar nesses recursos. Cada entidade do IAM (usuário ou função) começa sem permissões. Em outras palavras, por padrão, os usuários não podem fazer nada, nem mesmo alterar sua própria senha. Para dar permissão a um usuário para fazer algo, um administrador deve anexar uma política de permissões ao usuário. Ou o administrador pode adicionar o usuário a um grupo que tenha as permissões pretendidas. Quando um administrador concede permissões a um grupo, todos os usuários desse grupo recebem essas permissões.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de funções do Console de gerenciamento da AWS, da AWS CLI ou da API da AWS.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos JSON de políticas de permissões que você pode anexar a uma entidade, como um usuário, uma função ou um grupo do IAM. Essas políticas controlam quais ações cada identidade pode realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criar políticas do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou

função. As políticas gerenciadas são políticas independentes que podem ser anexadas a vários usuários, grupos e funções em sua conta da AWS. As políticas gerenciadas incluem políticas gerenciadas pela AWS e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas com base em recurso

Políticas baseadas em recursos são documentos de política JSON que você anexa a um recurso, como um bucket do Amazon S3. Os administradores do serviço podem usar essas políticas para definir quais ações um principal especificado (função, usuário ou membro da conta) pode executar nesse recurso e sob quais condições. As políticas baseadas em recurso são políticas em linha. Não há políticas baseadas em recurso gerenciadas.

Listas de controle de acesso (ACLs)

As políticas de controle de acesso (ACLs) controlam quais principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recurso, embora sejam o único tipo de política que não usa o formato de documento de política JSON. O Amazon S3, o AWS WAF e a Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. Para saber mais sobre as ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

A AWS oferece suporte a outros tipos de política menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões – um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou função do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade da entidade e seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou a função no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte Limites de permissões para entidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs – Service control policies) – SCPs são políticas JSON que especificam o máximo de permissões para uma organização ou unidade organizacional (UO) no AWS Organizations. O AWS Organizations é um serviço para agrupamento e gerenciamento central das várias contas da AWS que sua empresa possui. Se você habilitar todos os recursos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades em contas-membro, incluindo cada Usuário raiz da conta da AWS. Para obter mais informações sobre Organizações e SCPs, consulte Como SCPs funcionam no Guia do usuário do AWS Organizations.
- Políticas de sessão – as políticas de sessão são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para uma função ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou da função e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política s aplicam a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como a AWS determina se deve permitir uma solicitação quando

vários tipos de política estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como o Amazon Elastic Container Service funciona com o IAM

Antes de usar o IAM para gerenciar o acesso ao Amazon ECS, você deve entender quais recursos do IAM estão disponíveis para uso com o Amazon ECS. Para obter uma visão de alto nível de como o Amazon ECS e outros serviços da AWS trabalham com o IAM, consulte AWS Serviços compatíveis com o IAM no Guia do usuário do IAM.

Tópicos

- Políticas baseadas em identidade do Amazon ECS (p. 196)
- Políticas baseadas em recursos do Amazon ECS (p. 199)
- Autorização baseada em tags do Amazon ECS (p. 199)
- Funções do IAM do Amazon ECS (p. 200)

Políticas baseadas em identidade do Amazon ECS

Com as políticas baseadas em identidade do IAM, você pode especificar ações permitidas ou negadas e recursos, bem como as condições sob as quais as ações são permitidas ou negadas. O Amazon ECS oferece suporte a ações, recursos e chaves de condição específicos. Para saber mais sobre todos os elementos que você usa em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

O elemento `Action` de uma política baseada em identidade do IAM descreve a ação ou ações específicas que serão permitidas ou negadas pela política. As ações de política geralmente têm o mesmo nome que a operação de API da AWS associada. A ação é usada em uma política para conceder permissões para executar a operação associada.

As ações de políticas no Amazon ECS usam o seguinte prefixo antes da ação: `ecs:`. Por exemplo, para conceder a alguém permissão para criar um cluster do Amazon ECS com a operação de API `CreateCluster` do Amazon ECS, inclua a ação `ecs:CreateCluster` na política da pessoa. As declarações de política devem incluir um elemento `Action` ou `NotAction`. O Amazon ECS define seu próprio conjunto de ações que descrevem as tarefas que podem ser executadas com esse serviço.

Para especificar várias ações em uma única declaração, separe-as com vírgulas, conforme o seguinte:

```
"Action": [
      "ecs:action1",
      "ecs:action2"
```

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `Describe`, inclua a seguinte ação:

```
"Action": "ecs:Describe*"
```

Para ver uma lista das ações do Amazon ECS, consulte Permissões no nível do recurso com suporte para ações de API do Amazon ECS (p. 209).

Recursos

O elemento `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Você especifica um recurso usando um ARN ou usando o caractere curinga (*) para indicar que a instrução se aplica a todos os recursos.

O recurso de cluster do Amazon ECS tem o seguinte ARN:

```
arn:${Partition}:ecs:${Region}:${Account}:cluster/${clusterName}
```

Para obter mais informações sobre o formato de ARNs, consulte Nomes de recursos da Amazon (ARNs) e namespaces de serviços da AWS.

Por exemplo, para especificar o contêiner `my-cluster` em sua instrução, use o seguinte ARN:

```
"Resource": "arn:aws:ecs:us-east-1:123456789012:cluster/my-cluster"
```

Para especificar todos os clusters que pertencem a uma conta específica, use o caractere curinga (*):

```
"Resource": "arn:aws:ecs:us-east-1:123456789012:cluster/*"
```

Algumas ações do Amazon ECS, como as ações para a criação de recursos, não podem ser executadas em um recurso específico. Nesses casos, você deve usar o caractere curinga (*).

```
"Resource": "*"
```

Algumas ações da API do Amazon ECS podem ser executadas em vários recursos. Por exemplo, é possível fazer referência a vários clusters ao chamar a ação de API `DescribeClusters`. Para especificar vários recursos em uma única instrução, separe os ARNs com vírgulas.

```
"Resource": [
      "resource1",
      "resource2"
```

A tabela a seguir descreve os ARNs para cada tipo de recurso usado pelas ações da API do Amazon ECS.

> **Important**
>
> A tabela a seguir usa o novo formato mais longo de ARN para tarefas, serviços e instâncias de contêiner do Amazon ECS. Se você não tiver optado pelo formato longo do ARN, os ARNs não incluirão o nome do cluster. Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).

Tipo de recurso	ARN
Todos os recursos do Amazon ECS	arn:aws:ecs:*
Todos os recursos do Amazon ECS de propriedade da conta especificada, na região especificada	arn:aws:ecs:region:account:*
Cluster	arn:aws:ecs:region:account:cluster/cluster-name

Tipo de recurso	ARN
Instância de contêiner	arn:aws:ecs:região:account:container-instance/cluster-name/container-instance-id
Definição de tarefa	arn:aws:ecs:region:account:task-definition/task-definition-family-name:task-definition-revision-number
Serviço	arn:aws:ecs:região:account:service/cluster-name/service-name
Tarefa	arn:aws:ecs:região:account:task/cluster-name/task-id
Contêiner	arn:aws:ecs:região:account:container/container-id

Para saber com quais ações você pode especificar o ARN de cada recurso, consulte Permissões no nível do recurso com suporte para ações de API do Amazon ECS (p. 209).

Chaves de condição

O elemento `Condition` (ou bloco de `Condition`) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam operadores de condição, como "igual a" ou "menor que", para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único elemento `Condition`, a AWS os avaliará usando uma operação lógica `AND`. Se você especificar vários valores para uma única chave de condição, a AWS avaliará a condição usando uma operação lógica `OR`. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, você pode conceder a um usuário do IAM permissão para acessar um recurso somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte Elementos de política do IAM: variáveis e tags no Guia do usuário do IAM.

O Amazon ECS define seu próprio conjunto de chaves de condição e também oferece suporte ao uso de algumas chaves de condição globais. Para consultar todas as chaves de condição global da AWS, consulte Chaves de contexto de condição global da AWS no Guia do usuário do IAM.

O Amazon ECS implementa as seguintes chaves de condição específicas ao serviço.

Chave de condição	Descrição	Tipos de avaliação
aws:RequestTag/${TagKey}	Essa chave de contexto é formatada `"aws:RequestTag/`*`tag-key`*`":"`*`tag-value`*`"` em que *`tag-key`* e *`tag-value`* são um par de chave e valor da tag. Verifica se o par de chave/valor da tag está presente em uma solicitação da AWS. Por exemplo, você pode verificar que a solicitação inclui a chave de tag `"Dept"` e se ela tem o valor `"Accounting"`.	String
aws:ResourceTag/${TagKey}	Essa chave de contexto é formatada `"aws:ResourceTag/`*`tag-key`*`":"`*`tag-value`*`"` em que *`tag-key`* e *`tag-value`* são um par de chave e valor da tag. Verifica se a tag anexada ao recurso de identidade (usuário ou função) corresponde ao nome e ao valor da chave especificada.	String

Chave de condição	Descrição	Tipos de avaliação
aws:TagKeys	Essa chave de contexto é formatada como `"aws:TagKeys":"tag-key"` em que *tag-key* é uma lista de chaves de tags sem valores (por exemplo, `["Dept","Cost-Center"]`). Verifica as chaves de tags que estão presentes em uma solicitação da AWS.	String
ecs:ResourceTag/${TagKey}	Essa chave de contexto é formatada `"ecs:ResourceTag/tag-key":"tag-value"` em que *tag-key* e *tag-value* são um par de chave e valor da tag. Verifica se a tag anexada ao recurso de identidade (usuário ou função) corresponde ao nome e ao valor da chave especificada.	String
ecs:cluster	A chave de contexto é formatada `"ecs:cluster":"cluster-arn"` em que *cluster-arn* é o ARN do cluster do Amazon ECS.	ARN, nulo
ecs:container-instances	A chave de contexto é formatada `"ecs:container-instances":"container-instance-arns"` em que *container-instance-arns* é um ou mais ARNs de instância de contêiner.	ARN, nulo
ecs:task-definition	A chave de contexto é formatada `"ecs:task-definition":"task-definition-arn"` em que *tarefa-definition-arn* é o ARN da definição de tarefa do Amazon ECS.	ARN, nulo
ecs:service	A chave de contexto é formatada `"ecs:service":"service-arn"` em que *service-arn* é o ARN para o serviço do Amazon ECS.	ARN, nulo

Para saber com quais ações e recursos você pode usar uma chave de condição, consulte Permissões no nível do recurso com suporte para ações de API do Amazon ECS (p. 209).

Exemplos

Para visualizar exemplos de políticas baseadas em identidade do Amazon ECS, consulte Exemplos de políticas baseadas em identidade do Amazon Elastic Container Service (p. 200).

Políticas baseadas em recursos do Amazon ECS

O Amazon ECS não oferece suporte a políticas baseadas em recurso.

Autorização baseada em tags do Amazon ECS

Você pode anexar tags a recursos do Amazon ECS ou passar tags em uma solicitação ao Amazon ECS. Para controlar o acesso com base em tags, forneça informações sobre as tags no elemento de condição de uma política usando as chaves de condição , `aws:RequestTag/key-name` ou `aws:TagKeys`. Para obter mais informações, consulte Controlar o acesso usando tags no Guia do usuário do IAM.

Para obter mais informações sobre recursos de marcação do Amazon ECS, consulte Recursos e tags (p. 165).

Para visualizar um exemplo de política baseada em identidade para limitar o acesso a um recurso com base nas tags desse recurso, consulte Descrever serviços do Amazon ECS com base em tags (p. 209).

Funções do IAM do Amazon ECS

Uma função do IAM é uma entidade dentro de sua conta da AWS que tem permissões específicas.

Usar credenciais temporárias com o Amazon ECS

Você pode usar credenciais temporárias para fazer login com federação, assumir uma função do IAM ou assumir uma função entre contas. As credenciais de segurança temporárias são obtidas chamando operações da API do AWS STS, como AssumeRole ou GetFederationToken.

O Amazon ECS oferece suporte ao uso de credenciais temporárias.

Funções vinculadas ao serviço

Funções vinculadas ao serviço permitem que os serviços da AWS acessem recursos em outros serviços para concluir uma ação em seu nome. As funções vinculadas ao serviço aparecem em sua conta do IAM e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para funções vinculadas ao serviço.

O Amazon ECS oferece suporte a funções vinculadas ao serviço. Para obter detalhes sobre como criar ou gerenciar funções vinculadas ao serviço do Amazon ECS , consulte Usar funções vinculadas ao serviço do Amazon ECS (p. 216).

Funções de serviço

Esse recurso permite que um serviço assuma uma função de serviço em seu nome. A função permite que o serviço acesse recursos em outros serviços para concluir uma ação em seu nome. As funções de serviço aparecem em sua conta do IAM e são de propriedade da conta. Isso significa que um administrador do IAM pode alterar as permissões para essa função. Porém, fazer isso pode alterar a funcionalidade do serviço.

O Amazon ECS oferece suporte às funções de serviço.

Exemplos de políticas baseadas em identidade do Amazon Elastic Container Service

Por padrão, os usuários e funções do IAM não têm permissão para criar ou modificar recursos do Amazon ECS. Eles também não podem executar tarefas usando o Console de gerenciamento da AWS, a AWS CLI ou uma API da AWS. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e funções permissão para executarem operações da API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas no guia JSON no Guia do usuário do IAM.

Tópicos

- Melhores práticas de políticas (p. 201)
- Permitir que os usuários visualizem suas próprias permissões (p. 201)
- Permissões do assistente de primeira execução do Amazon ECS (p. 202)
- Exemplos de cluster (p. 205)
- Listar e descrever exemplos de tarefas (p. 207)
- Exemplo para criar serviço (p. 207)
- Exemplo para atualizar serviço (p. 208)

- Descrever serviços do Amazon ECS com base em tags (p. 209)

Melhores práticas de políticas

As políticas baseadas em identidade são muito eficientes. Elas determinam se alguém pode criar, acessar ou excluir recursos do Amazon ECS em sua conta. Essas ações podem incorrer em custos para sua conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece usando políticas gerenciadas pela AWS – para começar a usar o Amazon ECS rapidamente, use as políticas gerenciadas pela AWS para conceder a seus funcionários as permissões de que precisam. Essas políticas já estão disponíveis em sua conta e são mantidas e atualizadas pela AWS. Para obter mais informações, consulte Conceitos básicos do uso de permissões com políticas gerenciadas pela AWS no Guia do usuário do IAM.
- Conceder privilégio mínimo – ao criar políticas personalizadas, conceda apenas as permissões necessárias para executar uma tarefa. Comece com um conjunto mínimo de permissões e conceda permissões adicionais conforme necessário. Fazer isso é mais seguro do que começar com permissões que são muito lenientes e tentar restringi-las posteriormente. Para obter mais informações, consulte Conceder privilégio mínimo, no Guia do usuário do IAM.
- Habilitar o MFA para operações confidenciais – para segurança adicional, exija que os usuários do IAM usem a autenticação multifator (MFA) para acessar recursos ou operações de API confidenciais. Para obter mais informações, consulte Usar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.
- Usar condições de política para segurança adicional – na medida do possível, defina as condições sob as quais suas políticas baseadas em identidade permitem o acesso a um recurso. Por exemplo, você pode gravar condições para especificar um intervalo de endereços IP permitidos do qual a solicitação deve partir. Você também pode escrever condições para permitir somente solicitações em uma data especificada ou período ou para exigir o uso de SSL ou MFA. Para obter mais informações, consulte Elementos da política JSON do IAM: condição no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas à identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou de forma programática usando a AWS CLI ou a API da AWS.

```
{
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "ViewOwnUserInfo",
               "Effect": "Allow",
               "Action": [
                   "iam:GetUserPolicy",
                   "iam:ListGroupsForUser",
                   "iam:ListAttachedUserPolicies",
                   "iam:ListUserPolicies",
                   "iam:GetUser"
               ],
               "Resource": [
                   "arn:aws:iam::*:user/${aws:username}"
               ]
           },
           {
               "Sid": "NavigateInConsole",
               "Effect": "Allow",
               "Action": [
                   "iam:GetGroupPolicy",
```

```
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Permissões do assistente de primeira execução do Amazon ECS

O assistente da primeira execução do Amazon ECS simplifica o processo de criação de um cluster e de execução das tarefas e dos serviços. Porém, os usuários precisam de permissões para muitas operações de API de vários serviços do AWS para concluir o assistente. A política gerenciada AmazonECS_FullAccess (p. 222) abaixo mostra as permissões necessárias para concluir o assistente da primeira execução do Amazon ECS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:Describe*",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:PutMetricAlarm",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:GetApplication",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:ListApplications",
                "codedeploy:ListDeploymentGroups",
                "codedeploy:ListDeployments",
                "codedeploy:StopDeployment",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:ListDeploymentTargets",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetApplicationRevision",
                "codedeploy:RegisterApplicationRevision",
```

```
                "codedeploy:BatchGetApplicationRevisions",
                "codedeploy:BatchGetDeploymentGroups",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetApplications",
                "codedeploy:ListApplicationRevisions",
                "codedeploy:ListDeploymentConfigs",
                "codedeploy:ContinueDeployment",
                "sns:ListTopics",
                "lambda:ListFunctions",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotFleetRequests",
                "ec2:CreateInternetGateway",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSubnet",
                "ec2:DeleteVpc",
                "ec2:Describe*",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:RunInstances",
                "ec2:RequestSpotFleet",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateRule",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteRule",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTargetGroups",
                "ecs:*",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:ListRuleNamesByTarget",
                "events:ListTargetsByRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "iam:ListAttachedRolePolicies",
                "iam:ListInstanceProfiles",
                "iam:ListRoles",
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:FilterLogEvents",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetHealthCheck",
                "servicediscovery:CreatePrivateDnsNamespace",
                "servicediscovery:CreateService",
                "servicediscovery:GetNamespace",
                "servicediscovery:GetOperation",
                "servicediscovery:GetService",
```

```
                "servicediscovery:ListNamespaces",
                "servicediscovery:ListServices",
                "servicediscovery:UpdateService"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/aws/service/ecs*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteInternetGateway",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "EC2ContainerService-
*"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/ecsInstanceRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
```

```
                "arn:aws:iam::*:role/ecsAutoscaleRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "application-autoscaling.amazonaws.com",
                        "application-autoscaling.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "ecs.amazonaws.com",
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "ecs.application-autoscaling.amazonaws.com",
                        "autoscaling.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

O assistente de primeira execução também tenta criar automaticamente funções do IAM diferentes dependendo do tipo de execução das tarefas usadas. Os exemplos são a função de serviço do Amazon ECS, a função do IAM de instância de contêiner e a função do IAM de execução da tarefa. Para garantir que a experiência da primeira execução seja capaz de criar essas funções do IAM, uma das seguintes opções devem ser verdadeiras:

- O usuário tem acesso de administrador. Para obter mais informações, consulte Configuração com o Amazon ECS (p. 7).
- O usuário tem as permissões do IAM para criar uma função do serviço. Para obter mais informações, consulte Criar uma função para delegar permissões a um serviço da AWS.
- Você tem um usuário com acesso de administrador para criar a função do IAM obrigatória, de maneira que esteja disponível na conta a ser usada. Para obter mais informações, consulte as informações a seguir:
  - Função do IAM programador de serviço do Amazon ECS (p. 232)
  - Função do IAM da execução de tarefas do Amazon ECS (p. 228)

Exemplos de cluster

A política do IAM a seguir permite que a permissão crie e liste clusters. Como as ações `CreateCluster` e `ListClusters` não aceitam recursos, a definição de recurso é definida como `*` para todos os recursos.

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

```
      "Resource": [
        "*"
      ]
    }
  ]
}
```

A política do IAM a seguir permite que a permissão descreva e exclua um cluster específico. As ações `DescribeClusters` e `DeleteCluster` aceitam os ARNs de cluster como recursos.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeCluster",
        "ecs:DeleteCluster"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:<aws_account_id>:cluster/<cluster_name>"
      ]
    }
  ]
}
```

A política do IAM a seguir pode ser anexada a um usuário ou grupo que só permitiria que esse usuário ou grupo realizasse operações em um cluster específico.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:Describe*",
        "ecs:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "ecs:DeleteCluster",
        "ecs:DeregisterContainerInstance",
        "ecs:ListContainerInstances",
        "ecs:RegisterContainerInstance",
        "ecs:SubmitContainerStateChange",
        "ecs:SubmitTaskStateChange"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ecs:us-east-1:<aws_account_id>:cluster/default"
    },
    {
      "Action": [
        "ecs:DescribeContainerInstances",
        "ecs:DescribeTasks",
        "ecs:ListTasks",
        "ecs:UpdateContainerAgent",
        "ecs:StartTask",
        "ecs:StopTask",
        "ecs:RunTask"
      ],
      "Effect": "Allow",
```

```
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:us-east-1:<aws_account_id>:cluster/default"
        }
      }
    }
  ]
}
```

Listar e descrever exemplos de tarefas

A política do IAM a seguir permite que um usuário liste tarefas para um cluster especificado:

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

A política do IAM a seguir permite que um usuário descreva uma tarefa especificada em um cluster especificado:

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

Exemplo para criar serviço

A política do IAM a seguir permite que um usuário crie serviços do Amazon ECS no Console de gerenciamento da AWS:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:Describe*",
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling:RegisterScalableTarget",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "ecs:List*",
        "ecs:Describe*",
        "ecs:CreateService",
        "elasticloadbalancing:Describe*",
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:ListRoles",
        "iam:ListGroups",
        "iam:ListUsers"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Exemplo para atualizar serviço

A política do IAM a seguir permite que um usuário atualize serviços do Amazon ECS no Console de gerenciamento da AWS:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:Describe*",
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling:DeleteScalingPolicy",
        "application-autoscaling:RegisterScalableTarget",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "ecs:List*",
        "ecs:Describe*",
        "ecs:UpdateService",
        "iam:AttachRolePolicy",
        "iam:CreateRole",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:ListRoles",
        "iam:ListGroups",
        "iam:ListUsers"
      ],
      "Resource": [
```

```
        "*"
      ]
    }
  ]
}
```

Descrever serviços do Amazon ECS com base em tags

Você pode usar condições em sua política baseada em identidade para controlar o acesso aos recursos do Amazon ECS com base em tags. Este exemplo mostra como você pode criar uma política que permite descrever seus serviços. No entanto, a permissão será concedida somente se a tag `Owner` tiver o valor do nome desse usuário. Essa política também concede as permissões necessárias concluir essa ação no console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DescribeServices",
            "Effect": "Allow",
            "Action": "ecs:DescribeServices",
            "Resource": "*"
        },
        {
            "Sid": "ViewServiceIfOwner",
            "Effect": "Allow",
            "Action": "ecs:DescribeServices",
            "Resource": "arn:aws:ecs:*:*:service/*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Você pode anexar essa política aos usuários do IAM na sua conta. Se um usuário chamado `richard-roe` tentar descrever um serviço do Amazon ECS , o serviço deverá ser marcado `Owner=richard-roe` ou `owner=richard-roe`. Caso contrário, ele terá o acesso negado. A chave da tag de condição `Owner` corresponde a `Owner` e a `owner` porque os nomes de chaves de condição não fazem distinção entre maiúsculas e minúsculas. Para obter mais informações, consulte Elementos de política JSON do IAM: condição no Guia do usuário do IAM.

Permissões no nível do recurso com suporte para ações de API do Amazon ECS

O termo Permissões no nível do recurso se refere à capacidade de especificar quais recursos nos quais os usuários têm permissões para executar ações. O Amazon ECS oferece suporte parcial a permissões no nível do recurso. Isso significa que, para determinadas ações do Amazon ECS, você pode controlar quando os usuários têm permissão para usar essas ações com base em condições que devem ser atendidas, ou os recursos específicos que os usuários têm permissão para usar. Por exemplo, você pode conceder aos usuários permissão para ativar instâncias, mas apenas de um tipo específico, e usando um AMI específico.

Considerações sobre permissões em nível de recurso

Ao controlar o acesso a ações da API do Amazon ECS especificando o nome de recurso da Amazon (ARN) de um recurso em uma política do IAM, lembre-se que o ECS incluiu uma configuração de conta que afeta o formato do ARN para instâncias de contêiner, serviços e tarefas. Para usar permissões em

nível de recurso, recomendamos que você escolha o novo formato mais longo do ARN. Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).

Quando uma política do IAM é avaliada, os recursos especificados são avaliados com base em sua utilização do novo formato mais longo do ARN. Veja a seguir exemplos de como o acesso é controlado.

Especificação de um serviço com um apenas um cluster com um curinga

Exemplo: `arn:aws:ecs:region:aws_account_id:service/cluster_name*`

Neste exemplo, o acesso será controlado para os seguintes serviços:

- Todos os serviços que usam o novo formato do ARN que estão no cluster `cluster_name*`.
- Todos os serviços que usam o antigo formato do ARN que estão no cluster `cluster_name*`.

> **Important**
>
> Isso NÃO controlará o acesso aos serviços que usam o antigo formato do ARN que têm um nome de serviço com o prefixo `cluster_name` que não estão no cluster `cluster_name*`.

Especificação de um serviço com um cluster e um nome de serviço usando um curinga

Exemplo: `arn:aws:ecs:region:aws_account_id:service/cluster_name/service_name*`

Neste exemplo, o acesso será controlado para os seguintes serviços:

- Todos os serviços que usam o novo formato do ARN que estão no cluster `cluster_name` com o prefixo `service_name`.
- Todos os serviços que usam o antigo formato do ARN que estão no cluster `cluster_name` com o prefixo `service_name`, mesmo que o ARN real do serviço ainda tenha o formato `arn:aws:ecs:region:aws_account_id:service/service_name*` do ARN.

Especificação de um serviço com um ARN completo

Exemplo: `arn:aws:ecs:region:aws_account_id:service/cluster_name/service_name`

Neste exemplo, o acesso será controlado para os seguintes serviços:

- Todos os serviços que usam o novo formato do ARN que estão no cluster `cluster_name` com o nome de serviço `service_name`.
- Todos os serviços que usam o formato do ARN antigo que estão no cluster `cluster_name` com o nome do serviço `service_name`, mesmo que o ARN real do serviço ainda tenha o formato `arn:aws:ecs:region:aws_account_id:service/service_name` do ARN.

Permissões em nível de recurso

A tabela a seguir descreve as ações de API do Amazon ECS que dão suporte no momento a permissões no nível do recurso, bem como os recursos suportados, os ARNs de recurso e as chaves de condição para cada ação.

> **Important**
>
> Se uma ação da API do Amazon ECS nessa tabela estiver marcada como `N/A` em `Resource` e em `Condition keys`, isso significa que ela não oferece suporte a permissões no nível do recurso. É possível conceder aos usuários permissão para usar a ação, mas você deve especificar um `*` para o elemento do recurso de sua declaração de política.

Ação de API	Recurso	Chaves de condição
CreateCluster	N/A	aws:RequestTag/${TagKey} aws:TagKeys
CreateService	Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:RequestTag/${TagKey} aws:ResourceTag ecs:ResourceTag ecs:task-definition ecs:cluster aws:TagKeys
CreateTaskSet	N/A	ecs:service ecs:task-definition ecs:cluster
DeleteAccountSetting	N/A	N/A
DeleteAttributes	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
DeleteCluster	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
DeleteService	Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:ResourceTag ecs:ResourceTag ecs:cluster
DeleteTaskSet	TaskSet arn:aws:ecs:*região*:*aws_account_id*:cluster/*service-name*/*taskset-id*	ecs:service ecs:cluster
DeregisterContainerInstance	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
DeregisterTaskDefinition	N/A	N/A
DescribeClusters	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster1*, arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster2*	aws:ResourceTag ecs:ResourceTag

Ação de API	Recurso	Chaves de condição
DescribeContainerInstances	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
DescribeServices	Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:ResourceTag ecs:ResourceTag ecs:cluster
DescribeTaskDefinition	N/A	N/A
DescribeTasks	Task arn:aws:ecs:*region*:*aws_account_id*:task/*cluster-name*/*task-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
DescribeTaskSets	TaskSet arn:aws:ecs:*região*:*aws_account_id*:service/*service-name*/*taskset-id*	ecs:service ecs:cluster
DiscoverPollEndpoint	N/A	N/A
ListAccountSettings	N/A	N/A
ListAttributes	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
ListClusters	N/A	N/A
ListContainerInstances	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
ListServices	N/A	ecs:cluster

Ação de API	Recurso	Chaves de condição
ListTagsForResource	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	
	Task arn:aws:ecs:*region*:*aws_account_id*:task/*cluster-name*/*task-id*	
	Definição de tarefa arn:aws:ecs:*região*:*aws_account_id*:task-definition/*hello_world:8*	
ListTaskDefinitionFamilies	N/A	N/A
ListTaskDefinitions	N/A	N/A
ListTasks	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
Poll	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
PutAccountSetting	N/A	N/A
PutAccountSettingDefault	N/A	N/A
PutAttributes	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
RegisterContainerInstance	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:RequestTag/${TagKey} aws:ResourceTag ecs:ResourceTag aws:TagKeys
RegisterTaskDefinition	N/A	aws:RequestTag/${TagKey} aws:TagKeys

Ação de API	Recurso	Chaves de condição
RunTask	Definição de tarefa arn:aws:ecs:*região*:*aws_account_id*:task-definition/*hello_world:8*	aws:RequestTag/${TagKey} aws:ResourceTag ecs:ResourceTag aws:TagKeys ecs:cluster
StartTask	Definição de tarefa arn:aws:ecs:*região*:*aws_account_id*:task-definition/*hello_world:8*	aws:RequestTag/${TagKey} aws:ResourceTag ecs:ResourceTag aws:TagKeys ecs:cluster ecs:container-instances
StartTelemetrySession	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
StopTask	Task arn:aws:ecs:*region*:*aws_account_id*:task/*cluster-name*/*task-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
SubmitContainerStateChange	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag
SubmitTaskStateChange	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster*	aws:ResourceTag ecs:ResourceTag

Ação de API	Recurso	Chaves de condição
TagResource	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster* Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id* Task arn:aws:ecs:*region*:*aws_account_id*:task/*cluster-name*/*task-id* Definição de tarefa arn:aws:ecs:*região*:*aws_account_id*:task-definition/*hello_world:8* Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:RequestTag/${TagKey} aws:ResourceTag ecs:ResourceTag aws:TagKeys
UntagResource	Cluster arn:aws:ecs:*região*:*aws_account_id*:cluster/*my-cluster* Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id* Task arn:aws:ecs:*region*:*aws_account_id*:task/*cluster-name*/*task-id* Definição de tarefa arn:aws:ecs:*região*:*aws_account_id*:task-definition/*hello_world:8* Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:ResourceTag ecs:ResourceTag aws:TagKeys
UpdateContainerAgent	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster

Ação de API	Recurso	Chaves de condição
UpdateContainerInstancesState	Container instance arn:aws:ecs:*region*:*aws_account_id*:container-instance/*cluster-name*/*container-instance-id*	aws:ResourceTag ecs:ResourceTag ecs:cluster
UpdateService	Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:ResourceTag ecs:ResourceTag ecs:cluster ecs:task-definition
UpdateServicePrimaryTaskSet	Service arn:aws:ecs:*region*:*aws_account_id*:service/*cluster-name*/*service-name*	aws:ResourceTag ecs:ResourceTag ecs:cluster
UpdateTaskSet	TaskSet arn:aws:ecs:*região*:*aws_account_id*:cluster/*service-name*/*taskset-id*	ecs:cluster ecs:service

Usar funções vinculadas ao serviço do Amazon ECS

Amazon Elastic Container Service usa AWS Identity and Access Management (IAM) funções vinculadas ao serviço. A função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente ao Amazon ECS. As funções vinculadas a serviços são predefinidas pelo Amazon ECS e incluem todas as permissões que o serviço requer para chamar outros serviços da AWS em seu nome.

Uma função vinculada ao serviço facilita a configuração do Amazon ECS porque você não precisa adicionar as permissões necessárias manualmente. Amazon ECS define as permissões de suas funções vinculadas ao serviço e, a menos que definido em contrário, somente Amazon ECS pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, e essa política não pode ser anexada a nenhuma outra entidade do IAM.

Você pode excluir as funções somente depois de primeiro excluir seus recursos relacionados. Isso protege seus recursos do Amazon ECS, pois você não pode remover por engano as permissões para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte às funções vinculadas a serviço, consulte Serviços da AWS compatíveis com o IAM e procure os serviços que apresentam Sim na coluna Função vinculada a serviços. Escolha um Sim com um link para exibir a documentação da função vinculada a serviço desse serviço.

Permissões de função vinculada ao serviço do Amazon ECS

O Amazon ECS usa a função vinculada ao serviço chamada AWSServiceRoleForECS – Role to enable Amazon ECS to manage your cluster..

A função vinculada ao serviço AWSServiceRoleForECS confia nos seguintes serviços para assumir a função:

- `ecs.amazonaws.com`

A política de permissões da função permite que o Amazon ECS conclua as seguintes ações nos recursos especificados:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ECSTaskManagement",
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:Describe*",
                "ec2:DetachNetworkInterface",
                "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
                "elasticloadbalancing:DeregisterTargets",
                "elasticloadbalancing:Describe*",
                "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
                "elasticloadbalancing:RegisterTargets",
                "route53:ChangeResourceRecordSets",
                "route53:CreateHealthCheck",
                "route53:DeleteHealthCheck",
                "route53:Get*",
                "route53:List*",
                "route53:UpdateHealthCheck",
                "servicediscovery:DeregisterInstance",
                "servicediscovery:Get*",
                "servicediscovery:List*",
                "servicediscovery:RegisterInstance",
                "servicediscovery:UpdateInstanceCustomHealthStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ECSTagging",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*"
        }
    ]
}
```

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada ao serviço.

Para permitir que uma entidade do IAM crie a função vinculada ao serviço AWSServiceRoleForECS

Adicione a seguinte instrução à política de permissões para a entidade do IAM que precisa criar a função vinculada ao serviço:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS*",
```

```
    "Condition": {"StringLike": {"iam:AWSServiceName": "ecs.amazonaws.com"}}
}
```

Para permitir que uma entidade do IAM edite a descrição da função vinculada ao serviço AWSServiceRoleForECS

Adicione a instrução a seguir à política de permissões do IAM para a entidade que precisa editar a descrição de uma função vinculada ao serviço:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "ecs.amazonaws.com"}}
}
```

Para permitir que uma entidade do IAM exclua a função vinculada ao serviço AWSServiceRoleForECS

Adicione a seguinte instrução à política de permissões para a entidade do IAM que precisa excluir uma função vinculada ao serviço:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "ecs.amazonaws.com"}}
}
```

Criação de uma função vinculada a um serviço do Amazon ECS

Na maioria das circunstâncias, você não precisa criar manualmente uma função vinculada ao serviço. Por exemplo, quando você criar um novo cluster (por exemplo, com a primeira execução do Amazon ECS, o assistente de criação do cluster ou a AWS CLI ou os SDKs) ou criar ou atualizar um serviço no Console de gerenciamento da AWS, o Amazon ECS criará a função vinculada ao serviço para você, caso ela ainda não exista.

> Important
>
> A entidade do IAM que está criando o cluster deve ter as permissões adequadas do IAM para criar a função vinculada ao serviço e aplicar uma política a ele. Caso contrário, a criação automática falhará.

Criação de uma função vinculada ao serviço no IAM (AWS CLI)

Você pode usar os comandos do IAM na AWS Command Line Interface para criar uma função vinculada ao serviço com a política de confiança e as políticas em linha que o serviço precisa para assumir a função.

Para criar uma função vinculada ao serviço (CLI)

Use o seguinte comando:

```
$ aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

Edição de uma função vinculada ao serviço do Amazon ECS

O Amazon ECS não permite que você edite a função vinculada ao serviço AWSServiceRoleForECS. Depois que criar uma função vinculada ao serviço, você não poderá alterar o nome da função, pois várias entidades podem fazer referência a ela. No entanto, você pode editar a descrição da função. Para obter mais informações, consulte Modificar uma função, no Guia do usuário do IAM.

Exclusão de uma função vinculada ao serviço do Amazon ECS

Se você não usar mais o Amazon ECS, recomendamos que você exclua a função. Dessa forma, você não tem uma entidade não utilizada que não seja monitorada ativamente ou mantida. Contudo, você deve excluir todos os clusters do Amazon ECS em todas as regiões para poder excluir a função vinculada ao serviço.

Limpeza de uma função vinculada a serviço

Antes de você usar o IAM para excluir uma função vinculada ao serviço, é preciso primeiro confirmar que a função não tem sessões ativas e excluir todos os clusters do Amazon ECS em todas as regiões da AWS.

Para verificar se a função vinculada ao serviço tem uma sessão ativa

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções) e escolha o nome AWSServiceRoleForECS (não a caixa de seleção).
3. Na página Summary (Resumo), escolha Access Advisor (Consultor de acesso) e analise as atividades recentes para a função vinculada ao serviço.

   Note

   Se não tiver certeza se o Amazon ECS está usando a função AWSServiceRoleForECS, você poderá tentar excluir a função. Se o serviço está usando a função, a exclusão falha e você pode visualizar as regiões em que a função está sendo usada. Se a função está sendo usada, você deve aguardar a sessão final antes de excluir a função. Você não pode revogar a sessão para uma função vinculada a serviço.

Para remover os recursos do Amazon ECS usados pela função vinculada ao serviço AWSServiceRoleForECS

Você deve excluir todos os clusters do Amazon ECS em todas as regiões da AWS antes de excluir a função AWSServiceRoleForECS.

- Exclua todos os clusters do Amazon ECS em todas as regiões. Para obter mais informações, consulte Exclusão de um cluster (p. 23).

Exclusão de uma função vinculada ao serviço no IAM (Console)

Também é possível usar o console do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (console)

1. Faça login no Console de gerenciamento da AWS e abra o console da IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação do console do IAM, selecione Roles (Funções). Em seguida, marque a caixa de seleção ao lado de AWSServiceRoleForECS, não o nome ou a linha em si.
3. Selecione Delete role (Excluir função).
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço, que mostram quando cada uma das funções selecionadas acessou pela última vez um serviço da AWS. Isso ajuda você a confirmar se a função está ativo no momento. Se você deseja continuar, escolha Sim, excluir para enviar a função vinculada ao serviço para exclusão.
5. Acompanhe as notificações do console do IAM para monitorar o progresso da exclusão da função vinculada ao serviço. Como a exclusão da função vinculada ao serviço do IAM é assíncrona, depois de enviar a função para exclusão, a tarefa de exclusão pode ou não ser bem-sucedida.

   - Se a tarefa for bem-sucedida, a função será removida da lista e uma notificação de sucesso será exibida na parte superior da página.
   - Se a tarefa falhar, você pode escolher Visualizar detalhes ou Exibir recursos a partir das notificações para saber por que a exclusão falhou. Se a exclusão falhar porque a função está usando os recursos do serviço, a notificação inclui uma lista de recursos, se o serviço retorna essas informações. Você pode então limpar os recursos e submeter novamente a exclusão.

     Note

     Você pode repetir esse processo várias vezes, de acordo com as informações que o serviço retorna. Por exemplo, a função vinculada ao serviço pode usar seis recursos e seu serviço pode retornar informações sobre cinco deles. Se você limpar cinco recursos e enviar a função para exclusão novamente, a deleção falha e o serviço reporta o recurso remanescente. Um serviço pode retornar todos os recursos, alguns deles, ou pode não reportar nenhum recurso.
   - Se a tarefa falhar e a notificação não inclui uma lista de recursos, o serviço pode não retornar essas informações. Para saber como limpar os recursos para esse serviço, consulte Serviços da AWS compatíveis com o IAM. Encontre o serviço na tabela e escolha o link Sim para visualizar a documentação da função vinculada desse serviço.

Excluir uma função vinculada ao serviço no IAM (AWS CLI)

Você pode usar comandos do IAM na AWS Command Line Interface para excluir uma função vinculada ao serviço.

Para excluir uma função vinculado ao serviço (CLI)

1. Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tem recursos associados, você deve enviar uma solicitação de exclusão. Essa solicitação pode ser negada se essas condições não forem atendidas. Você deve capturar o `deletion-task-id` da resposta para verificar o estado da tarefa de exclusão. Digite o seguinte comando para enviar uma solicitação de exclusão de função vinculada ao serviço:

   ```
   $ aws iam delete-service-linked-role --role-name AWSServiceRoleForECS+OPTIONAL-SUFFIX
   ```
2. Use o seguinte comando para verificar o status da tarefa de exclusão:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada retorna o motivo de falha para que você possa solucionar problemas. Se a exclusão falhar porque a função está usando os recursos do serviço, a notificação inclui uma lista de recursos, se o serviço retorna essas informações. Você pode então limpar os recursos e submeter novamente a exclusão.

Note

Você pode repetir esse processo várias vezes, de acordo com as informações que o serviço retorna. Por exemplo, a função vinculada ao serviço pode usar seis recursos e seu serviço pode retornar informações sobre cinco deles. Se você limpar cinco recursos e enviar a função para exclusão novamente, a deleção falha e o serviço reporta o recurso remanescente. Um serviço pode retornar todos os recursos, alguns deles, ou pode não reportar nenhum recurso. Para saber como limpar os recursos para um serviço que não reporta nenhum recurso, consulte Serviços da AWS compatíveis com o IAM. Encontre o serviço na tabela e escolha o link Sim para visualizar a documentação da função vinculada desse serviço.

Excluir uma função vinculada ao serviço no IAM (API da AWS)

É possível usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculado ao serviço (API)

1. Para enviar uma solicitação de exclusão de um roll vinculada ao serviço, chame DeleteServiceLinkedRole. Na solicitação, especifique o nome da função AWSServiceRoleForECS.

   Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tem recursos associados, você deve enviar uma solicitação de exclusão. Essa solicitação pode ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o estado da tarefa de exclusão.
2. Para verificar o status da exclusão, chame GetServiceLinkedRoleDeletionStatus. Na solicitação, especifique o `DeletionTaskId`.

   O status da tarefa de exclusão pode ser `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, ou `FAILED`. Se a exclusão falhar, a chamada retorna o motivo de falha para que você possa solucionar problemas. Se a exclusão falhar porque a função está usando os recursos do serviço, a notificação inclui uma lista de recursos, se o serviço retorna essas informações. Você pode então limpar os recursos e submeter novamente a exclusão.

   Note

   Você pode repetir esse processo várias vezes, de acordo com as informações que o serviço retorna. Por exemplo, a função vinculada ao serviço pode usar seis recursos e seu serviço pode retornar informações sobre cinco deles. Se você limpar cinco recursos e enviar a função para exclusão novamente, a deleção falha e o serviço reporta o recurso remanescente. Um serviço pode retornar todos os recursos, alguns deles, ou pode não reportar nenhum recurso. Para saber como limpar os recursos para um serviço que não reporta nenhum recurso, consulte Serviços da AWS compatíveis com o IAM. Encontre o serviço na tabela e escolha o link Sim para visualizar a documentação da função vinculada desse serviço.

Políticas gerenciadas e relacionamentos de confiança

O Amazon ECS e o Amazon ECR fornecem várias políticas gerenciadas e relacionamentos de confiança que você pode anexar aos usuários do IAM, às instâncias do EC2 e às tarefas do Amazon ECS que permitem níveis de controle diferentes sobre as operações de API e os recursos. Você pode aplicar essas políticas diretamente ou usá-las como ponto de partida para criar suas próprias políticas.

Tópicos

- Políticas gerenciadas e relacionamentos de confiança do Amazon ECS (p. 222)
- Políticas gerenciadas do Amazon ECR (p. 227)

Políticas gerenciadas e relacionamentos de confiança do Amazon ECS

O Amazon ECS fornece várias políticas gerenciadas e relacionamentos de confiança que você pode anexar aos usuários do e o IAM ou às instâncias do EC2 ou tarefas do Amazon ECS que permitem níveis de controle diferentes sobre as operações de API e os recursos do Amazon ECS. Você pode aplicar essas políticas diretamente ou usá-las como ponto de partida para criar suas próprias políticas. Para obter mais informações sobre cada uma das operações de API mencionadas nessas políticas, consulte Ações no Amazon Elastic Container Service API Reference.

Tópicos

- AmazonECS_FullAccess (p. 222)
- AmazonEC2ContainerServiceFullAccess (p. 225)
- AmazonEC2ContainerServiceRole (p. 226)
- AmazonEC2ContainerServiceAutoscaleRole (p. 226)

AmazonECS_FullAccess

Esta política gerenciada fornece acesso administrativo a recursos do Amazon ECS e habilita recursos do ECS por meio do acesso a outros recursos do serviço da AWS, incluindo VPCs, grupos do Auto Scaling e pilhas do AWS CloudFormation.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:Describe*",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:PutMetricAlarm",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:GetApplication",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:ListApplications",
                "codedeploy:ListDeploymentGroups",
                "codedeploy:ListDeployments",
                "codedeploy:StopDeployment",
```

```
                "codedeploy:GetDeploymentTarget",
                "codedeploy:ListDeploymentTargets",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetApplicationRevision",
                "codedeploy:RegisterApplicationRevision",
                "codedeploy:BatchGetApplicationRevisions",
                "codedeploy:BatchGetDeploymentGroups",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetApplications",
                "codedeploy:ListApplicationRevisions",
                "codedeploy:ListDeploymentConfigs",
                "codedeploy:ContinueDeployment",
                "sns:ListTopics",
                "lambda:ListFunctions",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotFleetRequests",
                "ec2:CreateInternetGateway",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSubnet",
                "ec2:DeleteVpc",
                "ec2:Describe*",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:RunInstances",
                "ec2:RequestSpotFleet",
                "elasticloadbalancing:CreateListener",
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateRule",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteRule",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTargetGroups",
                "ecs:*",
                "events:DescribeRule",
                "events:DeleteRule",
                "events:ListRuleNamesByTarget",
                "events:ListTargetsByRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "iam:ListAttachedRolePolicies",
                "iam:ListInstanceProfiles",
                "iam:ListRoles",
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:FilterLogEvents",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetHealthCheck",
```

```
                "servicediscovery:CreatePrivateDnsNamespace",
                "servicediscovery:CreateService",
                "servicediscovery:GetNamespace",
                "servicediscovery:GetOperation",
                "servicediscovery:GetService",
                "servicediscovery:ListNamespaces",
                "servicediscovery:ListServices",
                "servicediscovery:UpdateService"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParametersByPath",
                "ssm:GetParameters",
                "ssm:GetParameter"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/aws/service/ecs*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteInternetGateway",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "ec2:ResourceTag/aws:cloudformation:stack-name": "EC2ContainerService-
*"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ecs-tasks.amazonaws.com"
                }
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/ecsInstanceRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                }
            }
```

```
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::*:role/ecsAutoscaleRole*"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "application-autoscaling.amazonaws.com",
                        "application-autoscaling.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "ecs.amazonaws.com",
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "ecs.application-autoscaling.amazonaws.com",
                        "autoscaling.amazonaws.com"
                    ]
                }
            }
        }
    ]
}
```

AmazonEC2ContainerServiceFullAccess

Essa política gerenciada permite o acesso total do administrador ao Amazon ECS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "autoscaling:Describe*",
        "autoscaling:UpdateAutoScalingGroup",
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStack*",
        "cloudformation:UpdateStack",
        "cloudwatch:GetMetricStatistics",
        "ec2:Describe*",
        "elasticloadbalancing:*",
        "ecs:*",
        "events:DescribeRule",
        "events:DeleteRule",
        "events:ListRuleNamesByTarget",
        "events:ListTargetsByRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "iam:ListInstanceProfiles",
        "iam:ListRoles",
```

```
        "iam:PassRole"
      ],
      "Resource": "*"
    }
  ]
}
```

AmazonEC2ContainerServiceRole

Essa política gerenciada permite que os load balancers do Elastic Load Balancing registrem e cancelem o registro das instâncias de contêiner do Amazon ECS em seu nome. Para obter mais informações, consulte Função do IAM programador de serviço do Amazon ECS (p. 232).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:Describe*",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:Describe*",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:RegisterTargets"
      ],
      "Resource": "*"
    }
  ]
}
```

AmazonEC2ContainerServiceAutoscaleRole

Essa política permite que o Application Auto Scaling dimensione a contagem crescente e decrescente desejada do serviço Amazon ECS em resposta aos alarmes do CloudWatch em seu nome. Para obter mais informações, consulte Função do IAM Serviço Auto Scaling do Amazon ECS (p. 237).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1456535218000",
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeServices",
                "ecs:UpdateService"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Stmt1456535243000",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "*"
            ]
        }
```

```
    ]
}
```

Políticas gerenciadas do Amazon ECR

O Amazon ECR fornece várias políticas gerenciadas que você pode anexar aos usuários do IAM ou às instâncias do EC2 que permitem níveis de controle diferentes nas operações de API e nos recursos do Amazon ECR. Você pode aplicar essas políticas diretamente ou usá-las como ponto de partida para criar suas próprias políticas. Para obter mais informações sobre cada uma das operações de API mencionadas nessas políticas, consulte Ações no Amazon Elastic Container Registry API Reference.

Tópicos

- AmazonEC2ContainerRegistryFullAccess (p. 227)
- AmazonEC2ContainerRegistryPowerUser (p. 227)
- AmazonEC2ContainerRegistryReadOnly (p. 228)

AmazonEC2ContainerRegistryFullAccess

Essa política gerenciada permite o acesso total do administrador ao Amazon ECR.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:*"
      ],
      "Resource": "*"
    }
  ]
}
```

AmazonEC2ContainerRegistryPowerUser

Essa política gerenciada permite o acesso de usuário avançado ao Amazon ECR, o que oferece acesso de leitura e gravação nos repositórios, mas não permite que os usuários excluam repositórios ou alterem os documentos de política aplicados a eles.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "ecr:GetAuthorizationToken",
   "ecr:BatchCheckLayerAvailability",
   "ecr:GetDownloadUrlForLayer",
   "ecr:GetRepositoryPolicy",
   "ecr:DescribeRepositories",
   "ecr:ListImages",
   "ecr:DescribeImages",
   "ecr:BatchGetImage",
   "ecr:InitiateLayerUpload",
   "ecr:UploadLayerPart",
   "ecr:CompleteLayerUpload",
   "ecr:PutImage"
  ],
  "Resource": "*"
```

```
 }]
}
```

AmazonEC2ContainerRegistryReadOnly

Essa política gerenciada permite o acesso somente leitura ao Amazon ECR, como a capacidade de listar repositórios e imagens nos repositórios, além de extrair imagens do Amazon ECR com a CLI do Docker.

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "ecr:GetAuthorizationToken",
   "ecr:BatchCheckLayerAvailability",
   "ecr:GetDownloadUrlForLayer",
   "ecr:GetRepositoryPolicy",
   "ecr:DescribeRepositories",
   "ecr:ListImages",
   "ecr:DescribeImages",
   "ecr:BatchGetImage"
  ],
  "Resource": "*"
 }]
}
```

Função do IAM da execução de tarefas do Amazon ECS

O agente de contêiner do Amazon ECS faz chamadas à API do Amazon ECS em seu nome, portanto, ele requer uma política e uma função do IAM para o serviço saber que o agente pertence a você. Essa função do IAM é chamada de função do IAM de execução de tarefa. Você pode ter várias funções de execução de tarefa para diferentes finalidades associadas à sua conta.

Veja a seguir os casos de uso comuns para uma função do IAM de execução de tarefa:

- Sua tarefa usa o tipo de execução Fargate e...
  - está extraindo uma imagem de contêiner do Amazon ECR.
  - usa o driver de log `awslogs`.
- Sua tarefa usa o tipo de execução EC2 ou Fargate e...
  - está usando a autenticação de registro privado. Para obter mais informações, consulte Permissões do IAM necessárias para a autenticação de registro privado (p. 230).
  - a definição de tarefa faz referência a dados confidenciais usando segredos do Secrets Manager ou parâmetros do Parameter Store do AWS Systems Manager. Para obter mais informações, consulte Permissões necessárias do IAM para segredos do Amazon ECS (p. 230).

  Note

  A função de execução de tarefa é aceita pelo agente de contêiner do Amazon ECS versão 1.16.0 e posterior.

O Amazon ECS fornece a seguinte política gerenciada `AmazonECSTaskExecutionRolePolicy` que contém as permissões que casos de uso comuns descritos acima exigem.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

A função de execução de tarefas do Amazon ECS é criada automaticamente para você na experiência de primeira execução do console do Amazon ECS. Porém, você deve associar manualmente a política gerenciada do IAM para as tarefas permitirem que o Amazon ECS adicione permissões para recursos e aprimoramentos futuros à medida que forem introduzidos. Você pode usar o procedimento a seguir para verificar e saber se a conta já tem a função de execução da tarefa do Amazon ECS e anexar a política do IAM caso necessário.

Para verificar **`ecsTaskExecutionRole`** no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure `ecsTaskExecutionRole` na lista de funções. Se a função não existir, use o procedimento a seguir para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
4. Escolha Permissions (Permissões). Verifique se a política gerenciada AmazonECSTaskExecutionRolePolicy está anexada à função. Se a política estiver anexada, isso significa que a função de execução de tarefa do Amazon ECS está configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para associar a política.

   a. Escolha Anexar política.
   b. Para restringir as políticas disponíveis a serem anexadas, em Filter (Filtro), digite AmazonECSTaskExecutionRolePolicy.
   c. Selecione a caixa à esquerda da política AmazonECSTaskExecutionRolePolicy e escolha Attach policy.
5. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
6. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

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

```
}
```

Para criar a função **`ecsTaskExecutionRole`** do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e depois Create Role.
3. Na seção Select type of trusted entity, escolha Elastic Container Service.
4. Em Select your use case (Selecionar seu caso de uso), escolha Elastic Container Service Task (Tarefa de serviço do contêiner elástico) e, em seguida, Next: Permissions (Próxima: Permissões).
5. Na seção Attach permissions policy (Anexar uma política ao usuário), procure AmazonECSTaskExecutionRolePolicy, selecione a política e, em seguida, escolha Next: Review (Próximo: análise).
6. Em Role Name, digite `ecsTaskExecutionRole` e escolha Create role.

Permissões do IAM necessárias para a autenticação de registro privado

A função de execução de tarefas do Amazon ECS é necessária para usar o recurso de autenticação de registro privado. Isso permite que o agente de contêiner obtenha a imagem do contêiner. Para obter mais informações, consulte Autenticação de registro privado para tarefas (p. 71).

Para dar acesso aos segredos criados por você, adicione manualmente as permissões a seguir como uma política em linha à função de execução da tarefa. Para obter mais informações, consulte Adicionar e remover políticas do IAM.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`—Só será exigido se a chave usar uma chave do KMS personalizada e não a chave padrão. O ARN da chave personalizada deve ser adicionado como um recurso.

Veja abaixo um exemplo de política em linha adicionando as permissões.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
        "arn:aws:kms:<region>:<aws_account_id>:key:key_id"
      ]
    }
  ]
}
```

Permissões necessárias do IAM para segredos do Amazon ECS

Para usar o recurso de segredos do Amazon ECS, você deve ter a função de execução de tarefas do Amazon ECS e fazer referência a ela na definição de tarefa. Isso permite que o agente de contêiner obtenha os recursos necessários do AWS Systems Manager ou do Secrets Manager. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

Para dar acesso aos parâmetros do Parameter Store do AWS Systems Manager criados por você, adicione manualmente as permissões a seguir como uma política em linha à função de execução da tarefa. Para obter mais informações, consulte Adicionar e remover políticas do IAM.

- `ssm:GetParameters` — Exigido se você estiver fazendo referência a um parâmetro do Repositório de parâmetros do Systems Manager em uma definição de tarefa.
- `secretsmanager:GetSecretValue` — Exigido se você estiver fazendo referência a um segredo do Secrets Manager diretamente ou se o parâmetro do Repositório de parâmetros do Systems Manager estiver fazendo referência a um segredo do Secrets Manager em uma definição de tarefa.
- `kms:Decrypt` — Exigido somente se o segredo usar uma chave personalizada do KMS e não a chave padrão. O ARN da chave personalizada deve ser adicionado como um recurso.

O exemplo de política em linha a seguir adiciona as permissões necessárias:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/parameter_name",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
        "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
      ]
    }
  ]
}
```

Permissões opcionais do IAM para tarefas Fargate que extraem imagens do Amazon ECR em endoints de interface

Ao executar tarefas que usam o tipo de execução Fargate que extraem imagens do Amazon ECR quando o Amazon ECR está configurado para usar um VPC endpoint, você pode restringir o acesso das tarefas a uma VPC ou VPC endpoint específico. Faça isso criando uma função de execução de tarefas que use chaves de condição do IAM.

Use as seguintes chaves de condição globais do IAM para restringir o acesso a uma VPC ou um VPC endpoint específico. Para obter mais informações, consulte Chaves de contexto de condição globais da AWS.

- `aws:SourceVpc`—restringe o acesso a uma VPC específica..
- `aws:SourceVpce`—restringe o acesso a um VPC endpoint específico..

A política da função de execução de tarefas a seguir fornece um exemplo para adicionar chaves de condição:

> **Important**
>
> A ação de API `ecr:GetAuthorizationToken` não pode ter as chaves de condição `aws:sourceVpc` e `aws:sourceVpce` aplicadas porque a chamada de API GetAuthorizationToken passa pela interface de rede elástica pertencente ao AWS Fargate em vez de pela interface de rede elástica da tarefa.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpce": "vpce-xxxxxx",
                    "aws:sourceVpc": "vpc-xxxxx"
                }
            }
        }
    ]
}
```

Função do IAM programador de serviço do Amazon ECS

O programador de serviço do Amazon ECS faz chamadas para as APIs do Amazon EC2 e do Elastic Load Balancing em seu nome a fim de registrar e cancelar o registro das instâncias de contêiner com os load balancers. Para anexar um load balancer a um serviço do Amazon ECS, você precisa criar uma função do IAM para os serviços a serem usados antes de iniciá-los. Este requisito se aplica a qualquer serviço do Amazon ECS que você pretende usar com um load balancer.

Na maioria dos casos, a função de serviço do Amazon ECS é criada para você automaticamente na primeira execução do console. Você pode usar o procedimento a seguir para verificar se a conta já tem a função de serviço do Amazon ECS.

A política `AmazonEC2ContainerServiceRole` é mostrada abaixo.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:Describe*",
        "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
        "elasticloadbalancing:DeregisterTargets",
        "elasticloadbalancing:Describe*",
        "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
        "elasticloadbalancing:RegisterTargets"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

Note

A regra `ec2:AuthorizeSecurityGroupIngress` é reservada para uso futuro. O Amazon ECS não atualiza automaticamente os grupos de segurança associados a load balancers do Elastic Load Balancing ou instâncias de contêiner do Amazon ECS.

Para verificar **`ecsServiceRole`** no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure `ecsServiceRole` na lista de funções. Se a função não existir, use o procedimento a seguir para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
4. Escolha a guia Permissions.
5. Na seção Políticas gerenciadas, certifique-se de que a política gerenciada AmazonEC2ContainerServiceRole esteja conectada à função. Se a política estiver anexada, sua função de serviço do Amazon ECS está configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para anexar a política.
   a. Escolha Attach Policy.
   b. Para restringir as políticas disponíveis a serem anexadas, em Filter (Filtro), digite AmazonEC2ContainerServiceRole.
   c. Selecione a caixa à esquerda da política AmazonEC2ContainerServiceRole e escolha Attach Policy.
6. Escolha Trust Relationships (Relacionamentos de confiança), Edit Trust Relationship (Editar relacionamento de confiança).
7. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Para criar uma função do IAM para os seus load balancers de programador de serviços

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e depois Create Role.
3. Na seção Select type of trusted entity, escolha Elastic Container Service.
4. Na seção Select your use case (Selecionar seu caso de uso), escolha Elastic Container Service (Serviço de contêiner elástico) e Next: Permissions (Próximo: permissões).

5. Na seção Attached permissions policy (Política de permissões anexadas), selecione a política AmazonEC2ContainerServiceRole e escolha Next: Review (Próximo: análise).
6. Para Role Name (Nome da função), digite `ecsServiceRole`, insira Role description (Descrição da função) e, em seguida, escolha Create role (Criar função).

Função do IAM CodeDeploy do Amazon ECS

Antes de poder usar o tipo de implantação azul/verde do CodeDeploy com o Amazon ECS, o serviço do CodeDeploy precisa de permissões para atualizar seu serviço do Amazon ECS em seu nome. Essas permissões são fornecidas pela função do IAM CodeDeploy (`ecsCodeDeployRole`).

> Note
>
> Os usuários do IAM também precisam de permissões para usar o CodeDeploy; essas permissões são descritas em Permissões do IAM exigidas para implantação azul/verde (p. 110).

Existem duas políticas gerenciadas fornecidas. A política `AWSCodeDeployRoleForECS`, mostrada abaixo, concede ao CodeDeploy permissão para atualizar qualquer recurso usando a ação associada.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ecs:DescribeServices",
                "ecs:CreateTaskSet",
                "ecs:UpdateServicePrimaryTaskSet",
                "ecs:DeleteTaskSet",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:ModifyListener",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:ModifyRule",
                "lambda:InvokeFunction",
                "cloudwatch:DescribeAlarms",
                "sns:Publish",
                "s3:GetObject",
                "s3:GetObjectMetadata",
                "s3:GetObjectVersion"
            ],
            "Resource": "*",
            "Effect": "Allow"
        }
    ]
}
```

A política `AWSCodeDeployRoleForECSLimited`, mostrada abaixo, concede ao CodeDeploy permissões mais limitadas.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ecs:DescribeServices",
                "ecs:CreateTaskSet",
                "ecs:UpdateServicePrimaryTaskSet",
                "ecs:DeleteTaskSet",
                "cloudwatch:DescribeAlarms"
```

```
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "sns:Publish"
            ],
            "Resource": "arn:aws:sns:*:*:CodeDeployTopic_*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:ModifyListener",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:ModifyRule"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": "arn:aws:lambda:*:*:function:CodeDeployHook_*",
            "Effect": "Allow"
        },
        {
            "Action": [
                "s3:GetObject",
                "s3:GetObjectMetadata",
                "s3:GetObjectVersion"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "s3:ExistingObjectTag/UseWithCodeDeploy": "true"
                }
            },
            "Effect": "Allow"
        }
    ]
}
```

Para criar uma função IAM do CodeDeploy

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles e depois Create Role.
3. Na seção Select type of trusted entity (Selecionar tipo de entidade confiável), escolha AWS service (Serviço da AWS).
4. Em Choose the service that will use this role (Selecionar o serviço que usará essa função), selecione CodeDeploy.
5. Em Select your use case (Selecionar seu caso de uso), escolha CodeDeploy, Next: Permissions (Próximo: permissões).
6. Selecione Next: Tags (Próximo: tags).
7. Em Add tags (optional) (Adicionar tags (opcional)), você pode adicionar tags do IAM opcionais à função. Selecione Next: Review (Próximo: revisão) ao concluir.
8. Em Role name (Nome da função), digite `ecsCodeDeployRole`, insira uma descrição opcional e, em seguida, escolha Create role.

Para adicionar as permissões necessárias à função do IAM CodeDeploy Amazon ECS

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. Procure `ecsCodeDeployRole` na lista de funções. Se a função não existir, use o procedimento acima para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
3. Na seção Permissions policies (Políticas de permissões), certifique-se de que a política gerenciada AWSCodeDeployRoleForECS ou AWSCodeDeployRoleForECSLimited esteja conectada à função. Se a política estiver anexada, sua função de serviço do Amazon ECS CodeDeploy estará configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para anexar a política.

   a. Escolha Attach policies (Anexar políticas).

   b. Para restringir as políticas disponíveis a serem anexadas, em Filter (Filtrar), digite AWSCodeDeployRoleForECS ou AWSCodeDeployRoleForECSLimited.

   c. Selecione a caixa à esquerda da política gerenciada da AWS e escolha Attach Policy (Anexar política).
4. Escolha Trust Relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
5. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

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

6. Se as tarefas no seu serviço do Amazon ECS que usam o tipo de implantação azul/verde exigirem o uso da função de execução de tarefa ou uma substituição de função de tarefa, você deverá adicionar a permissão `iam:PassRole` para cada função de execução de tarefa ou substituição de função de tarefa para a função do IAM do CodeDeploy como uma política em linha. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228) e Amazon ECS Task Role (Função da tarefa do Amazon ECS) (p. 241).

   Siga as subetapas abaixo para criar uma política em linha.

   a. Abra o console do IAM em https://console.aws.amazon.com/iam/.

   b. Procure `ecsCodeDeployRole` na lista de funções. Se a função não existir, use o procedimento acima para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.

   c. Na seção Permissions policies (Políticas de permissões), escolha Add inline policy (Adicionar política em linha).

   d. Selecione a guia JSON e adicione o seguinte texto da política.

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

```
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"
            ]
        }
    ]
}
```

Note

Especifique o ARN completo da função de execução de tarefa ou da substituição da função de tarefa.

e. Escolha Review policy (Revisar política)

f. Em Name (Nome), digite um nome para a política adicionada e escolha Create policy (Criar política).

Função do IAM Serviço Auto Scaling do Amazon ECS

Para usar Serviço Auto Scaling com o Amazon ECS, o serviço Application Auto Scaling precisa de permissão para descrever os alarmes do CloudWatch e os serviços registrados, bem como de permissão para atualizar a contagem desejada do serviço Amazon ECS em seu nome. Essas permissões são fornecidas pela função do Serviço Auto Scaling IAM (`ecsAutoscaleRole`).

Note

Os usuários do IAM também precisam de permissões para usar Serviço Auto Scaling; essas permissões são descritas em Serviço Auto Scaling Permissões obrigatórias do IAM (p. 129). Caso um usuário do IAM tenha as permissões obrigatórias para usar Serviço Auto Scaling no console do Amazon ECS, crie funções do IAM e anexe políticas de função do IAM a elas, e esse usuário poderá criar essa função automaticamente como parte dos fluxos de trabalho criar serviço (p.) ou atualizar serviço (p. 160) do console do Amazon ECS e usar a função em qualquer outro serviço depois (no console ou com AWS CLI ou SDKs).

Você pode usar o procedimento a seguir para verificar e saber se a conta já tem a função Serviço Auto Scaling IAM.

A política `AmazonEC2ContainerServiceAutoscaleRole` é mostrada abaixo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Stmt1456535218000",
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeServices",
                "ecs:UpdateService"
            ],
            "Resource": [
                "*"
            ]
        },
        {
            "Sid": "Stmt1456535243000",
            "Effect": "Allow",
            "Action": [
```

```
                "cloudwatch:DescribeAlarms"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Para verificar a função Serviço Auto Scaling no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure `ecsAutoscaleRole` na lista de funções. Se a função não existir, use o procedimento a seguir para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
4. Escolha a guia Permissions (Permissões).
5. Na seção Permissions policies (Políticas de permissões), certifique-se de que a política gerenciada AmazonEC2ContainerServiceAutoscaleRole esteja associada à função. Se a política estiver associada, sua função de serviço do Amazon ECS estará configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para associar a política.
   a. Escolha Attach policies (Associar políticas).
   b. Para restringir as políticas disponíveis a serem associadas, em Filter (Filtrar), digite `AmazonEC2ContainerServiceAutoscaleRole`.
   c. Selecione a caixa à esquerda da política AmazonEC2ContainerAutoscaleRole e escolha Attach Policy (Associar política).
6. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
7. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

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

Para criar uma função do IAM do Serviço Auto Scaling

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções) e Create role (Criar função).
3. Na seção Choose the service that will use this role (Escolher o serviço que usará esta função), escolha Elastic Container Service (Serviço de contêiner elástico).
4. Na seção Select your use case (Selecionar seu caso de uso), escolha Elastic Container Service Autoscale (Dimensionamento automático do Serviço de contêiner elástico) e Next: Permissions (Próximo: Permissões).

5. Para Add tags (optional) (Adicionar tags [adicional]), insira qualquer tag de valor de chave chave que você deseja adicionar à função do IAM. Escolha Next: Review (Próximo: Revisar) quando terminar.
6. No campo Role name (Nome da função), digite `ecsAutoscaleRole` para fornecer um nome à função e, depois, escolha Create Role (Criar função) para terminar.

Função do IAM Eventos do CloudWatch

Para usar tarefas programadas do Amazon ECS com regras e destinos do Eventos do CloudWatch, o serviço do Eventos do CloudWatch precisa de permissões para executar tarefas do Amazon ECS em seu nome. Essas permissões são fornecidas pela função do IAM Eventos do CloudWatch (`ecsEventsRole`).

A função Eventos do CloudWatch é criada automaticamente para você no Console de gerenciamento da AWS quando configurar uma tarefa programada. Para obter mais informações, consulte Tarefas programadas (`cron`) (p. 93).

A política `AmazonEC2ContainerServiceEventsRole` é mostrada abaixo.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

Se suas tarefas programadas exigirem o uso da função de execução da tarefa ou de uma substituição de função de tarefa, você deverá adicionar as permissões `iam:PassRole` a cada função de execução de tarefa ou substituição de função de tarefa para a função do IAM do Eventos do CloudWatch. Para obter mais informações sobre a função de execução de tarefas, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

> **Note**
>
> Especifique o ARN completo da função de execução de tarefa ou da substituição da função de tarefa.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"
            ]
        }
    ]
}
```

Você pode usar o procedimento a seguir para verificar se sua conta já tem a função Eventos do CloudWatch do IAM e se necessário, criá-la manualmente.

Para verificar a função Eventos do CloudWatch do IAM no console do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure `ecsEventsRole` na lista de funções. Se a função não existir, use o procedimento a seguir para criar a função. Se a função existir, selecione-a para visualizar as políticas anexadas.
4. Escolha Permissions (Permissões).
5. Na seção Permissions policies (Políticas de permissões), certifique-se de que a política gerenciada AmazonEC2ContainerServiceEventsRole esteja associada à função. Se a política estiver associada, sua função de serviço do Amazon ECS estará configurada corretamente. Caso contrário, siga as etapas secundárias abaixo para associar a política.
   a. Escolha Attach policies (Associar políticas).
   b. Para restringir as políticas disponíveis a serem anexadas, em Filter (Filtrar), digite `AmazonEC2ContainerServiceEventsRole`.
   c. Selecione a caixa à esquerda da política AmazonEC2ContainerServiceEventsRole e escolha Attach Policy (Associar política).
6. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
7. Verifique se o relacionamento de confiança contém a seguinte política. Se o relacionamento de confiança corresponder à política abaixo, escolha Cancel. Se o relacionamento de confiança não corresponder, copie a política para a janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política confiável).

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

Para criar uma função IAM do Eventos do CloudWatch

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Roles (Funções) e Create role (Criar função).
3. Na seção Select type of trusted entity, escolha Elastic Container Service. Em Select your use case, escolha Elastic Container Service Task. Escolha Next: Permissions (Próximo: Permissões).
4. Na seção Attach permissions policy (Associar política de permissões), selecione a política AmazonEC2ContainerServiceEventsRole e Next: Tags (Próximo: Tags).
5. Na seção Add tags (optional) (Adicionar tags [opcional]), insira todas as tags que você gostaria de associar à função e escolha Next: Review (Próximo: Revisar).
6. Em Role name (Nome da função), digite `ecsEventsRole` para atribuir um nome à função, insira, opcionalmente, uma descrição e, depois, escolha Create role (Criar função).
7. Revise as informações da sua função e escolha Create Role (Criar função).
8. Procure na lista de funções `ecsEventsRole` e selecione a função que você acabou de criar.

9. Escolha Trust relationships (Relacionamentos de confiança), Edit trust relationship (Editar relacionamento de confiança).
10. Substitua a relação de confiança existente pelo seguinte texto na janela Policy Document (Documento da política) e escolha Update Trust Policy (Atualizar política de confiança).

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

Para adicionar permissões à função de execução de tarefas da função Eventos do CloudWatch do IAM

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Policies, Create policy.
3. Selecione JSON, cole a política a seguir e depois escolha Review policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"
            ]
        }
    ]
}
```

4. Em Name, digite `AmazonECSEventsTaskExecutionRole`, insira, opcionalmente, uma descrição e, em seguida, escolha Create policy.
5. No painel de navegação, selecione Roles.
6. Procure `ecsEventsRole` na lista de funções e selecione a função para visualizar as políticas anexadas.
7. Escolha Anexar política.
8. Na seção Attach policy, selecione a política AmazonECSEventsTaskExecutionRole e escolha Attach policy.

Amazon ECS Task Role (Função da tarefa do Amazon ECS)

Para usar funções do IAM para tarefas, o Amazon ECS precisa de permissão para fazer chamadas a APIs da AWS em seu nome. Essas permissões são fornecidas pela função da tarefa do Amazon ECS.

Você pode criar uma função do IAM para cada definição de tarefa que precisa de permissão para chamar APIs da AWS. Basta criar uma política do IAM que defina quais permissões a tarefa deve ter e anexar essa política a uma função que use a política de relacionamento de confiança da função de tarefa do Amazon ECS. Para obter mais informações, consulte Como criar uma função e uma política do IAM para suas tarefas (p. 243).

A relação de confiança da função de tarefa do Amazon ECS é mostrada abaixo.

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

Funções do IAM para tarefas

Com as funções do IAM para tarefas do Amazon ECS, você pode especificar uma função do IAM que pode ser usada pelos contêineres em uma tarefa. Os aplicativos devem assinar suas solicitações à API da AWS com as credenciais da AWS, e este recurso fornece uma estratégia para gerenciar as credenciais que seus aplicativos usam, de forma semelhante ao modo que os perfis de instância do Amazon EC2 fornecem credenciais a instâncias do EC2. Em vez de criar e distribuir suas credenciais da AWS aos contêineres ou utilizar a função da instância do EC2, você pode associar uma função do IAM a uma definição de tarefa do ECS ou à operação da API `RunTask`. Os aplicativos nos contêineres da tarefa podem então usar o AWS SDK ou a CLI para fazer solicitações de API para serviços da AWS autorizados.

Você define a função do IAM a usar em suas definições de tarefa, ou pode usar um cancelamento `taskRoleArn` ao executar uma tarefa manualmente com a operação da API `RunTask`. O agente do Amazon ECS recebe uma mensagem de carga para iniciar a tarefa com campos adicionais que contêm as credenciais da função. O agente do Amazon ECS define o ID exclusivo da credencial da tarefa como um token de identificação e atualiza seu cache interno de credenciais para que o token de identificação da tarefa aponte para as credenciais da função que são recebidas na carga. O agente do Amazon ECS preenche a variável de ambiente `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` no objeto `Env` (disponível com o comando docker inspect ***`container_id`***) para todos os contêineres que pertencem a essa tarefa com o seguinte URI relativo: `/`*`credential_provider_version`*`/credentials?id=`*`task_credential_id`*.

> **Note**
>
> Quando você especifica uma função do IAM para uma tarefa, a AWS CLI ou outros SDKs nos contêineres dessa tarefa usam as credenciais da AWS fornecidas pela função da tarefa exclusivamente e não herdam mais quaisquer permissões do IAM da instância de contêiner.

De dentro de contêiner, você pode consultar as credenciais com o seguinte comando:

```
curl 169.254.170.2$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

Resultado:

```
{
```

```
    "AccessKeyId": "ACCESS_KEY_ID",
    "Expiration": "EXPIRATION_DATE",
    "RoleArn": "TASK_ROLE_ARN",
    "SecretAccessKey": "SECRET_ACCESS_KEY",
    "Token": "SECURITY_TOKEN_STRING"
}
```

Tópicos

- Benefícios de usar funções do IAM para tarefas (p. 243)
- Como criar uma função e uma política do IAM para suas tarefas (p. 243)
- Como usar o SDK compatível da AWS (p. 245)
- Como especificar uma função do IAM para suas tarefas (p. 245)

Benefícios de usar funções do IAM para tarefas

- Isolamento de credenciais: um contêiner só pode recuperar credenciais para a função do IAM definida na definição da tarefa à qual pertence; um contêiner nunca tem acesso a credenciais que são destinadas a outro contêiner que pertença a outra tarefa.
- Autorização: os contêiner não autorizados não podem acessar credenciais de função do IAM definidas para outras tarefas.
- Auditabilidade: o registro em log de acessos e eventos está disponível por meio do CloudTrail para garantir auditoria retrospectiva. As credenciais de tarefas têm um contexto `taskArn` anexado à sessão, assim os logs do CloudTrail mostram qual tarefa está usando qual função.

Como criar uma função e uma política do IAM para suas tarefas

Você deve criar uma política do IAM para suas tarefas que especifiquem as permissões que você gostaria que os contêineres em suas tarefas tivessem. Você tem várias maneiras de criar uma nova política de permissões do IAM. Você pode copiar uma política gerenciada pela AWS completa que já faça algo do que você está procurando e personalizá-la de acordo com seus requisitos específicos. Para obter mais informações, consulte Criar uma nova política no Guia do usuário do IAM.

Você também precisa criar uma função para que suas tarefas usem antes que você possa especificá-la nas definições de sua tarefa. Você pode criar a função usando a função de serviço Amazon Elastic Container Service Task Role (Função de tarefa do Amazon Elastic Container Service) no console do IAM. Então é possível anexar sua política do IAM específica à função que oferece aos contêineres em sua tarefa as permissões desejadas. Os procedimentos a seguir descrevem como fazer isso.

> **Note**
>
> Para visualizar o relacionamento de confiança desta função, consulte Amazon ECS Task Role (Função da tarefa do Amazon ECS) (p. 241).

Se você tiver várias definições ou serviços de tarefas que exigem permissões do IAM, você deve considerar a criação de uma função para cada definição ou serviço de tarefa específico com as permissões mínimas necessárias para que as tarefas operem, de modo que você possa minimizar o acesso que fornece para cada tarefa.

Para criar uma política do IAM para suas tarefas

Neste exemplo, criamos uma política para permitir o acesso somente leitura a um bucket do Amazon S3. Você pode armazenar as credenciais do banco de dados ou outros segredos nesse bucket, e os contêineres em sua tarefa podem ler as credenciais do bucket e carregá-los no seu aplicativo.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.

2. No painel de navegação, escolha Policies (Políticas) e, em seguida, selecione Create policy (Criar política).
3. Siga as etapas em uma das guias a seguir, que mostra como usar os editores JSON ou visuais.

Using the visual editor

1. Para Service, escolha S3.
2. Em Actions (Ações), expanda a opção Read (Leitura) e selecione GetObject.
3. Em Resources (Recursos), selecione Add ARN (Adicionar ARN) e digite o ARN completo do bucket do Amazon S3.
4. Escolha Revisar política.
5. Na seção Review policy (Examinar política), em Name (Nome), digite o nome exclusivo próprio, como `AmazonECSTaskS3BucketPolicy`.
6. Escolha Create policy (Criar política) para terminar.

Using the JSON editor

1. No campo Documento de política, cole a política a ser aplicada às suas tarefas. No exemplo a seguir é concedida a permissão para o *`bucket my-task-secrets-bucket`* do Amazon S3. Você pode modificar o documento de política de acordo com as suas necessidades específicas.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-task-secrets-bucket/*"
      ]
    }
  ]
}
```

2. Escolha Create Policy (Criar política).

Para criar uma função do IAM para suas tarefas

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, escolha Funções, Criar nova função.
3. Na seção Select Role Type (Selecionar tipo de função), para a função de serviço Amazon Elastic Container Service Task Role (Função de tarefa do Amazon Elastic Container Service), escolha Select (Selecionar).

   Note

   Para visualizar o relacionamento de confiança desta função, consulte Amazon ECS Task Role (Função da tarefa do Amazon ECS) (p. 241).
4. Na seção Attach Policy (Anexar política), selecione a política a ser usada para suas tarefas (neste exemplo `AmazonECSTaskS3BucketPolicy`) e, em seguida, escolha Next Step (Próxima etapa).
5. Em Role Name (Nome da função), digite um nome para a função. Para este exemplo, digite `AmazonECSTaskS3BucketRole` para definir um nome para a função e, em seguida, escolha Create Role (Criar função) para concluir.

Como usar o SDK compatível da AWS

O suporte para as funções do IAM para tarefas foi adicionado aos SDKs da AWS em 13 de julho de 2016. Os contêineres das tarefas devem usar uma versão do SDK da AWS criada a partir dessa data. Os SDKs da AWS que estão incluídos em gerenciadores de pacotes de distribuição do Linux podem não ser novos o suficiente para oferecer suporte a esse recurso.

Para garantir que você esteja usando um SDK compatível, siga as instruções de instalação do SDK preferencial em Ferramentas para Amazon Web Services quando estiver criando seus contêineres para obter a última versão.

Como especificar uma função do IAM para suas tarefas

Depois de ter criado uma função e anexado uma política para a função, você pode executar tarefas que assumam a função. Você tem várias opções para fazer isso:

- Especifique uma função do IAM para suas tarefas na definição da tarefa. Você pode criar uma definição de tarefa nova ou uma nova revisão de uma definição de tarefa existente e especificar a função que você criou anteriormente. Se você usar o console para criar sua definição de tarefa, escolha sua função do IAM no campo Task Role (Função da tarefa). Se você usa a AWS CLI ou os SDKs, especifique sua função de tarefa ARN usando o parâmetro `taskRoleArn`. Para obter mais informações, consulte Como criar uma definição de tarefa (p. 29).

  Note

  Essa opção é obrigatória se você quiser usar as funções de tarefas do IAM em um serviço do Amazon ECS.
- Especifique uma substituição de função de tarefa do IAM ao executar uma tarefa. Você pode especificar uma substituição de função de tarefa do IAM ao executar uma tarefa. Se você usar o console para executar sua tarefa, escolha Advanced Options (Opções avançadas), em seguida sua função do IAM no campo Task Role (Função de tarefa). Se você usa a AWS CLI ou os SDKs, especifique sua função de tarefa do ARN usando o parâmetro `taskRoleArn` no objeto JSON `overrides`. Para obter mais informações, consulte Tarefas em execução (p. 91).

  Note

  Além das permissões padrão do Amazon ECS necessárias para executar tarefas e serviços, os usuários do IAM também precisam de permissões `iam:PassRole` para usar funções do IAM para tarefas.

Solução de problemas de identidade e acesso do Amazon Elastic Container Service

Use as seguintes informações para ajudar a diagnosticar e corrigir problemas comuns que podem ser encontrados ao trabalhar com o Amazon ECS e o IAM.

Tópicos

- Não tenho autorização para executar uma ação no Amazon ECS (p. 246)
- Não estou autorizado a executar iam:PassRole (p. 246)
- Quero visualizar minhas chaves de acesso (p. 246)
- Sou administrador e desejo conceder acesso ao Amazon ECS para outros usuários. (p. 247)
- Quero permitir que as pessoas fora da minha conta da AWS acessem meus recursos do Amazon ECS (p. 247)

Não tenho autorização para executar uma ação no Amazon ECS

Se o Console de gerenciamento da AWS informar que você não está autorizado a executar uma ação, você deverá entrar em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seu nome de usuário e senha.

O exemplo de erro a seguir ocorre quando o usuário `mateojackson` do IAM tenta usar o console para visualizar detalhes sobre um *`widget`*, mas não tem as permissões `ecs:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ecs:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso *`my-example-widget`* usando a ação `ecs:`*`GetWidget`*.

Não estou autorizado a executar iam:PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, entre em contato com o administrador para obter assistência. O administrador é a pessoa que forneceu a você o seu nome de usuário e senha. Peça a essa pessoa para atualizar suas políticas para permitir que você passe uma função para o Amazon ECS.

Alguns serviços da AWS permitem que você passe uma função existente para o serviço, em vez de criar uma nova função de serviço ou função vinculada ao serviço. Para fazer isso, um usuário deve ter permissões para passar a função para o serviço.

O erro de exemplo a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta usar o console para executar uma ação no Amazon ECS. No entanto, a ação exige que o serviço tenha permissões concedidas por uma função de serviço. Mary não tem permissões para passar a função para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

Neste caso, Mary pede ao administrador para atualizar suas políticas para permitir que ela execute a ação `iam:PassRole`.

Quero visualizar minhas chaves de acesso

Depois de criar suas chaves de acesso de usuário do IAM, é possível visualizar seu ID de chave de acesso a qualquer momento. No entanto, você não pode visualizar sua chave de acesso secreta novamente. Se você perder sua chave secreta, crie um novo par de chaves de acesso.

As chaves de acesso consistem em duas partes: um ID de chave de acesso (por exemplo, `AKIAIOSFODNN7EXAMPLE`) e uma chave de acesso secreta (por exemplo, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). Como um nome de usuário e uma senha, você deve usar o ID da chave de acesso e a chave de acesso secreta em conjunto para autenticar suas solicitações. Gerencie suas chaves de acesso de forma tão segura quanto você gerencia seu nome de usuário e sua senha.

> **Important**
>
> Não forneça as chaves de acesso a terceiros, mesmo que seja para ajudar a encontrar seu ID de usuário canônico. Ao fazer isso, você pode dar a alguém acesso permanente à sua conta.

Ao criar um par de chaves de acesso, você é solicitado a guardar o ID da chave de acesso e a chave de acesso secreta em um local seguro. A chave de acesso secreta só está disponível no momento em que é criada. Se você perder sua chave de acesso secreta, você deverá adicionar novas chaves de acesso para seu usuário do IAM. Você pode ter no máximo duas chaves de acesso. Se você já tiver duas, você deverá

excluir um par de chaves para poder criar um novo. Para visualizar as instruções, consulte Gerenciar chaves de acesso no Guia do usuário do IAM.

Sou administrador e desejo conceder acesso ao Amazon ECS para outros usuários.

Para permitir que outros usuários acessem o Amazon ECS, é necessário criar uma entidade do IAM (usuário ou função) para a pessoa ou o aplicativo que precisa do acesso. Eles usarão as credenciais dessa entidade para acessar a AWS. Você deve anexar uma política à entidade que concede a eles as permissões corretas no Amazon ECS.

Para começar a usar imediatamente, consulte Criar os primeiros usuário e grupo delegados do IAM no Guia do usuário do IAM.

Quero permitir que as pessoas fora da minha conta da AWS acessem meus recursos do Amazon ECS

É possível criar uma função que os usuários de outras contas ou pessoas fora da sua organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir a função. Para serviços que oferecem suporte a políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso a seus recursos.

Para saber mais, consulte o seguinte:

- Para saber se o Amazon ECS oferece suporte a esses recursos, consulte Como o Amazon Elastic Container Service funciona com o IAM (p. 196).
- Para saber como conceder acesso aos seus recursos em todas as contas da AWS pertencentes a você, consulte Conceder acesso a um usuário do IAM em outra conta da AWS pertencente a você no Guia do usuário do IAM.
- Para saber como conceder acesso aos seus recursos para contas da AWS de terceiros, consulte Conceder acesso a contas da AWS pertencentes a terceiros no Guia do usuário do IAM.
- Para saber como fornecer acesso por meio de federação de identidades, consulte Fornecer acesso a usuários autenticados externamente (federação de identidades) no Guia do usuário do IAM.
- Para saber a diferença entre usar funções e políticas baseadas em recurso para acesso entre contas, consulte Como as funções do IAM diferem de políticas baseadas em recursos no Guia do usuário do IAM.

Como usar a interface de linha de comando do Amazon ECS

A interface de linha de comando (CLI) do Amazon Elastic Container Service (Amazon ECS) fornece comandos de alto nível para simplificar a criação, a atualização e o monitoramento de clusters e tarefas de um ambiente de desenvolvimento local. A CLI do Amazon ECS oferece suporte a arquivos do Docker Compose, uma especificação de código aberto popular para definição e execução de aplicativos para vários contêineres. Use a CLI do ECS como parte do ciclo diário de desenvolvimento e testes como uma alternativa ao Console de gerenciamento da AWS.

> **Important**
>
> No momento, a versão mais recente da CLI do Amazon ECS só é compatível com as versões principais da Sintaxe de arquivo do Docker Compose versões 1, 2 e 3. A versão especificada no arquivo Compose deve ser a string `"1"`, `"1.0"`, `"2"`, `"2.0"`, `"3"` ou `"3.0"`. As versões secundárias do Docker Compose não são compatíveis.

A versão mais recente da CLI do Amazon ECS é 1.16.0. Para consultar as notas de release, acesse Changelog.

> **Note**
>
> O código-fonte da CLI do Amazon ECS está disponível no GitHub. Incentivamos você a enviar solicitações de envio para alterações que você gostaria de ter incluído. Porém, o Amazon Web Services atualmente não oferece suporte à execução de cópias modificadas desse software.

Tópicos

- Como instalar a CLI do Amazon ECS (p. 248)
- Como configurar a CLI do Amazon ECS (p. 254)
- Migração de arquivos de configuração (p. 255)
- Tutorial: Como criar um cluster com uma tarefa Fargate usando a CLI do Amazon ECS (p. 256)
- Tutorial: Criação de um serviço do Amazon ECS que usa o Descoberta de serviço usando a CLI do Amazon ECS (p. 261)

Como instalar a CLI do Amazon ECS

Siga estas instruções para instalar a CLI do Amazon ECS no seu sistema macOS, Linux ou Windows.

Etapa 1: faça download da CLI do Amazon ECS

Faça download da CLI do Amazon ECS binário.

- Para macOS:

```
sudo curl -o /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest
```

- Para sistemas Linux:

```
sudo curl -o /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-
linux-amd64-latest
```

- Apenas para sistemas Windows:

  Abra o Windows PowerShell e execute os seguintes comandos:

```
PS C:\> New-Item 'C:\Program Files\Amazon\ECSCLI' -type
 directory
PS C:\> Invoke-WebRequest -OutFile 'C:\Program Files\Amazon\ECSCLI\ecs-cli.exe' https://
amazon-ecs-cli.s3.amazonaws.com/ecs-cli-windows-amd64-latest.exe
```

  Note

  Se você encontrar problemas de permissão, assegure-se de que você está executando o PowerShell como administrador.

Etapa 2: (opcional) verificar a CLI do Amazon ECS

Para verificar a validade do arquivo da CLI do Amazon ECS, você pode usar a soma MD5 ou as assinaturas PGP fornecidas. Ambos os métodos estão descritos nas seções a seguir.

Verificar usando a soma MD5

Verifique se o binário baixado com a soma MD5 fornecida.

- Para macOS (compare as duas strings de saída para verificar se são correspondentes):

```
curl -s https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest.md5 && md5 -
q /usr/local/bin/ecs-cli
```

- Para sistemas Linux (procure um `OK` na string de saída):

```
echo "$(curl -s https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-linux-amd64-latest.md5) /
usr/local/bin/ecs-cli" | md5sum -c -
```

- Apenas para sistemas Windows:

  Abra o Windows PowerShell e encontre o hash md5 do executável que você baixou:

```
PS C:\> Get-FileHash ecs-cli.exe -Algorithm MD5
```

  Compare isso com este hash md5:

```
PS C:\> Invoke-WebRequest -OutFile md5.txt https://
amazon-ecs-cli.s3.amazonaws.com/ecs-cli-windows-amd64-
latest.md5
PS C:\> Get-Content md5.txt
```

Verificar usando a assinatura PGP

Os executáveis da CLI do Amazon ECS são assinados de maneira criptografada, com assinaturas PGP. Você pode usar as etapas a seguir para verificar as assinaturas por meio da ferramenta GnuPG.

1. Faça download e instale a GnuPG. Para obter mais informações, consulte o Site do GnuPG.

   - Para macOS, recomendamos o uso do Homebrew. Instale o Homebrew seguindo as instruções disponíveis no site da ferramenta. Para obter mais informações, consulte Homebrew. Depois que o Homebrew estiver instalado, use o seguinte comando no seu terminal do MacOS:

     ```
     brew install gnupg
     ```

   - Para sistemas Linux, instale `gpg` usando o gerenciador de pacotes no seu tipo de Linux.
   - Para sistemas Windows, faça download e use o instalador simples do Windows no site do GnuPG. Para obter mais informações, consulte GnuPG Download.

2. Recupere a chave pública PGP do Amazon ECS. Você pode usar um comando para fazer isso ou pode criar a chave manualmente e depois importá-la.

   a. Opção 1: recupere a chave usando o seguinte comando.

      ```
      gpg --keyserver hkp://keys.gnupg.net --recv BCE9D9A42D51784F
      ```

   b. Opção 2: crie um arquivo com o seguinte conteúdo da chave pública PGP do Amazon ECS e o importe:

      ```
      -----BEGIN PGP PUBLIC KEY BLOCK-----
      Version: GnuPG v2

      mQINBFq1SasBEADliGcT1NVJ1ydfN8DqebYYe9ne3dt6jqKFmKowLmm6LLGJe7HU
      jGtqhCWRDkN+qPpHqdArRgDZAtn2pXY5fEipHgar4CP8QgRnRMO2fl74lmavr4Vg
      7K/KH8VHlq2uRw32/B94XLEgRbGTMdWFdKuxoPCttBQaMj3LGn6Pe+6xVWRkChQu
      BoQAhjBQ+bEm0kNy0LjNgjNlnL3UMAG56t8E3LANIgGgEnpNsB1UwfWluPoGZoTx
      N+6pHBJrKIL/1v/ETU4FXpYw2zvhWNahxeNRnoYj3uycHkeliCrw4kj0+skizBgO
      2K7oVX8Oc3j5+ZilhL/qDLXmUCb2az5cMM1mOoF8EKX5HaNuq1KfwJxqXE6NNIcO
      lFTrT7QwD5fMNld3FanLgv/ZnIrsSaqJOL6zRSq8O4LN1OWBVbndExk2Kr+5kFxn
      5lBPgfPgRj5hQ+KTHMa9Y8Z7yUc64BJiN6F9Nl7FJuSsfqbdkvRLsQRbcBG9qxX3
      rJAEhieJzVMEUNl+EgeCkxj5xuSkNU7zw2c3hQZqEcrADLV+hvFJktOz9Gm6xzbq
      lTnWWCz4xrIWtuEBA2qE+MlDheVd78a3gIsEaSTfQq0osYXaQbvlnSWOoc1y/5Zb
      zizHTJIhLtUyls9WisP2s0emeHZicVMfW61EgPrJAiupgc7kyZvFt4YwfwARAQAB
      tCRBbWF6b24gRUNTIDxlY3Mtc2VjdXJpdHlAYW1hem9uLmNvbT6JAhwEEAECAAYF
      AlrjL0YACgkQHivRXs0TaQrg1g/+JppwPqHnlVPmv7lessB8I5UqZeD6p6uVpHd7
      Bs3pcPp8BV7BdRbs3sPLt5bV1+rkqOlw+0gZ4Q/ue/YbWtOAt4qY0OcEo0HgcnaX
      lsB827QIfZIVtGWMhuh94xzm/SJkvngml6KB3YJNnWP61A9qJ37/VbVVLzvcmazA
      McWB4HUMNrhd0JgBCo0gIpqCbpJEvUc02Bjn23eEJsS9kC7OUAHyQkVnx4d9UzXF
      4OoISF6hmQKIBoLnRrAlj5Qvs3GhvHQ0ThYq0Grk/KMJJX2CSqt7tWJ8gk1n3H3Y
      SReRXJRnv7DsDDBwFgT6r5Q2HW1TBUvaoZy5hF6maD09nHcNnvBjqADzeT8Tr/Qu
      bBCLzkNSYqqkpgtwv7seoD2P4n1giRvDAOEfMZpVkUr+C252IaH1HZFEz+TvBVQM
      Y8OWWxmIJW+J6evjo3N1eO19UHv71jvoF8zljbI4bsL2c+QTJmOv7nRqzDQgCWyp
      Id/v2dUVVTk1j9omuLBBwNJzQCB+72LcIzJhYmaP1HC4LcKQG+/f41exuItenatK
      lEJQhYtyVXcBlh6Yn/wzNg2NWOwb3vqY/F7m6u9ixAwgtIMgPCDE4aJ86zrrXYFz
      N2HqkTSQh77Z8KPKmyGopsmN/reMuilPdINb249nA0dzoN+nj+tTFOYCIaLaFyjs
      Z0r1QAOJAjkEEwECACMFAlq1SasCGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIX
      gAAKCRC86dmkLVF4T9iFEACEnkm1dNXsWUx34R3c0vamHrPxvfkyI1FlEUen8D1h
      uX9xy6jCEROHWEp0rjGK4QDPgM93sWJ+s1UAKg214QRVzft0y9/DdR+twApA0fzy
      uavIthGd6+03jAAo6udYDE+cZC3P7XBbDiYEWk4XAF9I1JjB8hTZUgvXBL046JhG
      eM17+crgUyQeetkiOQemLbsbXQ40Bd9V7zf7XJraFd8VrwNUwNb+9KFtgAsc9rk+
      YIT/PEf+YOPysgcxI4sTWghtyCulVnuGoskgDv4v73PALU0ieUrvvQVqWMRvhVx1
      0X90J7cC1KOyhlEQQ1aFTgmQjmXexVTwIBm8LvysFK6YXM41KjOrlz3+6xBIm/qe
      bFyLUnf4WoiuOplAaJhK9pRY+XEnGNxdtN4D26Kd0F+PLkm3Tr3Hy3b1Ok34FlGr
      KVHUq1TZD7cvMnnNKEELTUcKX+1mV3an16nmAg/my1JSUt6BNK2rJpY1s/kkSGSE
      XQ4zuF2IGCpvBFhYAlt5Un5zwqkwwQR3/n2kwAoDzonJcehDw/C/cGos5D0aIU7I
      K2X2aTD3+pA7Mx3IMe2hqmYqRt9X42yF1PIEVRneBRJ3HDezAgJrNh0GQWRQkhIx
      gz6/cTR+ekr5TptVszS9few2GpI5bCgBKBisZIssT89aw7mAKWut0Gcm4qM9/yK6
      1bkCDQRatUmrARAAxNPvVwreJ2yAiFcUpdRlVhsuOgnxvs1QgsIw3H7+Pacr9Hpe
      8uftYZqdC82KeSKhpHq7c8gMTMucIINtH25x9BCc73E33EjCL9Lqov1TL7+QkgHe
      T+JIhZwdD8Mx2K+LVVVu/aWkNrfMuNwyDUciSI4D5QHa8T+F8fgN4OTpwYjirzel
      ```

```
5yoICMr9hVcbzDNv/ozKCxjx+XKgnFc3wrnDfJfntfDAT7ecwbUTL+viQKJ646s+
psiqXRYtVvYInEhLVrJ0aV6zHFoigE/Bils6/g7ru1Q6CEHqEw++APs5CcE8VzJu
WAGSVHZgun5Y9N4quR/M9Vm+IPMhTxrAg7rOvyRN9cAXfeSMf77I+XTifigNna8x
t/MOdjXr1fjF4pThEi5u6WsuRdFwjY2azEv3vevodTi4HoJReH6dFRa6y8c+UDgl
2iHiOKIpQqLbHEfQmHcDd2fix+AaJKMnPGNku9qCFEMbgSRJpXz6BfwnY1QuKE+I
R6jA0frUNt2jhiGG/F8RceXzohaaC/Cx7LUCUFWc0n7z32C9/Dtj7I1PMOacdZzz
bjJzRKO/ZDv+UN/c9dwAkllzAyPMwGBkUaY68EBstnIliW34aWm6IiHhxioVPKSp
VJfyiXPO0EXqujtHLAeChfjcns3I12YshT1dv2PafG53fp33ZdzeUgsBo+EAEQEA
AYkCHwQYAQIACQUCWrVJqwIbDAAKCRC86dmkLVF4T+ZdD/9x/8APzgNJF3o3STrF
jvnV1ycyhWYGAeBJiu7wjsNWwzMFOv15tLjB7AqeVxZn+WKDD/mIOQ45OZvnYZuy
X7DR0JszaH9wrYTxZLVruAu+t6UL0y/XQ4L1GZ9QR6+r+7t1Mvbfy7BlHbvX/gYt
Rwe/uwdibI0CagEzyX+2D3kTOlHO5XThbXaNf8AN8zha91Jt2Q2UR2X5T6JcwtMz
FBvZnl3LSmZyE0EQehS2iUurU4uWOpGppuqVnbi0jbCvCHKgDGrqZ0smKNAQng54
F365W3g8AfY48s8XQwzmcliowYX9bT8PZiEi0J4QmQh0aXkpqZyFefuWeOL2R94S
XKzr+gRh3BAULoqF+qK+IUMxTip9KTPNvYDpiC66yBiT6gFDji5Ca9pGpJXrC3xe
TXiKQ8DBWDhBPVPrruLIaenTtZEOsPc4I85yt5U9RoPTStcOr34s3w5yEaJagt6S
Gc5r9ysjkfH6+6rbi1ujxMgROSqtqr+RyB+V9A5/OgtNZc8llK6u4UoOCde8jUUW
vqWKvjJB/Kz3u4zaeNu2ZyyHaOqOuH+TETcW+jsY9IhbEzqN5yQYGi4pVmDkY5vu
lXbJnbqPKpRXgM9BecV9AMbPgbDq/5LnHJJXg+G8YQOgp4lR/hC1TEFdIp5wM8AK
CWsENyt2o1rjgMXiZOMF8A5oBLkCDQRatUuSARAAr77kj7j2QR2SZeOSlFBvV7oS
mFeSNnz9xZssqrsm6bTwSHM6YLDwc7Sdf2esDdyzONETwqrVCg+FxgL8hmo9hS4c
rR6tmrP0mOmptr+xLLsKcaP7ogIXsyZnrEAEsvW8PnfayoiPCdc3cMCR/lTnHFGA
7EuR/XLBmi7Qg9tByVYQ5Yj5wB9V4B2yeCt3XtzPqeLKvaxl7PNelaHGJQY/xo+m
V0bndxf9IY+4oFJ4blD32WqvyxESo7vW6WBh7oqv3Zbm0yQrr8a6mDBpqLkvWwNI
3kpJR974tg5o5LfDu1BeeyHWPSGm4U/G4JB+JIG1ADy+RmoWEt4BqTCZ/knnoGvw
D5sTCxbKdmuOmhGyTssoG+3OOcGYHV7pWYPhazKHMPm201xKCjH1RfzRULzGKjD+
yMLT1I3AXFmLmZJXikAOlvE3/wgMqCXscbycbLjLD/bXIuFWo3rzoezeXjgi/DJx
jKBAyBTYO5nMcth1O9oaFd9d0HbsOUDkIMnsgGBE766Piro6MHo0T0rXl07Tp4pI
rwuSOsc6XzCzdImj0Wc6axS/HeUKRXWdXJwno5awTwXKRJMXGfhCvSvbcbc2Wx+L
IKvmB7EB4K3fmjFFE67yolmiw2qRcUBfygtH3eL5XZU28MiCpue8Y8GKJoBAUyvf
KeM1rO8Jm3iRAc5a/D0AEQEAAYkEPgQYAQIACQUCWrVLkgIbAgIpCRC86dmkLVF4
T8FdIAQZAQIABgUCWrVLkgAKCRDePL1hra+LjtHYD/9MucxdFe6bXO1dQR4tKhhQ
P0LRqy6zlBY9ILCLowNdGZdqorogUiUymgn3VhEhVtxTOoHcN7qOuM01PNsRnOeS
EYjf8Xrb1clzkD6xULwmOclTb9bBxnBc/4PFvHAbZW3QzusaZniNgkuxt6BTfloS
Of4inq71kjmGK+TlzQ6mUMQUg228NUQC+a84EPqYyAeY1sgvgB7hJBhYL0QAxhcW
6m20Rd8iEc6HyzJ3yCOCsKip/nRWAbf0OvfHfRBp0+m0ZwnJM8cPRFjOqqzFpKH9
HpDmTrC4wKP1+TL52LyEqNh4yZitXmZNV7giSRIkk0eDSko+bFy6VbMzKUMkUJK3
D3eHFAMkujmbfJmSMTJOPGn5SB1HyjCZNx6bhIIbQyEUB9gKCmUFaqXKwKpF6rj0
iQXAJxLR/shZ5Rk96VxzOphUl7T90m/PnUEEPwq8KsBhnMRgxa0RFidDP+n9fgtv
HLmrOqX9zBCVXh0mdWYLrWvmzQFWzG7AoE55fkf8nAEPsalrCdtaNUBHRXA0OQxG
AHMOdJQQvBsmqMvuAdjkDWpFu5y0My5ddU+hiUzUyQLjL5Hhd5LOUDdewlZgIw1j
xrEAUzDKetnemM8GkHxDgg8koev5frmShJuce7vSjKpCNg3EIJSgqMOPFjJuLWtZ
vjHeDNbJy6uNL65ckJy6WhGjEADS2WAW1D6Tfekkc21SsIXk/LqEpLMR/0g5OUif
wcEN1rS9IJXBwIy8MelN9qr5KcKQLmfdfBNEyyceBhyVl0MDyHOKC+7PofMtkGBq
13QieRHv5GJ8LB3fclqHV8pwTTo3Bc8z2g0TjmUYAN/ixETdReDoKavWJYSE9yoM
aaJu279ioVTrwpECse0XkiRyKToTjwOb73CGkBZZpJyqux/rmCV/fp4ALdSW8zbz
FJVORaivhoWwzjpfQKhwcU9lABXi2UvVm14v0AfeI7oiJPSU1zM4fEny4oiIBXlR
zhFNih1UjIu82X16mTm3BwbIga/s1fnQRGzyhqUIMii+mWra23EwjChaxpvjjcUH
5ilLc5Zq781aCYRygYQw+hu5nFkOH1R+Z50Ubxjd/aqUfnGIAX7kPMD3Lof4KldD
Q8ppQriUvxVo+4nPV6rpTy/PyqCLWDjkguHpJsEFsMkwajrAz0QNSAU5CJ0G2Zu4
yxvYlumHCEl7nbFrm0vIiA75Sa8KnywTDsyZsu3XcOcf3g+g1xWTpjJqy2bYXlqz
9uDOWtArWHOis6bq8l9RE6xr1RBVXS6uqgQIZFBGyq66b0dIq4D2JdsUvgEMaHbc
e7tBfeB1CMBdA64e9Rq7bFR7Tvt8gasCZYlNr3lydh+dFHIEkH53HzQe6l88HEic
+0jVnLkCDQRa55wJARAAyLya2Lx6gyoWoJN1a6740q3o8e9d4KggQOfGMTCflmeq
ivuzgN+3DZHN+9ty2KxXMtn0mhHBerZdbNJyjMNT1gAgrhPNB4HtXBXum2wS57WK
DNmade914L7FWTPAWBG2Wn448OEHTqsClICXXWy9IICgclAEyIq0Yq5mAdTEgRJS
Z8t4GpwtDL9gNQyFXaWQmDmkAsCygQMvhAlmu9xOIzQG5CxSnZFk7zcuL60k14Z3
Cmt49k4T/7ZU8goWi8tt+rU78/IL3J/fF9+1civ1OwuUidgfPCSvOUW1JojsdCQA
L+RZJcoXq7lfOFj/eNjeOSstCTDPfTCL+kThE6E5neDtbQHBYkEX1BRiTedsV4+M
ucgiTrdQFWKf89G72xdv8ut9AYYQ2BbEYU+JAYhUH8rYYui2dHKJIgjNvJscuUWb
+QEqJIRleJRhrO+/CHgMs4fZAkWF1VFhKBkcKmEjLn1f7EJUUW84ZhKXjO/AUPX
1CHsNjziRceuJCJYox1cwsoq6jTE50GiNzcIxTn9xUc0UMKFeggNAFys1K+TDTm3
Bzo8H5ucjCUEmUm9lhkGwqTZgOlRX5eqPX+JBoSaObqhgqCa5IPinKRa6MgoFPHK
6sYKqroYwBGgZm6Js5chpNchvJMs/3WXNOEVg0J3z3vP0DMhxqWm+r+n9zlW8qsA
EQEAAYkEPgQYAQgACQUCWuecCQIbAgIpCRC86dmkLVF4T8FdIAQZAQgABgUCWuec
CQAKCRBQ3szEcQ5hr+ykD/4tOLRHFHXuKUcxgGaubUcVtsFrwBKma1cYjqaPms8u
```

```
6Sk0wfGRI32G/GhOrp0Ts/MOkbObq6VLTh8N5Yc/53MEl8zQFw9Y5AmRoW4PZXER
ujs5s7p4oR7xHMihMjCCBn1bvrR+34YPfgzTcgLiOEFHYT8UTxwnGmXOvNkMM7md
xD3CV5q6VAte8WKBo/220II3fcQlc9r/oWX4kXXkb0v9hoGwKbDJ1tzqTPrp/xFt
yohqnvImpnlz+Q9zXmbrWYL9/g8VCmW/NN2gju2G3Lu/TlFUWIT4v/5OPK6TdeNb
VKJO4+S8bTayqSG9CML1S57KSgCo5HUhQWeSNHI+fpe5oX6FALPT9JLDce8OZz1i
cZZ0MELP37mOOQun0AlmHm/hVzf0f311PtbzcqWaE51tJvgUR/nZFo6Ta3O5Ezhs
3VlEJNQ1Ijf/6DH87SxvAoRIARCuZd0qxBcDK0avpFzUtbJd24lRA3WJpkEiMqKv
RDVZkE4b6TW61f0o+LaVfK6E8oLpixegS4fiqC16mFrOdyRk+RJJfIUyz0WTDVmt
g0U1CO1ezokMSqkJ7724pyjr2xf/r9/sC6aOJwB/lKgZkJfC6NqL7TlxVA31dUga
LEOvEJTTE4gl+tYtfsCDvALCtqL0jduSkUo+RXcBItmXhA+tShW0pbS2Rtx/ixua
KohVD/0R4QxiSwQmICNtm9mw9ydIl1yjYXX5a9x4wMJracNY/LBybJPFnZnT4dYR
z4XjqysDwvvYZByaWoIe3QxjX84V6MlI2IdAT/xImu8gbaCI8tmyfpIrLnPKiR9D
VFYfGBXuAX7+HgPPSFtrHQONCALxxzlbNpS+zxt9r0MiLgcLyspWxSdmoYGZ6nQP
RO5Nm/ZVS+u2imPCRzNUZEMa+dlE6kHx0rS0dPiuJ4O7NtPeYDKkoQtNagspsDvh
cK7CSqAiKMq06UBTxqlTSRkm62eOCtcs3p3OeHu5GRZF1uzTET0ZxYkaPgdrQknx
ozjP5mC7X+45lcCfmcVt94TFNL5HwEUVJpmOgmzILCI8yoDTWzloo+i+fPFsXX4f
kynhE83mSEcr5VHFYrTY3mQXGmNJ3bCLuc/jq7ysGq69xiKmTlUeXFm+aojcRO5i
zyShIRJZ0GZfuzDYFDbMV9amA/YQGygLw//zP5ju5SW26dNxlf3MdFQE5JJ86rn9
MgZ4gcpazHEVUsbZsgkLizRp9imUiH8ymLqAXnfRGlU/LpNSefnvDFTtEIRcpOHc
bhayG0bk51Bd4mioOXnIsKy4j63nJXA27x5EVVHQ1sYRN8Ny4Fdr2tMAmj2O+X+J
qX2yy/UX5nSPU492e2CdZ1UhoU0SRFY3bxKHKB7SDbVeav+K5g==
=Gi5D
-----END PGP PUBLIC KEY BLOCK-----
```

Detalhes da chave pública PGP do Amazon ECS para referência:

```
Key ID: BCE9D9A42D51784F
Type: RSA
Size: 4096/4096
Expires: Never
User ID: Amazon ECS
Key fingerprint: F34C 3DDA E729 26B0 79BE AEC6 BCE9 D9A4 2D51 784F
```

Importe a chave pública PGP do Amazon ECS usando o seguinte comando.

```
gpg --import <public_key_filename>
```

3. Faça download das assinaturas da CLI do Amazon ECS. As assinaturas são assinaturas PGP desanexadas de caracteres ASCII armazenadas em arquivos com a extensão `.asc`. O arquivo de assinaturas tem o mesmo nome do executável correspondente, com `.asc` adicionado.

   - Para sistemas macOS:

     ```
     curl -o ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest.asc
     ```

   - Para sistemas Linux:

     ```
     curl -o ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-linux-amd64-latest.asc
     ```

   - Apenas para sistemas Windows:

     ```
     PS C:\> Invoke-WebRequest -OutFile ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-windows-amd64-latest.exe.asc
     ```

4. Verifique a assinatura.

   - Para sistemas macOS e Linux:

```
gpg --verify ecs-cli.asc /usr/local/bin/ecs-cli
```

- Apenas para sistemas Windows:

```
PS C:\> gpg --verify ecs-cli.asc 'C:\Program Files\Amazon\ECSCLI\ecs-cli.exe'
```

Saída esperada:

```
gpg: Signature made Tue Apr  3 13:29:30 2018 PDT
gpg:                using RSA key DE3CBD61ADAF8B8E
gpg: Good signature from "Amazon ECS <ecs-security@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: F34C 3DDA E729 26B0 79BE  AEC6 BCE9 D9A4 2D51 784F
     Subkey fingerprint: EB3D F841 E2C9 212A 2BD4  2232 DE3C BD61 ADAF 8B8E
```

Important

O aviso na saída é esperado e não é um problema. Isso ocorre porque não há uma cadeia de confiança entre a chave PGP pessoal (se você tiver uma) e a chave PGP do Amazon ECS. Para mais informações, consulte Web of trust.

Etapa 3: aplicar permissões de execução ao binário

Aplique permissões de execução ao binário.

- Para sistemas macOS e Linux:

```
sudo chmod +x /usr/local/bin/ecs-cli
```

- Apenas para sistemas Windows:

Edite as variáveis de ambiente e adicione `C:\Program Files\Amazon\ECSCLI` ao campo de variável `PATH`, separado das entradas existentes usando um ponto-e-vírgula. Por exemplo:

```
PS C:\> C:\existing\path;C:\Program Files\Amazon\ECSCLI
```

Reinicie o PowerShell (ou o prompt de comando) para que as alterações entrem em vigor.

Note

Assim que a variável `PATH` for definida, a CLI do Amazon ECS poderá ser usada pelo Windows PowerShell ou pelo prompt de comando.

Etapa 4: concluir a instalação

Verifique se a CLI está funcionando corretamente.

```
ecs-cli --version
```

Vá para Como configurar a CLI do Amazon ECS (p. 254).

Important

Você deve configurar a CLI do Amazon ECS com suas credenciais da AWS, uma região da AWS e um nome de cluster do Amazon ECS para poder usá-la.

Como configurar a CLI do Amazon ECS

A CLI do Amazon ECS requer algumas informações básicas sobre a configuração antes que você possa utilizá-la, como suas credenciais da AWS, a região da AWS em que criará seu cluster e o nome do cluster do Amazon ECS a ser usado. As informações de configuração são armazenadas no diretório `~/.ecs` dos sistemas macOS e Linux e em `C:\Users\<username>\AppData\local\ecs` nos sistemas Windows.

Para configurar a CLI do Amazon ECS

1. Configure um perfil da CLI com o comando a seguir, substituindo *profile_name* pelo nome de perfil desejado, variáveis de ambiente *$AWS_ACCESS_KEY_ID* e *$AWS_SECRET_ACCESS_KEY* pelas credenciais da AWS.

```
ecs-cli configure profile --profile-name profile_name --access-key $AWS_ACCESS_KEY_ID
 --secret-key $AWS_SECRET_ACCESS_KEY
```

2. Preencha a configuração com o seguinte comando, substituindo *launch_type* pelo tipo de inicialização de tarefa que deseja usar por padrão, *region_name* por sua região da AWS desejada, *cluster_name* pelo nome de um cluster existente ou um cluster novo ou existente do Amazon ECS a ser usado e *configuration_name* para o nome que você gostaria de fornecer a essa configuração.

```
ecs-cli configure --cluster cluster_name --default-launch-type launch_type --
region region_name --config-name configuration_name
```

Depois de instalar e configurar a CLI, você pode testar o Tutorial: Como criar um cluster com uma tarefa Fargate usando a CLI do Amazon ECS (p. 256). Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.

Perfis

A CLI do Amazon ECS é compatível com a configuração de vários conjuntos de credenciais da AWS, como perfis designados, usando o comando ecs-cli configure profile. Um perfil padrão pode ser definido usando o comando ecs-cli configure profile default. Esses perfis poderão então ser mencionados quando você executar comandos da CLI do Amazon ECS que exigem credenciais usando o indicador `--ecs-profile`; caso contrário, o perfil padrão será usado.

Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.

Configurações de cluster

Uma configuração de cluster é um conjunto de campos que descrevem um cluster do Amazon ECS, incluindo o nome do cluster e a região. Uma configuração padrão do cluster pode ser definida usando o comando ecs-cli configure default. A CLI do Amazon ECS é compatível com a configuração de várias configurações de cluster designado usando a opção `--config-name`.

Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.

Ordem de precedência

Há vários métodos para aprovar tanto as credenciais quanto a região em um comando de CLI do Amazon ECS. A lista a seguir é a ordem de precedência para cada um desses.

A ordem de precedência para credenciais é:

1. Indicadores de perfil da CLI do Amazon ECS:
   a. Perfil do ECS (`--ecs-profile`)
   b. Perfil do AWS (`--aws-profile`)
2. Variáveis de ambiente:
   a. `ECS_PROFILE`
   b. `AWS_PROFILE`
   c. `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, e `AWS_SESSION_TOKEN`
3. O config do ECS tenta buscar- as credenciais do perfil padrão do ECS.
4. O perfil padrão da AWS tenta usar credenciais— (`aws_access_key_id`, `aws_secret_access_key`) ou `assume_role` (`role_arn`, `source_profile`) do nome do perfil da AWS .
   a. Variável de ambiente `AWS_DEFAULT_PROFILE` (usa como padrão `default`).
5. Função da instância do EC2

A ordem de precedência para região é:

1. Indicadores da CLI do Amazon ECS:
   a. Indicador de região (`--region`)
   b. Indicador de config do cluster (`--cluster-config`)
2. A configuração do ECS- tenta buscar a região do perfil padrão do ECS.
3. Variáveis do ambiente—tentam buscar a região a partir das seguintes variáveis do ambiente:
   a. `AWS_REGION`
   b. `AWS_DEFAULT_REGION`
4. O perfil da AWS tenta -usar o nome do perfil da AWS :
   a. `AWS_PROFILE` variável de ambiente
   b. Variável de ambiente `AWS_DEFAULT_PROFILE` (usa como padrão `default`)

Migração de arquivos de configuração

O processo de configuração da CLI do Amazon ECS foi alterado significativamente na versão mais recente (v1.0.0), de forma a permitir a adição de novos recursos. Foi introduzido um comando de migração, que converte um arquivo de configuração mais antigo (versão v0.6.6 ou anterior) no formato atual. Os arquivos de configuração antigos estão obsoletos; recomendamos converter sua configuração para o mais novo formato para aproveitar os novos recursos. As alterações relacionadas à configuração e os novos recursos apresentados introduzidos na v1.0.0 dos novos arquivos de configuração formatados para YAML incluem:

- Divisão das informações de configuração relacionadas a credenciais e cluster em dois arquivos separados. As informações de credencial são armazenadas em `~/.ecs/credentials` e as informações de configuração do cluster são armazenadas em `~/.ecs/config`.
- Os arquivos de configuração são formatados em YAML.
- Suporte para armazenar configurações de vários nomes.
- A obsolescência do campo `compose-service-name-prefix` (nome usado para criar um serviço `<compose_service_name_prefix> + <project_name>`). Este campo ainda pode ser configurado.

No entanto, se não estiver configurado, não haverá mais um valor padrão atribuído. Para a CLI do Amazon ECS v0.6.6 e anteriores, o padrão era `ecscompose-service-`.

- Remoção do campo `compose-project-name-prefix` (nome usado para a criação de uma definição de tarefa `<compose_project_name_prefix> + <project_name>`). A CLI do Amazon ECS v1.0.0 e posteriores ainda podem ler os arquivos de configuração antigos; portanto, se esse campo estiver presente, ele continuará sendo lido e usado. No entanto, não há suporte para a configuração desse campo na v1.0.0+ com o comando `ecs-cli configure`, e se o campo for adicionado manualmente a um arquivo de configuração v1.0.0+, isso fará com que a CLI do Amazon ECS emita um erro.
- O campo `cfn-stack-name-prefix` (nome usado para criar pilhas de CFN `<cfn_stack_name_prefix> + <cluster_name>`) foi alterado para `cfn-stack-name`. Em vez de especificar um prefixo, pode ser configurado o nome exato de um modelo do CloudFormation.
- A CLI do Amazon ECS v0.6.6 e anteriores podem configurar credenciais usando um perfil da AWS designado do arquivo `~/.aws/credentials` no seu sistema. Essa funcionalidade foi removida. No entanto, foi adicionado um novo indicador, `--aws-profile`, que permite fazer referência a um perfil da AWS em linha com todos os comandos que exigem credenciais.

  Note

  O indicador `--project-name` pode ser usado para definir o nome do projeto.

Migração de arquivos de configuração mais antigos para o formato v1.0.0+

Embora todas as versões da CLI do Amazon ECS ofereçam suporte com relação ao formato de arquivo de configuração mais antigo, a atualização para o novo formato será necessária para aproveitar alguns novos recursos, como uso de vários perfis de cluster designados. A migração do seu arquivo de configuração existente para o novo formato é fácil com o comando `ecs-cli configure migrate`. O comando usa as informações de configuração armazenadas no formato antigo `~/.ecs/config` e as converte em um par de arquivos no novo formato, substituindo o arquivo de configuração antigo no processo.

Ao executar o comando `ecs-cli configure migrate`, é exibida uma mensagem de aviso com o antigo arquivo de configuração e uma pré-visualização dos novos arquivos de configuração. A confirmação do usuário é necessária antes que a migração prosseguir. Se o indicador `--force` for usado, a mensagem de aviso não será exibida e a migração prosseguirá sem nenhuma confirmação. Se o `cfn-stack-name-prefix` for usado no arquivo existente, `cfn-stack-name` será armazenado no novo arquivo como `<cfn_stack_name_prefix> + <cluster_name>`.

Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.

Tutorial: Como criar um cluster com uma tarefa Fargate usando a CLI do Amazon ECS

Este tutorial mostra como configurar um cluster e implantar um serviço com tarefas usando o tipo de execução Fargate.

Pré-requisitos

Conclua os seguintes pré-requisitos:

- Configure uma conta da AWS.
- Instale a CLI do Amazon ECS. Para obter mais informações, consulte Como instalar a CLI do Amazon ECS (p. 248).
- Instale e configure a AWS CLI. Para obter mais informações, consulte Interface da linha de comando da AWS.

Etapa 1: criar a função do IAM para execução de tarefas

O agente de contêiner do Amazon ECS faz chamadas às APIs da AWS em seu nome, portanto, ele requer uma política e uma função do IAM para o serviço saber que o agente pertence a você. Essa função do IAM é chamada de função do IAM de execução de tarefa. Se você já tiver uma função de execução de tarefa criada para usar, ignore esta etapa. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Para criar a execução de tarefa da função de IAM usando a AWS CLI

1. Crie um arquivo denominado `task-execution-assume-role.json` com o seguinte conteúdo:

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

2. Criar a função de execução da tarefa:

```
aws iam --region us-west-2 create-role --role-name ecsTaskExecutionRole --assume-role-
policy-document file://task-execution-assume-role.json
```

3. Anexe a política de função de execução de tarefa:

```
aws iam --region us-west-2 attach-role-policy --role-name ecsTaskExecutionRole --
policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

Etapa 2: configurar a CLI do Amazon ECS

A CLI do Amazon ECS requer credenciais para fazer solicitações de API em seu nome. Ela pode retirar credenciais das variáveis do ambiente, de um perfil da AWS ou de um perfil do Amazon ECS. Para obter mais informações, consulte Como configurar a CLI do Amazon ECS (p. 254).

Para criar uma configuração da CLI do Amazon ECS

1. Crie uma configuração de cluster para definir a região da AWS a ser usada, os prefixos de criação de recursos e o nome do cluster a ser usado com a CLI do Amazon ECS:

```
ecs-cli configure --cluster tutorial --default-launch-type FARGATE --config-
name tutorial --region us-west-2
```

2. Crie um perfil de CLI usando a chave de acesso e a chave secreta:

```
ecs-cli configure profile --access-key AWS_ACCESS_KEY_ID --secret-
key AWS_SECRET_ACCESS_KEY --profile-name tutorial-profile
```

Etapa 3: Criar um cluster e configurar o grupo de segurança

Para criar um cluster de ECS e grupo de segurança

1. Crie um cluster do Amazon ECS com o comando ecs-cli up. Desde que você especificou Fargate como seu tipo de inicialização padrão na configuração do cluster, este comando criará um cluster vazio e uma VPC configurada com duas sub-redes públicas.

   ```
   ecs-cli up --cluster-config tutorial --ecs-profile tutorial-profile
   ```

   Este comando pode levar alguns minutos para se concluído à medida que seus recursos são criados. A saída desse comando contém os IDs da sub-rede e da VPC que são criados. Anote esses IDs, pois serão usados posteriormente.
2. Usando a AWS CLI, recupere o ID do grupo de segurança padrão para a VPC. Use o ID da VPC da saída anterior:

   ```
   aws ec2 describe-security-groups --filters Name=vpc-id,Values=VPC_ID --region us-west-2
   ```

   A saída desse comando contém o ID do grupo de segurança, que será usado na próxima etapa.
3. Usando a AWS CLI, adicione uma regra de grupo de segurança para permitir acesso de entrada na porta 80:

   ```
   aws ec2 authorize-security-group-ingress --group-id security_group_id --protocol tcp --
   port 80 --cidr 0.0.0.0/0 --region us-west-2
   ```

Etapa 4: Criar um arquivo de composição

Para esta etapa, crie um arquivo de composição do Docker simples que crie um aplicativo web PHP. Neste momento, a CLI do Amazon ECS é compatível com a sintaxe do arquivo de composição do Docker versões 1, 2 e 3. Este tutorial usa o Docker compose v3.

Veja o arquivo de composição, que você pode chamar de `docker-compose.yml`. O contêiner `web` expõe a porta 80 para tráfego de entrada para o servidor da web. Ele também configura os logs do contêiner para acessarem o grupo de logs do CloudWatch criado anteriormente. Essa é a prática recomendada para tarefas do Fargate.

```
version: '3'
services:
  web:
    image: amazon/amazon-ecs-sample
```

```
    ports:
      - "80:80"
    logging:
      driver: awslogs
      options:
        awslogs-group: tutorial
        awslogs-region: us-west-2
        awslogs-stream-prefix: web
```

Note

Se sua conta já tiver um grupo de logs do CloudWatch Logs chamado `tutorial` na região `us-west-2`, escolha um nome exclusivo para que a CLI do ECS crie um novo grupo de logs para este tutorial.

Além de informações de composição do Docker, há alguns parâmetros específicos do Amazon ECS que você precisa especificar para o serviço. Usando os IDs da VPC, da sub-rede e do grupo de segurança da etapa anterior, crie um arquivo chamado `ecs-params.yml` com o seguinte conteúdo:

```
version: 1
task_definition:
  task_execution_role: ecsTaskExecutionRole
  ecs_network_mode: awsvpc
  task_size:
    mem_limit: 0.5GB
    cpu_limit: 256
run_params:
  network_configuration:
    awsvpc_configuration:
      subnets:
        - "subnet ID 1"
        - "subnet ID 2"
      security_groups:
        - "security group ID"
      assign_public_ip: ENABLED
```

Etapa 5: Implantar o arquivo de composição em um cluster

Depois de criar o arquivo de composição, você poderá implantá-lo no cluster com ecs-cli compose service up. Por padrão, o comando procura arquivos chamados `docker-compose.yml` e `ecs-params.yml` no diretório atual; você pode especificar outro arquivo de composição do docker com a opção `--file`, e um arquivo ECS Params diferente com a opção `--ecs-params`. Por padrão, os recursos criados por esse comando têm o diretório atual no título, mas você pode substituí-lo pela opção `--project-name`. A opção `--create-log-groups` cria os grupos de log do CloudWatch para logs de contêiner.

```
ecs-cli compose --project-name tutorial service up --create-log-groups --cluster-
config tutorial --ecs-profile tutorial-profile
```

Etapa 6: Visualizar os contêineres em execução em um cluster

Depois de implantar o arquivo de composição, você verá os contêineres em execução no serviço com ecs-cli compose service ps.

```
ecs-cli compose --project-name tutorial service ps --cluster-config tutorial --ecs-
profile tutorial-profile
```

Resultado:

```
Name                                                State    Ports
 TaskDefinition  Health
tutorial/0c2862e6e39e4eff92ca3e4f843c5b9a/web  RUNNING  34.222.202.55:80->80/tcp
 tutorial:1      UNKNOWN
```

No exemplo acima, você verá o contêiner `web` e de seu arquivo de composição, e também o endereço IP e a porta do servidor web. Se você colocar esse endereço em um navegador web, deverá visualizar o aplicativo web PHP. Além disso, na saída está o valor de `task-id` do contêiner. Copie o ID da tarefa; você poderá usá-lo na próxima etapa.

Etapa 7: Visualizar os logs do contêiner

Visualize os logs da tarefa:

```
ecs-cli logs --task-id 0c2862e6e39e4eff92ca3e4f843c5b9a --follow --cluster-config tutorial
 --ecs-profile tutorial-profile
```

> **Note**
>
> A opção `--follow` informa à CLI do Amazon ECS para pesquisar continuamente logs.

Etapa 8: Dimensionar as tarefas no cluster

Você pode escalar a contagem de tarefas para aumentar o número de instâncias do seu aplicativo com ecs-cli compose service scale. Neste exemplo, a contagem em execução do aplicativo é aumentada para dois.

```
ecs-cli compose --project-name tutorial service scale 2 --cluster-config tutorial --ecs-
profile tutorial-profile
```

Agora você deve visualizar mais dois contêineres em seu cluster:

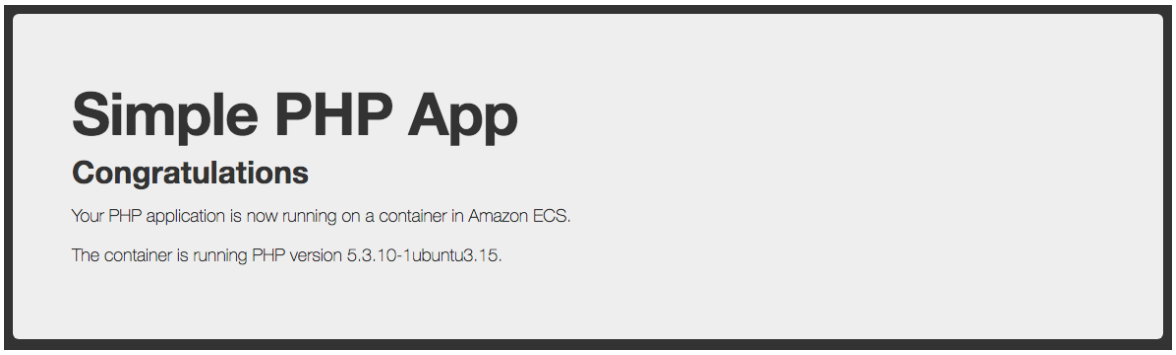
```
ecs-cli compose --project-name tutorial service ps --cluster-config tutorial --ecs-
profile tutorial-profile
```

Resultado:

```
Name                                                State    Ports
 TaskDefinition  Health
tutorial/0c2862e6e39e4eff92ca3e4f843c5b9a/web  RUNNING  34.222.202.55:80->80/tcp
 tutorial:1      UNKNOWN
tutorial/d9fbbc931d2e47ae928fcf433041648f/web  RUNNING  34.220.230.191:80->80/tcp
 tutorial:1      UNKNOWN
```

Etapa 9: (Opcional) Visualizar seu aplicativo web

Insira o endereço IP para a tarefa no navegador da web. Feito isso, você deverá ver uma página da web que exibe o aplicativo web Simple PHP App.



Etapa 10: Limpeza

Ao concluir este tutorial, você deve limpar seus recursos para que não incorram em quaisquer cobranças adicionais. Primeiro, exclua o serviço para que pare os contêineres existentes e não tente executar nenhuma outra tarefa.

```
ecs-cli compose --project-name tutorial service down --cluster-config tutorial --ecs-profile tutorial-profile
```

Agora, deixe seu cluster inativo, o que limpa os recursos que você criou anteriormente com o comando ecs-cli up.

```
ecs-cli down --force --cluster-config tutorial --ecs-profile tutorial-profile
```

Tutorial: Criação de um serviço do Amazon ECS que usa o Descoberta de serviço usando a CLI do Amazon ECS

Esse tutorial mostra uma explicação simples da criação de um serviço do Amazon ECS que é configurado para usar o descoberta de serviço. Muitos dos valores de configuração do descoberta de serviço podem ser especificados com o arquivo de parâmetros ou os sinalizadores do ECS. Quando os sinalizadores forem usados, eles têm precedência sobre o arquivo de parâmetros do ECS, se os dois estiverem presentes. Ao usar a CLI do Amazon ECS, o nome do projeto de composição é usado como o nome do seu serviço do ECS.

Pré-requisitos

Conclua os seguintes pré-requisitos antes de continuar:

- Configure uma conta da AWS.
- Instale a CLI do Amazon ECS. Para obter mais informações, consulte Como instalar a CLI do Amazon ECS (p. 248).

Configurar a CLI do Amazon ECS

Antes que você possa iniciar este tutorial, você deve instalar e configurar a CLI do Amazon ECS. Para obter mais informações, consulte Como instalar a CLI do Amazon ECS (p. 248).

A CLI do Amazon ECS requer credenciais para fazer solicitações de API em seu nome. Ela pode retirar credenciais das variáveis do ambiente, de um perfil da AWS ou de um perfil do Amazon ECS. Para obter mais informações, consulte Como configurar a CLI do Amazon ECS (p. 254).

Para criar uma configuração da CLI do Amazon ECS

1. Crie uma configuração do cluster:

```
ecs-cli configure --cluster ec2-tutorial --region us-east-1 --default-launch-type EC2
 --config-name ec2-tutorial
```

2. Crie um perfil usando a chave de acesso e a chave secreta:

```
ecs-cli configure profile --access-key AWS_ACCESS_KEY_ID --secret-
key AWS_SECRET_ACCESS_KEY --profile-name ec2-tutorial
```

> **Note**
>
> Se esta for a primeira vez que você estiver configurando a CLI do Amazon ECS, essas configurações serão marcadas como padrão. Se essa não for a primeira vez que você configura a CLI do Amazon ECS, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide para definir isso como configuração e perfil padrão.

Criar um serviço do Amazon ECS configurado para usar o Descoberta de serviço

Use as seguintes etapas para criar um serviço do Amazon ECS que esteja configurado para usar o descoberta de serviço com a CLI do Amazon ECS.

Para criar um serviço do Amazon ECS configurado para usar o descoberta de serviço

1. Crie um serviço do Amazon ECS chamado `backend` e crie um namespace DNS privado chamado `tutorial` dentro de uma VPC. Neste exemplo, a tarefa está usando o modo de rede `awsvpc`, portanto, os valores `container_name` e `container_port` não são necessários.

```
ecs-cli compose --project-name backend service up --private-dns-namespace tutorial --
vpc vpc-04deee8176dce7d7d --enable-service-discovery
```

Resultado:

```
INFO[0001] Using ECS task definition                     TaskDefinition="backend:1"
INFO[0002] Waiting for the private DNS namespace to be created...
INFO[0002] Cloudformation stack status                   stackStatus=CREATE_IN_PROGRESS
WARN[0033] Defaulting DNS Type to A because network mode was awsvpc
INFO[0033] Waiting for the Service Discovery Service to be created...
INFO[0034] Cloudformation stack status                   stackStatus=CREATE_IN_PROGRESS
INFO[0065] Created an ECS service                        service=backend
 taskDefinition="backend:1"
INFO[0066] Updated ECS service successfully              desiredCount=1
 serviceName=backend
INFO[0081] (service backend) has started 1 tasks: (task 824b5a76-8f9c-4beb-
a64b-6904e320630e).  timestamp="2018-09-12 00:00:26 +0000 UTC"
INFO[0157] Service status                                desiredCount=1 runningCount=1
 serviceName=backend
```

```
INFO[0157] ECS Service has reached a stable state        desiredCount=1 runningCount=1
 serviceName=backend
```

2. Crie outro serviço chamado `frontend` no mesmo namespace DNS privado. Como o namespace já existe, a CLI do Amazon ECS usa-o em vez de criar um novo.

```
ecs-cli compose --project-name frontend service up --private-dns-namespace tutorial --
vpc vpc-04deee8176dce7d7d --enable-service-discovery
```

Resultado:

```
INFO[0001] Using ECS task definition                     TaskDefinition="frontend:1"
INFO[0002] Using existing namespace ns-kvhnzhb5vxplfmls
WARN[0033] Defaulting DNS Type to A because network mode was awsvpc
INFO[0033] Waiting for the Service Discovery Service to be created...
INFO[0034] Cloudformation stack status                   stackStatus=CREATE_IN_PROGRESS
INFO[0065] Created an ECS service                        service=frontend
 taskDefinition="frontend:1"
INFO[0066] Updated ECS service successfully              desiredCount=1
 serviceName=frontend
INFO[0081] (service frontend) has started 1 tasks: (task 824b5a76-8f9c-4beb-
a64b-6904e320630e).  timestamp="2018-09-12 00:00:26 +0000 UTC"
INFO[0157] Service status                                desiredCount=1 runningCount=1
 serviceName=frontend
INFO[0157] ECS Service has reached a stable state        desiredCount=1 runningCount=1
 serviceName=frontend
```

3. Verifique se os dois serviços podem descobrir um ao outro dentro da VPC usando o DNS. O nome de host do DNS usa o seguinte formato: `<service_discovery_service_name>.<service_discovery_namespace>` Para este exemplo, o serviço `frontend` pode ser descoberto no `frontend.tutorial` e o serviço `backend` pode ser descoberto no `backend.tutorial`. Como esses são namespaces DNS privados, esses nomes DNS só são resolvidos quando dentro da VPC especificada.
4. Para atualizar as configurações do descoberta de serviço, atualize as configurações do serviço `frontend`. Os valores que podem ser atualizados são o DNS TTL e o valor do limite de falha de configuração personalizada da verificação de integridade.

```
ecs-cli compose --project-name frontend service up --update-service-discovery --dns-
type SRV --dns-ttl 120 --healthcheck-custom-config-failure-threshold 2
```

Resultado:

```
INFO[0001] Using ECS task definition                     TaskDefinition="frontend:1"
INFO[0001] Updated ECS service successfully              desiredCount=1
 serviceName=frontend
INFO[0001] Service status                                desiredCount=1 runningCount=1
 serviceName=frontend
INFO[0001] ECS Service has reached a stable state        desiredCount=1 runningCount=1
 serviceName=frontend
INFO[0002] Waiting for your Service Discovery resources to be updated...
INFO[0002] Cloudformation stack status                   stackStatus=UPDATE_IN_PROGRESS
```

5. Para limpar, exclua o serviço do Amazon ECS e os recursos do descoberta de serviço. Quando o serviço `frontend` é excluído, a CLI do Amazon ECS remove automaticamente o serviço do descoberta de serviço associado.

```
ecs-cli compose --project-name frontend service rm
```

```
INFO[0000] Updated ECS service successfully              desiredCount=0
 serviceName=frontend
INFO[0001] Service status                                desiredCount=0 runningCount=1
 serviceName=frontend
INFO[0016] Service status                                desiredCount=0 runningCount=0
 serviceName=frontend
INFO[0016] (service frontend) has stopped 1 running tasks: (task 824b5a76-8f9c-4beb-
a64b-6904e320630e).  timestamp="2018-09-12 00:37:25 +0000 UTC"
INFO[0016] ECS Service has reached a stable state        desiredCount=0 runningCount=0
 serviceName=frontend
INFO[0016] Deleted ECS service                           service=frontend
INFO[0016] ECS Service has reached a stable state        desiredCount=0 runningCount=0
 serviceName=frontend
INFO[0027] Waiting for your Service Discovery Service resource to be deleted...
INFO[0027] Cloudformation stack status                   stackStatus=DELETE_IN_PROGRESS
```

6. Para concluir a limpeza, exclua o serviço `backend` junto com o namespace DNS privado que foi criado com ele. A CLI do Amazon ECS associa a pilha do AWS CloudFormation para o namespace DNS privado ao serviço do Amazon ECS para o qual foi criada. Quando o serviço é excluído, o namespace também é excluído.

```
ecs-cli compose --project-name backend service rm --delete-namespace
```

Usar a AWS CLI com o Amazon ECS

A interface de linha de comando (CLI) da AWS é uma ferramenta unificada para gerenciar os serviços da AWS. Com apenas uma ferramenta para baixar e configurar, você pode controlar vários serviços da AWS pela linha de comando e automatizá-los por meio de scripts. Para obter mais informações sobre a AWS CLI, consulte http://aws.amazon.com/cli/.

Para obter mais informações sobre as outras ferramentas disponíveis para gerenciar os recursos da AWS, inclusive os diferentes SDKs da AWS, os toolkits do IDE e as ferramentas da linha de comando do Windows PowerShell, consulte http://aws.amazon.com/tools/.

As etapas a seguir o ajudarão a configurar um cluster do Amazon ECS usando tarefa Fargate :

Tópicos

- Tutorial: criar um cluster com uma tarefa do Fargate usando a AWS CLI (p. 265)

Tutorial: criar um cluster com uma tarefa do Fargate usando a AWS CLI

As etapas a seguir ajudarão você a configurar um cluster, registrar uma definição de tarefa, executar uma tarefa e realizar outros cenários comuns no Amazon ECS com a AWS CLI. Você deve usar a versão mais recente da AWS CLI. Para obter mais informações sobre como atualizar para a versão mais recente, consulte Instalar a interface de linha de comando da AWS.

Tópicos

- Pré-requisitos (p. 265)
- Etapa 1: (Opcional) Criar um cluster (p. 266)
- Etapa 2: Registrar uma definição de tarefa (p. 266)
- Etapa 3: Listar definições de tarefa (p. 268)
- Etapa 4: Criar um serviço (p. 268)
- Etapa 5: Listar serviços (p. 269)
- Etapa 6: Descrever o serviço em execução (p. 270)

Pré-requisitos

Este tutorial pressupõe que os seguintes pré-requisitos foram concluídos:

- A versão mais recente da AWS CLI está instalada e configurada. Para obter mais informações sobre como instalar ou fazer o upgrade do seu AWS CLI, consulte Instalar a interface de linha do comando da AWS.
- As etapas em Configuração com o Amazon ECS (p. 7) foram concluídas.
- Seu usuário da AWS tem as permissões necessárias especificadas no exemplo de política Permissões do assistente de primeira execução do Amazon ECS (p. 202) do IAM.
- Você tem uma VPC e um grupo de segurança criados para uso. Para obter mais informações, consulte Tutorial: como criar uma VPC com sub-redes públicas e privadas para seus clusters.

Etapa 1: (Opcional) Criar um cluster

Por padrão, sua conta recebe um cluster `default`.

> **Note**
>
> O benefício de usar o cluster `default` fornecido é que você não precisa especificar a opção `--cluster` *`cluster_name`* nos comandos subsequentes. Caso você crie o próprio cluster, não padrão, é necessário especificar `--cluster` *`cluster_name`* para cada comando que deseja usar com esse cluster.

Crie o próprio cluster com um nome exclusivo usando o seguinte comando:

```
aws ecs create-cluster --cluster-name fargate-cluster
```

Resultado:

```
{
    "cluster": {
        "status": "ACTIVE",
        "statistics": [],
        "clusterName": "fargate-cluster",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster"
    }
}
```

Etapa 2: Registrar uma definição de tarefa

Para executar uma tarefa no cluster do ECS, você deve registrar uma definição de tarefa. As definições de tarefa são listas de contêineres agrupados. O exemplo a seguir é uma definição de tarefa simples que cria um aplicativo web em PHP. Para obter mais informações sobre os parâmetros de definição de tarefa disponíveis, consulte Definições de tarefa do Amazon ECS (p. 25).

```
{
    "family": "sample-fargate",
    "networkMode": "awsvpc",
    "containerDefinitions": [
        {
            "name": "fargate-app",
            "image": "httpd:2.4",
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": [
                "sh",
  "-c"
            ],
            "command": [
                "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div
```

```
 style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!
</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></
html>' >  /usr/local/apache2/htdocs/index.html && httpd-foreground\""
            ]
        }
    ],
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "256",
    "memory": "512"
}
```

O exemplo de JSON acima pode ser passado para a AWS CLI de duas maneiras: você pode salvar o JSON de definição de tarefa como um arquivo e passá-lo com a opção `--cli-input-json file://path_to_file.json`. Ou, você pode ignorar as aspas no JSON e passar as definições do contêiner do JSON na linha de comando, como no exemplo abaixo. Caso você opte por passar as definições de contêiner na linha de comando, o comando ainda exige um parâmetro `--family` usado para manter várias versões da definição de tarefa associadas entre si.

Para usar um arquivo JSON para definições de contêiner:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/fargate-task.json
```

O register-task-definition retornará uma descrição da definição de tarefa depois de concluir o registro.

```
{
    "taskDefinition": {
        "status": "ACTIVE",
        "networkMode": "awsvpc",
        "family": "sample-fargate",
        "placementConstraints": [],
        "requiresAttributes": [
            {
                "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
            },
            {
                "name": "ecs.capability.task-eni"
            }
        ],
        "cpu": "256",
        "compatibilities": [
            "EC2",
            "FARGATE"
        ],
        "volumes": [],
        "memory": "512",
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "taskDefinitionArn": "arn:aws:ecs:region:aws_account_id:task-definition/sample-
fargate:2",
        "containerDefinitions": [
            {
                "environment": [],
                "name": "fargate-app",
                "mountPoints": [],
                "image": "httpd:2.4",
                "cpu": 0,
                "portMappings": [
                    {
                        "protocol": "tcp",
                        "containerPort": 80,
```

```
                    "hostPort": 80
                }
            ],
            "entryPoint": [
                "sh",
                "-c"
            ],
            "command": [
                "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div
 style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!
</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></
html>' >  /usr/local/apache2/htdocs/index.html && httpd-foreground\""
            ],
            "essential": true,
            "volumesFrom": []
        }
    ],
    "revision": 2
  }
}
```

Etapa 3: Listar definições de tarefa

Você pode listar as definições de tarefa para a conta a qualquer momento com o comando list-task-definitions. A saída desse comando mostra os valores `family` e `revision` valores que você pode usar juntos ao chamar run-task ou start-task.

```
aws ecs list-task-definitions
```

Resultado:

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate:1",
        "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate:2"
    ]
}
```

Etapa 4: Criar um serviço

Depois de você registrar uma tarefa para sua conta, poderá criar um serviço para a tarefa registrada no seu cluster. Para este exemplo, você cria um serviço no qual pelo menos duas instâncias da definição de tarefa `sample-fargate:1` são mantidas em execução no seu cluster.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service --
task-definition sample-fargate:1 --desired-count 2 --launch-type "FARGATE" --network-
configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-
abcd1234]}"
```

Resultado:

```
{
    "service": {
        "status": "ACTIVE",
        "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/sample-
fargate:1",
```

```
        "pendingCount": 0,
        "launchType": "FARGATE",
        "loadBalancers": [],
        "roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/ecs.amazonaws.com/
AWSServiceRoleForECS",
        "placementConstraints": [],
        "createdAt": 1510811361.128,
        "desiredCount": 2,
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "subnets": [
                    "subnet-abcd1234"
                ],
                "securityGroups": [
                    "sg-abcd1234"
                ],
                "assignPublicIp": "DISABLED"
            }
        },
        "platformVersion": "LATEST",
        "serviceName": "fargate-service",
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
        "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
        "deploymentConfiguration": {
            "maximumPercent": 200,
            "minimumHealthyPercent": 100
        },
        "deployments": [
            {
                "status": "PRIMARY",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "subnets": [
                            "subnet-abcd1234"
                        ],
                        "securityGroups": [
                            "sg-abcd1234"
                        ],
                        "assignPublicIp": "DISABLED"
                    }
                },
                "pendingCount": 0,
                "launchType": "FARGATE",
                "createdAt": 1510811361.128,
                "desiredCount": 2,
                "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
sample-fargate:1",
                "updatedAt": 1510811361.128,
                "platformVersion": "0.0.1",
                "id": "ecs-svc/9223370526043414679",
                "runningCount": 0
            }
        ],
        "events": [],
        "runningCount": 0,
        "placementStrategy": []
    }
}
```

Etapa 5: Listar serviços

Liste os serviços para o seu cluster. Você deve ver o serviço que criou na seção anterior. Você pode utilizar o nome do serviço ou o ARN completo retornado por esse comando e usá-lo para descrever o serviço depois.

```
aws ecs list-services --cluster fargate-cluster
```

Resultado:

```
{
    "serviceArns": [
        "arn:aws:ecs:region:aws_account_id:service/fargate-service"
    ]
}
```

Etapa 6: Descrever o serviço em execução

Descreva o serviço usando o nome do serviço recuperado anteriormente para obter mais informações sobre a tarefa.

```
aws ecs describe-services --cluster fargate-cluster --services fargate-service
```

Resultado:

```
{
    "services": [
        {
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/sample-
fargate:1",
            "pendingCount": 2,
            "launchType": "FARGATE",
            "loadBalancers": [],
            "roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
            "placementConstraints": [],
            "createdAt": 1510811361.128,
            "desiredCount": 2,
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-abcd1234"
                    ],
                    "securityGroups": [
                        "sg-abcd1234"
                    ],
                    "assignPublicIp": "DISABLED"
                }
            },
            "platformVersion": "LATEST",
            "serviceName": "fargate-service",
            "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
            "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
            "deploymentConfiguration": {
                "maximumPercent": 200,
                "minimumHealthyPercent": 100
            },
            "deployments": [
                {
                    "status": "PRIMARY",
                    "networkConfiguration": {
                        "awsvpcConfiguration": {
                            "subnets": [
                                "subnet-abcd1234"
                            ],
```

```
                            "securityGroups": [
                                "sg-abcd1234"
                            ],
                            "assignPublicIp": "DISABLED"
                        }
                    },
                    "pendingCount": 2,
                    "launchType": "FARGATE",
                    "createdAt": 1510811361.128,
                    "desiredCount": 2,
                    "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
sample-fargate:1",
                    "updatedAt": 1510811361.128,
                    "platformVersion": "0.0.1",
                    "id": "ecs-svc/9223370526043414679",
                    "runningCount": 0
                }
            ],
            "events": [
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 53c0de40-ea3b-489f-a352-623bf1235f08) (task d0aec985-901b-488f-9fb4-61b991b332a3).",
                    "id": "92b8443e-67fb-4886-880c-07e73383ea83",
                    "createdAt": 1510811841.408
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 b4911bee-7203-4113-99d4-e89ba457c626) (task cc5853e3-6e2d-4678-8312-74f8a7d76474).",
                    "id": "d85c6ec6-a693-43b3-904a-a997e1fc844d",
                    "createdAt": 1510811601.938
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 cba86182-52bf-42d7-9df8-b744699e6cfc) (task f4c1ad74-a5c6-4620-90cf-2aff118df5fc).",
                    "id": "095703e1-0ca3-4379-a7c8-c0f1b8b95ace",
                    "createdAt": 1510811364.691
                }
            ],
            "runningCount": 0,
            "placementStrategy": []
        }
    ],
    "failures": []
}
```

Endpoint de metadados de tarefas do Amazon ECS

O Amazon ECS fornece um método para recuperar diversos metadados de tarefas e estatísticas do Docker. Isso é chamado de endpoint de metadados de tarefas. O endpoint de metadados de tarefas está disponível para tarefas que usam o tipo de inicialização Fargate na versão de plataforma v1.1.0 ou posterior.

Todos os contêineres que pertencem a tarefas que são executadas no modo de rede `awsvpc` recebem um endereço IPv4 local dentro de um intervalo pré-definido de endereços locais de link. Quando um contêiner consulta o endpoint de metadados, o agente de contêiner do Amazon ECS pode determinar a qual contêiner a tarefa pertence com base em seu endereço IP exclusivo, e os metadados e as estatísticas dessa tarefa são retornados.

Para obter informações sobre um aplicativo Go de exemplo que consulta os endpoints da API de metadados e estatísticas, consulte https://github.com/aws/amazon-ecs-agent/blob/2bf4348a0ff89e23be4e82a6c5ff28edf777092c/misc/taskmetadata-validator/taskmetadata-validator.go.

Habilitação de metadados de tarefas

O recurso de endpoint de metadados de tarefas é habilitado por padrão para tarefas que usam o tipo de inicialização Fargate na versão de plataforma v1.1.0 ou posterior. Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).

Caminhos do endpoint de metadados de tarefas

Os seguintes endpoints de API estão disponíveis para os contêineres:

`169.254.170.2/v2/metadata`

Esse endpoint retorna o JSON de metadados para a tarefa, incluindo uma lista dos nomes e IDs de todos os contêineres associados à tarefa. Para obter mais informações sobre a resposta para esse endpoint, consulte Resposta do JSON de metadados de tarefas (p. 273).

`169.254.170.2/v2/metadata/<container-id>`

Esse endpoint retorna o JSON de metadados para o ID de contêiner do Docker especificado.

`169.254.170.2/v2/stats`

Esse endpoint retorna o JSON de estatísticas do Docker para todos os contêineres associados à tarefa. Para obter mais informações sobre cada uma das estatísticas retornadas, consulte ContainerStats na documentação da API do Docker.

`169.254.170.2/v2/stats/<container-id>`

Esse endpoint retorna o JSON de estatísticas do Docker para o ID de contêiner do Docker especificado. Para obter mais informações sobre cada uma das estatísticas retornadas, consulte ContainerStats na documentação da API do Docker.

Resposta do JSON de metadados de tarefas

As seguintes informações são retornadas da resposta em JSON (`169.254.170.2/v2/metadata`) do endpoint de metadados de tarefas.

`Cluster`

The Amazon ECS cluster to which the task belongs.

`TaskARN`

The full Amazon Resource Name (ARN) of the task to which the container belongs.

`Family`

The family of the Amazon ECS task definition for the task.

`Revision`

The revision of the Amazon ECS task definition for the task.

`DesiredStatus`

The desired status for the task from Amazon ECS.

`KnownStatus`

The known status for the task from Amazon ECS.

`Containers`

A list of container metadata for each container associated with the task.

`DockerId`

The Docker ID for the container.

`Name`

The name of the container as specified in the task definition.

`DockerName`

The name of the container supplied to Docker. The Amazon ECS container agent generates a unique name for the container to avoid name collisions when multiple copies of the same task definition are run on a single instance.

`Image`

The image for the container.

`ImageID`

The SHA-256 digest for the image.

`Ports`

Any ports exposed for the container. This parameter is omitted if there are no exposed ports.

`Labels`

Any labels applied to the container. This parameter is omitted if there are no labels applied.

`DesiredStatus`

The desired status for the container from Amazon ECS.

`KnownStatus`

The known status for the container from Amazon ECS.

`ExitCode`

The exit code for the container. This parameter is omitted if the container has not exited.

`Limits`

The resource limits specified at the container level (such as CPU and memory). This parameter is omitted if no resource limits are defined.

`CreatedAt`

The time stamp for when the container was created. This parameter is omitted if the container has not been created yet.

`StartedAt`

The time stamp for when the container started. This parameter is omitted if the container has not started yet.

`FinishedAt`

The time stamp for when the container stopped. This parameter is omitted if the container has not stopped yet.

`Type`

The type of the container. Containers that are specified in your task definition are of type `NORMAL`. You can ignore other container types, which are used for internal task resource provisioning by the Amazon ECS container agent.

`Networks`

The network information for the container, such as the network mode and IP address. This parameter is omitted if no network information is defined.

`Limits`

The resource limits specified at the task level (such as CPU and memory). This parameter is omitted if no resource limits are defined.

`PullStartedAt`

The time stamp for when the first container image pull began.

`PullStoppedAt`

The time stamp for when the last container image pull finished.

`ExecutionStoppedAt`

The time stamp for when the tasks `DesiredStatus` moved to `STOPPED`. This occurs when an essential container moves to `STOPPED`.

`AvailabilityZone`

The Availability Zone the task is in.

Exemplo de resposta de metadados de tarefas

A seguinte resposta em JSON apresenta uma tarefa de contêiner único.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-
f63cb662a5d3",
  "Family": "nginx",
```

```
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RESOURCES_PROVISIONED",
      "KnownStatus": "RESOURCES_PROVISIONED",
      "Limits": {
        "CPU": 0,
        "Memory": 0
      },
      "CreatedAt": "2018-02-01T20:55:08.366329616Z",
      "StartedAt": "2018-02-01T20:55:09.058354915Z",
      "Type": "CNI_PAUSE",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    },
    {
      "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
      "Name": "nginx-curl",
      "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
      "Image": "nrdlngr/nginx-curl",
      "ImageID": "sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "nginx-curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
        "CPU": 512,
        "Memory": 512
      },
      "CreatedAt": "2018-02-01T20:55:10.554941919Z",
      "StartedAt": "2018-02-01T20:55:11.064236631Z",
      "Type": "NORMAL",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
```

```
      ]
    }
  ],
  "PullStartedAt": "2018-02-01T20:55:09.372495529Z",
  "PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
  "AvailabilityZone": "us-east-2b"
}
```

Limites de serviço do Amazon ECS

A tabela a seguir fornece os limites de serviço padrão, também conhecidos como cotas, para o Amazon ECS para uma conta da AWS que pode ser alterada. Para obter mais informações sobre os limites de serviço para outros serviços da AWS que você pode usar com o Amazon ECS, como Elastic Load Balancing e Auto Scaling, consulte AWS Service Limits no Referência geral do Amazon Web Services.

Recurso	Limite padrão
Número de clusters por região, por conta	2000
Número de instâncias de contêiner por cluster	2000
Número de serviços por cluster	1000
Número de tarefas por serviço (a contagem desejada)	1000
Número de grupos de destino do Elastic Load Balancing por serviço	5
Número de tarefas que usam o tipo de inicialização Fargate, por região, por conta	50
Número de endereços IP públicos para tarefas usando o tipo de inicialização do Fargate, por região, por conta	50
Número de tarefas iniciadas usando o tipo de inicialização Fargate por segundo, por região, por conta Note Esse limite se aplica a tarefas autônomas e a tarefas iniciadas como parte de um serviço.	1 sustentada, com a possibilidade de intermitência até 10
Número de tarefas iniciadas usando o tipo de inicialização EC2 por segundo, por região, por conta Note Esse limite se aplica a tarefas autônomas e a tarefas iniciadas como parte de um serviço.	20 sustentadas, com a possibilidade de intermitência até 100

A tabela a seguir apresenta outras limitações para o Amazon ECS que não podem ser alteradas.

Recurso	Limit
Número de Classic Load Balancer por serviço	1
Número de tarefas ativadas (`count`) por run-task	10

Recurso	Limit
Número de instâncias de contêiner por start-task	10
Número de revisões por família de definição de tarefa Note Cancelar o registro de uma revisão de definição de tarefa não a isenta de ser incluída nesse limite.	1.000.000
Limite de tamanho de definição da tarefa	32 KiB
Máximo de contêineres de definição da tarefa	10
Número de sub-redes especificado em `awsvpcConfiguration`	16
Número de grupos de segurança especificado em `awsvpcConfiguration`	5
Tamanho máximo de um volume compartilhado usado por vários contêineres dentro de uma tarefa usando o tipo de inicialização Fargate	4 GB
Armazenamento de contêiner máximo para tarefas que usam o tipo de inicialização Fargate	10 GB
Número máximo de tags por recurso (tarefas, serviços, definições de tarefa, clusters e instâncias de contêiner)	50

Conceitos básicos do AWS App Mesh e do Amazon ECS

O AWS App Mesh é um serviço baseado no proxy Envoy que facilita o monitoramento e o controle de microsserviços. O App Mesh padroniza o modo como seus microsserviços se comunicam, oferecendo a você visibilidade completa e ajudando a garantir alta disponibilidade para seus aplicativos.

O App Mesh oferece controles do tráfego de rede e visibilidade consistentes para cada microsserviço em um aplicativo. Para obter mais informações, consulte o Guia do usuário do AWS App Mesh.

Este tópico ajuda você a usar o AWS App Mesh com um aplicativo de microsserviço existente em execução no Amazon ECS.

Pré-requisitos

O App Mesh oferece suporte a aplicativos de microsserviços que usam a nomeação de descoberta de serviço para seus componentes. Para usar este guia de conceitos básicos, você deve ter um aplicativo de microsserviço em execução no Amazon ECS que já tenha a descoberta de serviço configurada.

Para obter mais informações sobre a descoberta de serviço no Amazon ECS, consulte Descoberta de serviço (p. 137).

Etapa 1: criar sua malha de serviços

Uma malha de serviços é um limite lógico para o tráfego de rede entre os serviços que residem nela. Para obter mais informações, consulte Malhas de serviços no Guia do usuário do AWS App Mesh.

Depois de criar sua malha de serviços, você poderá criar serviços virtuais, nós virtuais, roteadores virtuais e rotas para distribuir o tráfego entre os aplicativos na sua malha.

Para criar uma nova malha de serviços com o Console de gerenciamento da AWS

1. Abra o console do App Mesh em https://console.aws.amazon.com/appmesh/.
2. Escolha Create mesh (Criar malha).
3. Em Mesh name (Nome da malha), especifique um nome para sua malha de serviços.
4. Escolha Create mesh (Criar malha) para concluir.

Etapa 2: criar seus nós virtuais

Um nó virtual funciona como um apontador lógico para um serviço do Amazon ECS. Para obter mais informações, consulte Nós virtuais no Guia do usuário do AWS App Mesh.

Ao criar um nó virtual, você deve especificar um método de descoberta de serviço para seu grupo de tarefas. Qualquer tráfego de entrada que seu nó virtual esperar, deverá ser especificado como um listener. Qualquer tráfego de saída que seu nó virtual espera alcançar deverá ser especificado como um back-end.

Você deve criar nós virtuais para cada microsserviço no seu aplicativo.

Para criar um nó virtual no Console de gerenciamento da AWS.

1. Escolha a malha que você criou nas etapas anteriores.
2. Escolha Virtual nodes (Nós virtuais) no painel de navegação à esquerda.
3. Selecione Create nó virtual (Criar nó virtual).
4. Em Virtual node name (Nome do nó virtual), escolha um nome para seu nó virtual.
5. Em Service discovery method (Método de descoberta de serviço), escolha uma das seguintes opções:
   - DNS – especifique o nome do host com registro DNS do serviço real que o nó virtual representa. Para obter mais informações sobre como usar o DNS como método de descoberta de serviço, consulte Nós virtuais.
   - AWS Cloud Map – especifique o nome do serviço e o namespace. Opcionalmente, você também pode especificar atributos para os quais o App Mesh pode consultar o AWS Cloud Map. Somente as instâncias que correspondam a todos os pares de chave/valor serão retornadas. Para usar o AWS Cloud Map, sua conta deve ter a `AWSServiceRoleForAppMesh` função vinculada ao serviço (p. 216).
6. Para especificar back-ends (para o tráfego de saída) para seu nó virtual, ou para configurar informações de registro em log de acesso de entrada e saída, escolha Additional configuration (Configuração adicional).
   a. Para especificar um back-end, selecione Add backend (Adicionar back-end) e insira um nome de serviço virtual ou um Amazon Resource Name (ARN) completo para o serviço virtual com o qual o seu nó virtual se comunica. Repita esta etapa até que todos os seus back-ends de nó virtual sejam considerados.
   b. Para configurar o registro em log, insira o caminho de logs de acesso HTTP que o Envoy deve usar. Recomendamos o caminho `/dev/stdout` para que você possa usar drivers de log do Docker para exportar seus logs do Envoy para um serviço, como o Amazon CloudWatch Logs.

      Note

      Os logs ainda devem ser ingeridos por um agente no seu aplicativo e enviados para um destino. Esse caminho de arquivo apenas informa ao Envoy para onde os logs devem ser enviados.
7. Especifique uma Port (Porta) e um Protocol (Protocolo) para o Listener.
8. Se desejar configurar verificações de integridade para seu listener, verifique se a opção Health check enabled (Verificação de integridade habilitada) está selecionada e conclua as subetapas a seguir. Se não desejar, desmarque essa caixa de seleção.
   a. Em Health check protocol (Protocolo da verificação de integridade), escolha usar uma verificação de integridade HTTP ou TCP.
   b. Em Health check port (Porta de verificação de integridade), especifique a porta em que a verificação de integridade deve ser executada.
   c. Em Healthy threshold (Limite de integridade), especifique o número de verificações de integridade consecutivas bem-sucedidas que devem ocorrer antes de o listener ser declarado íntegro.
   d. Em Health check interval (Intervalo de verificação de integridade), especifique o período em milissegundos entre cada execução de verificação de integridade.
   e. Em Path (Caminho), especifique o caminho de destino para a solicitação de verificação de integridade. Isso será necessário apenas se o protocolo especificado for HTTP. Se o protocolo for TCP, esse parâmetro será ignorado.

f. Em Timeout period (Período de tempo limite), especifique o tempo de espera ao receber uma resposta da verificação de integridade, em milissegundos.

g. Em Unhealthy threshold (Limite de não integridade), especifique o número de verificações de integridade consecutivas com falha que devem ocorrer antes de o listener ser declarado não íntegro.

9. Escolha Create nó virtual (Criar nó virtual) para concluir.
10. Repita este procedimento conforme necessário para criar nós virtuais para cada microsserviço restante no seu aplicativo.

Etapa 3: criar seus roteadores virtuais

Os roteadores virtuais cuidam do tráfego de um ou mais serviços virtuais dentro da malha. Depois de criar um roteador virtual, você poderá criar e associar rotas para seu roteador virtual que direcionem as solicitações de entrada para diferentes nós virtuais. Para obter mais informações, consulte Roteadores virtuais no Guia do usuário do AWS App Mesh.

Crie roteadores virtuais para cada microsserviço no seu aplicativo.

Criação de um roteador virtual no Console de gerenciamento da AWS.

1. Escolha Virtual routers (Roteadores virtuais) no painel de navegação à esquerda.
2. Escolha Create virtual router (Criar roteador virtual).
3. Em Virtual router name (Nome do roteador virtual), especifique um nome para seu roteador virtual. São permitidos até 255 letras, números, hifens e sublinhados.
4. Em Listener, especifique uma Port (Porta) e um Protocol (Protocolo) para seu roteador virtual.
5. Escolha Create virtual router (Criar roteador virtual) para concluir.
6. Repita este procedimento conforme necessário para criar roteadores virtuais para cada microsserviço restante no seu aplicativo.

Etapa 4: criar suas rotas

Uma rota é associada a um roteador virtual e é usada para fazer a correspondência de solicitações para a um roteador virtual e distribuir o tráfego para os nós virtuais associados da forma adequada. Para obter mais informações, consulte Rotas no Guia do usuário do AWS App Mesh.

Crie rotas para cada microsserviço no seu aplicativo.

Criação de uma rota no Console de gerenciamento da AWS.

1. Escolha Virtual routers (Roteadores virtuais) no painel de navegação à esquerda.
2. Escolha o roteador ao qual você deseja associar uma nova rota.
3. Na tabela Routes (Rotas), selecione Create route (Criar rota).
4. Em Route name (Nome da rota), especifique o nome a ser usado para a rota.
5. Em Route type (Tipo de rota), escolha o protocolo para a rota.
6. (Opcional) Em Route priority (Prioridade de rota), especifique a prioridade de 0 a 1000 para sua rota. A correspondência das rotas é feita com base em um valor especificado, e 0 é a prioridade mais alta.
7. Em Virtual node name (Nome do nó virtual), escolha o nó virtual para o qual essa rota enviará o tráfego.

8. Em Weight (Peso), escolha um peso relativo para a rota. Escolha Add target (Adicionar destino) para adicionar nós virtuais extras. O peso total de todos os destinos combinados deve ser menor ou igual a 100.
9. (Opcional) Para usar o caminho HTTP e o roteamento baseado no cabeçalho, escolha Additional configuration (Configurações adicionais).
10. (Opcional) Para usar um roteamento baseado em caminho HTTP, especifique o Prefix (Prefixo) que corresponde à rota. Para obter mais informações sobre o roteamento baseado em caminho, consulte Roteamento baseado em caminho.
11. (Opcional) Selecione um Method (Método) para usar o roteamento baseado no cabeçalho para sua rota. Para obter mais informações sobre o roteamento baseado em caminho, consulte Cabeçalhos HTTP.
12. (Opcional) Selecione um Scheme (Esquema) para usar o roteamento baseado no cabeçalho para sua rota.
13. (Opcional) Selecione Add header (Adicionar cabeçalho). Insira o Header name (Nome do cabeçalho) que você quer usar para o roteamento, selecione Match type (Tipo de correspondência) e insira um Match value (Valor de correspondência). Ao selecionar Invert (Inverter), a correspondência será oposta.
14. (Opcional) Selecione Add header (Adicionar cabeçalho) para adicionar até dez cabeçalhos.
15. Escolha Create route (Criar rota) para concluir.
16. Repita este procedimento conforme necessário para criar rotas para cada microsserviço restante no seu aplicativo.

Etapa 5: criar seus serviços virtuais

Um serviço virtual é uma abstração de um serviço real que é fornecido por um nó virtual direta ou indiretamente por meio de um roteador virtual. Os serviços dependentes chamam o serviço virtual pelo seu `virtualServiceName` e essas solicitações são roteadas para o nó virtual ou para o roteador virtual que é especificado como o provedor do serviço virtual. Para obter mais informações, consulte Serviços virtuais no Guia do usuário do AWS App Mesh.

Crie serviços virtuais para cada microsserviço no seu aplicativo.

Criação de um serviço virtual no Console de gerenciamento da AWS.

1. Selecione Virtual services (Serviços virtuais) no painel de navegação à esquerda.
2. Escolha Create virtual service (Criar serviço virtual).
3. Em Virtual service name (Nome do serviço virtual), escolha um nome para o serviço virtual. Recomendamos que você use o nome da descoberta de serviço do serviço real que você está visando (como `service-a.default.svc.cluster.local`). O nome especificado deve resultar em um endereço IP sem loopback.
4. Em Provider (Provedor), escolha o tipo de provedor para o serviço virtual:
   - Se desejar que o serviço virtual distribua o tráfego entre vários nós virtuais, selecione Virtual router (Roteador virtual) e escolha o roteador virtual a ser usado no menu suspenso.
   - Se desejar que o serviço virtual acesse um nó virtual diretamente, sem um roteador virtual, selecione Virtual node (Nó virtual) e escolha o nó virtual a ser usado no menu suspenso.
   - Se não desejar que o serviço virtual faça o roteamento do tráfego no momento (por exemplo, se os seus nós virtuais ou o roteador virtual ainda não existirem), escolha None (Nenhum). Você poderá atualizar o provedor desse serviço virtual mais tarde.
5. Escolha Create virtual service (Criar serviço virtual) para concluir.
6. Repita este procedimento conforme necessário para criar serviços virtuais para cada microsserviço restante no seu aplicativo.

Update Your Microservice Task Definitions

Proxy Configuration

To configure your Amazon ECS service to use App Mesh, your service's task definition must have the following proxy configuration section. Set the proxy configuration `type` to `APPMESH` and the `containerName` to `envoy`. Set the following property values accordingly.

`IgnoredUID`

Envoy doesn't proxy traffic from processes that use this user ID. You can choose any user ID that you want for this (our examples use `1337` for historical purposes), but this ID must be the same as the `user` ID for the Envoy container in your task definition. This matching allows Envoy to ignore its own traffic without using the proxy.

`ProxyIngressPort`

This is the ingress port for the Envoy proxy container. Set this value to `15000`.

`ProxyEgressPort`

This is the egress port for the Envoy proxy container. Set this value to `15001`.

`AppPorts`

Specify any ingress ports that your application containers listen on. In this example, the application container listens on port `9080`.

`EgressIgnoredIPs`

Envoy doesn't proxy traffic to these IP addresses. Set this value to `169.254.170.2,169.254.169.254`, which ignores the Amazon EC2 metadata server and the Amazon ECS task metadata endpoint (which provides IAM roles for tasks credentials).

```
"proxyConfiguration": {
  "type": "APPMESH",
  "containerName": "envoy",
  "properties": [
    {
      "name": "IgnoredUID",
      "value": "1337"
    },
    {
      "name": "ProxyIngressPort",
      "value": "15000"
    },
    {
      "name": "ProxyEgressPort",
      "value": "15001"
    },
    {
      "name": "AppPorts",
      "value": "9080"
    },
    {
      "name": "EgressIgnoredIPs",
      "value": "169.254.170.2,169.254.169.254"
    }
  ]
}
```

Application Container Envoy Dependency

The application containers in your task definitions must wait for the Envoy proxy to bootstrap and start before they can start. To ensure that this happens, you set a `dependsOn` section in each application container definition to wait for the Envoy container to report as `HEALTHY`. The following code block shows an application container definition example with this dependency.

```
    {
      "name": "app",
      "image": "application_image",
      "portMappings": [
        {
          "containerPort": 9080,
          "hostPort": 9080,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "dependsOn": [
        {
          "containerName": "envoy",
          "condition": "HEALTHY"
        }
      ]
    }
```

Envoy Container Definition

Your Amazon ECS services' task definitions must contain the App Mesh custom Envoy container image. You can replace the *Region* with any Region that App Mesh is supported in. For a list of supported regions, see AWS Service Endpoints.

```
840364872350.dkr.ecr.us-west-2.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-prod
```

The Envoy container definition must be marked as `essential`. The virtual node name for the Amazon ECS service should be set to the `APPMESH_VIRTUAL_NODE_NAME`, and the `user` ID value should match the `IgnoredUID` value from the task definition proxy configuration (in this example, we use `1337`). The health check shown here waits for the Envoy container to bootstrap properly before reporting to Amazon ECS that it is healthy and ready for the application containers to start.

The following code block shows an Envoy container definition example.

```
    {
      "name": "envoy",
      "image": "840364872350.dkr.ecr.us-west-2.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-
prod",
      "essential": true,
      "environment": [
        {
          "name": "APPMESH_VIRTUAL_NODE_NAME",
          "value": "mesh/meshName/virtualNode/virtualNodeName"
        }
      ],
      "healthCheck": {
        "command": [
          "CMD-SHELL",
          "curl -s http://localhost:9901/server_info | grep state | grep -q LIVE"
        ],
        "startPeriod": 10,
```

```
        "interval": 5,
        "timeout": 2,
        "retries": 3
      },
      "user": "1337"
    }
```

Credentials

The Envoy container requires AWS Identity and Access Management credentials for signing requests that are sent to the App Mesh service. Amazon ECS tasks deployed with the Fargate launch type do not have access to the Amazon EC2 metadata server that supplies instance IAM profile credentials. To supply the credentials, you must attach an IAM task role to any tasks deployed with the Fargate launch type. The role doesn't need to have a policy attached to it, but for a task to work properly with App Mesh, the role must be attached to each task deployed with the Fargate launch type.

Update an Existing Task Definition

The Amazon ECS console assists in the process of updating your existing task definitions to add App Mesh integration.

Update a task definition to add App Mesh integration

1. Open the Amazon ECS console at https://console.aws.amazon.com/ecs/.
2. From the navigation bar, choose the region that contains your task definition.
3. In the navigation pane, choose Task Definitions.
4. On the Task Definitions page, select the box to the left of the task definition to revise and choose Create new revision.
5. On the Create new revision of Task Definition page, make the following changes to enable App Mesh integration.

    a. For Service Integration, to configure the parameters for App Mesh integration choose Enable App Mesh integration and then do the following:

        i. For Application container name, choose the container name to use for the App Mesh application. This container must already be defined within the task definition.

        ii. For Envoy image, use the auto-populated Envoy container image which is 840364872350.dkr.ecr.*us-west-2*.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-prod.

        iii. For Mesh name, choose the App Mesh service mesh to use. This must already be created in order for it to show up. For more information, see Service Meshes in the AWS App Mesh User Guide.

        iv. For Virtual node name, choose the App Mesh virtual node to use. This must already be created in order for it to show up. For more information, see Virtual Nodes in the AWS App Mesh User Guide.

        v. For Virtual node port, this will be pre-populated with the listener port set on the virtual node.

        vi. Choose Apply, Confirm. This will create a new Envoy proxy container to the task definition, as well as the settings to support it. It will then pre-populate the App Mesh proxy configuration settings for the next step.

    b. For Proxy Configuration, verify all of the pre-populated values. For more information on these fields, see Proxy Configuration (p. 283).
6. Verify the information and choose Create.
7. If your task definition is used in a service, update your service with the updated task definition. For more information, see Atualizar um serviço (p. 160).

Example Task Definitions

The following example Amazon ECS task definitions show, in context, the snippets that you can merge with your existing task groups. Substitute your mesh name and virtual node name for the `APPMESH_VIRTUAL_NODE_NAME` value and a list of ports that your application listens on for the proxy configuration `AppPorts` value.

If you're running an Amazon ECS task as described in the section called “Credentials” (p. 285), you need an existing task IAM role (p. 242). You should also add this line of code to the example task definitions that follow: `"taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskRole"`

Example JSON for Amazon ECS task definition

```
{
  "family": "appmesh-gateway",
  "memory": "256",
  "proxyConfiguration": {
    "type": "APPMESH",
    "containerName": "envoy",
    "properties": [
      {
        "name": "IgnoredUID",
        "value": "1337"
      },
      {
        "name": "ProxyIngressPort",
        "value": "15000"
      },
      {
        "name": "ProxyEgressPort",
        "value": "15001"
      },
      {
        "name": "AppPorts",
        "value": "9080"
      },
      {
        "name": "EgressIgnoredIPs",
        "value": "169.254.170.2,169.254.169.254"
      }
    ]
  },
  "containerDefinitions": [
    {
      "name": "app",
      "image": "application_image",
      "portMappings": [
        {
          "containerPort": 9080,
          "hostPort": 9080,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "dependsOn": [
        {
          "containerName": "envoy",
          "condition": "HEALTHY"
        }
      ]
    },
    {
      "name": "envoy",
```

```
      "image": "840364872350.dkr.ecr.us-west-2.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-
prod",
      "essential": true,
      "environment": [
        {
          "name": "APPMESH_VIRTUAL_NODE_NAME",
          "value": "mesh/meshName/virtualNode/virtualNodeName"
        }
      ],
      "healthCheck": {
        "command": [
          "CMD-SHELL",
          "curl -s http://localhost:9901/server_info | grep state | grep -q LIVE"
        ],
        "startPeriod": 10,
        "interval": 5,
        "timeout": 2,
        "retries": 3
      },
      "user": "1337"
    }
  ],
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "networkMode": "awsvpc"
}
```

Example JSON for Amazon ECS task definition with AWS X-Ray

X-Ray allows you to collect data about requests that an application serves and provides tools that you can use to visualize traffic flow. Using the X-Ray driver for Envoy enables Envoy to report tracing information to X-Ray. You can enable X-Ray tracing using the Envoy configuration. Based on the configuration, Envoy sends tracing data to the X-Ray daemon running as a sidecar container and the daemon forwards the traces to the X-Ray service. Once the traces are published to X-Ray, you can use the X-Ray console to visualize the service call graph and request trace details. The following JSON represents a task definition to enable X-Ray integration.

```
{
  "family": "appmesh-gateway",
  "memory": "256",
  "proxyConfiguration": {
    "type": "APPMESH",
    "containerName": "envoy",
    "properties": [
      {
        "name": "IgnoredUID",
        "value": "1337"
      },
      {
        "name": "ProxyIngressPort",
        "value": "15000"
      },
      {
        "name": "ProxyEgressPort",
        "value": "15001"
      },
      {
        "name": "AppPorts",
        "value": "9080"
      },
      {
        "name": "EgressIgnoredIPs",
        "value": "169.254.170.2,169.254.169.254"
      }
    ]
```

```
  },
  "containerDefinitions": [
    {
      "name": "app",
      "image": "application_image",
      "portMappings": [
        {
          "containerPort": 9080,
          "hostPort": 9080,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "dependsOn": [
        {
          "containerName": "envoy",
          "condition": "HEALTHY"
        }
      ]
    },
    {
      "name": "envoy",
      "image": "840364872350.dkr.ecr.us-west-2.amazonaws.com/aws-appmesh-envoy:v1.11.2.0-
prod",
      "essential": true,
      "environment": [
        {
          "name": "APPMESH_VIRTUAL_NODE_NAME",
          "value": "mesh/meshName/virtualNode/virtualNodeName"
        },
        {
         "name": "ENABLE_ENVOY_XRAY_TRACING",
         "value": "1"
        }
      ],
      "healthCheck": {
        "command": [
          "CMD-SHELL",
          "curl -s http://localhost:9901/server_info | grep state | grep -q LIVE"
        ],
        "startPeriod": 10,
        "interval": 5,
        "timeout": 2,
        "retries": 3
      },
      "user": "1337"
    },
    {
      "name": "xray-daemon",
      "image": "amazon/aws-xray-daemon",
      "user": "1337",
      "essential": true,
      "cpu": 32,
      "memoryReservation": 256,
      "portMappings": [
        {
          "containerPort": 2000,
          "protocol": "udp"
        }
      ]
    }
  ],
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "networkMode": "awsvpc"
}
```

Tutoriais para Amazon ECS

Os seguintes tutoriais mostram como executar tarefas comuns ao usar o Amazon ECS.

Tópicos

- Tutorial: Creating a VPC with Public and Private Subnets for Your Clusters (p. 289)
- Tutorial: Especificação de dados confidenciais usando segredos do Secrets Manager (p. 291)
- Tutorial: como criar um serviço usando uma implantação azul/verde (p. 297)
- Tutorial: implantação contínua com o CodePipeline (p. 305)

Tutorial: Creating a VPC with Public and Private Subnets for Your Clusters

Container instances in your clusters need external network access to communicate with the Amazon ECS service endpoint. However, you might have tasks and services that you would like to run in private subnets. Creating a VPC with both public and private subnets provides you the flexibility to launch tasks and services in either a public or private subnet. Tasks and services in the private subnets can access the internet through a NAT gateway. Services in both the public and private subnets can be configured to use a load balancer so that they can still be reached from the public internet.

This tutorial guides you through creating a VPC with two public subnets and two private subnets, which are provided with internet access through a NAT gateway.

Step 1: Create an Elastic IP Address for Your NAT Gateway

A NAT gateway requires an Elastic IP address in your public subnet, but the VPC wizard does not create one for you. Create the Elastic IP address before running the VPC wizard.

To create an Elastic IP address

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the left navigation pane, choose Elastic IPs.
3. Choose Allocate new address, Allocate, Close.
4. Note the Allocation ID for your newly created Elastic IP address; you enter this later in the VPC wizard.

Step 2: Run the VPC Wizard

The VPC wizard automatically creates and configures most of your VPC resources for you.

To run the VPC wizard

1. In the left navigation pane, choose VPC Dashboard.

2. Choose Start VPC Wizard, VPC with Public and Private Subnets, Select.
3. For VPC name, give your VPC a unique name.
4. For Elastic IP Allocation ID, choose the ID of the Elastic IP address that you created earlier.
5. Choose Create VPC.
6. When the wizard is finished, choose OK. Note the Availability Zone in which your VPC subnets were created. Your additional subnets should be created in a different Availability Zone.

Step 3: Create Additional Subnets

The wizard creates a VPC with a single public and a single private subnet in a single Availability Zone. For greater availability, you should create at least one more of each subnet type in a different Availability Zone so that your VPC has both public and private subnets across two Availability Zones.

To create an additional private subnet

1. In the left navigation pane, choose Subnets.
2. Choose Create Subnet.
3. For Name tag, enter a name for your subnet, such as Private subnet.
4. For VPC, choose the VPC that you created earlier.
5. For Availability Zone, choose a different Availability Zone than your original subnets in the VPC.
6. For IPv4 CIDR block, enter a valid CIDR block. For example, the wizard creates CIDR blocks in 10.0.0.0/24 and 10.0.1.0/24 by default. You could use 10.0.3.0/24 for your second private subnet.
7. Choose Yes, Create.

To create an additional public subnet

1. In the left navigation pane, choose Subnets and then Create Subnet.
2. For Name tag, enter a name for your subnet, such as Public subnet.
3. For VPC, choose the VPC that you created earlier.
4. For Availability Zone, choose the same Availability Zone as the additional private subnet that you created in the previous procedure.
5. For IPv4 CIDR block, enter a valid CIDR block. For example, the wizard creates CIDR blocks in 10.0.0.0/24 and 10.0.1.0/24 by default. You could use 10.0.2.0/24 for your second public subnet.
6. Choose Yes, Create.
7. Select the public subnet that you just created and choose Route Table, Edit.
8. By default, the private route table is selected. Choose the other available route table so that the 0.0.0.0/0 destination is routed to the internet gateway (igw-***xxxxxxxx***) and choose Save.
9. With your second public subnet still selected, choose Subnet Actions, Modify auto-assign IP settings.
10. Select Enable auto-assign public IPv4 address and choose Save, Close.

Next Steps

After you have created your VPC, you should consider the following next steps:

- Create security groups for your public and private resources if they require inbound network access. For more information, see Working with Security Groups in the Amazon VPC User Guide.

- Create Amazon ECS clusters in your private or public subnets. For more information, see Criação de um cluster (p. 22). If you use the cluster creation wizard in the Amazon ECS console, you can specify the VPC that you just created and the public or private subnets in which to launch your instances, depending on your use case.
  - To make your containers directly accessible from the internet, launch instances into your public subnets. Be sure to configure your container instance security groups appropriately.
  - To avoid making containers directly accessible from the internet, launch instances into your private subnets.
- Create a load balancer in your public subnets that can route traffic to services in your public or private subnets. For more information, see Balanceamento de carga do serviço (p. 115).

Tutorial: Especificação de dados confidenciais usando segredos do Secrets Manager

O Amazon ECS permite que você injete dados confidenciais em seus contêineres armazenando os dados confidenciais nos segredos do AWS Secrets Manager e fazendo referência a eles na definição do contêiner. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).

O tutorial a seguir mostra como criar um segredo do Secrets Manager, fazer referência ao segredo em uma definição de tarefa do Amazon ECS e verificar se isso funcionou consultando a variável de ambiente dentro de um contêiner que mostra o conteúdo do segredo.

Pré-requisitos

Este tutorial pressupõe que os seguintes pré-requisitos foram concluídos:

- As etapas em Configuração com o Amazon ECS (p. 7) foram concluídas.
- O usuário da AWS tem as permissões do IAM necessárias para criar os recursos do Amazon ECS e do Secrets Manager descritos.

Etapa 1: criar um segredo do Secrets Manager

Você pode usar o console do Secrets Manager para criar um segredo para seus dados confidenciais. Neste tutorial, vamos criar um segredo básico para armazenar um nome de usuário e uma senha para consulta posterior em um contêiner. Para obter mais informações, consulte Criar um segredo básico no Guia do usuário do AWS Secrets Manager.

Para criar um segredo básico

Use o Secrets Manager para criar um segredo para seus dados confidenciais.

1. Abra o console do Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
2. Selecione Store a new secret (Armazenar um novo segredo).
3. Em Select secret type (Selecionar tipo de segredo), selecione Other type of secrets (Outro tipo de segredos).
4. Em Specify the key/value pairs to be stored in this secret (Especificar os pares de chave-valor a serem armazenados neste segredo), escolha a guia Plaintext (Texto simples) e substitua o texto existente pelo seguinte texto. O valor do texto especificado aqui será o valor da variável de ambiente em seu contêiner no final do tutorial.

```
password_value
```

5. Escolha Next (Próximo).
6. Em Secret name (Nome do segredo), digite `username_value` e escolha Next (Avançar). O valor do nome do segredo especificado aqui será o nome da variável de ambiente em seu contêiner no final do tutorial.
7. Em Configure automatic rotation (Configurar rotação automática), deixe Disable automatic rotation (Desabilitar rotação automática) selecionada e escolha Next (Avançar).
8. Reveja suas configurações e selecione Store (Armazenar) para salvar tudo o que inseriu como um novo segredo no Secrets Manager.
9. Selecione o segredo que você acabou de criar e salve o Secret ARN (ARN do segredo) para fazer referência na política do IAM de sua execução de tarefa e na definição de tarefa em etapas posteriores.

Etapa 2: atualizar a função do IAM de execução de tarefa

Para que o Amazon ECS recupere os dados confidenciais de seu segredo do Secrets Manager, você deve ter a função de execução da tarefa do Amazon ECS e fazer referência a ela em sua definição de tarefa. Isso permite que o agente de contêiner obtenha os recursos necessários do Secrets Manager. Se você ainda não tiver criado a função do IAM de execução de tarefa, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

As etapas a seguir pressupõem que a função do IAM de execução de tarefa já tenha sido criada e configurada corretamente.

Para atualizar a função do IAM de execução de tarefa

Use o console do IAM para atualizar a função de execução de tarefa com as permissões necessárias.

1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação, selecione Roles.
3. Procure por `ecsTaskExecutionRole` na lista de funções e selecione-a.
4. Em Permissions (Permissões), escolha Add inline policy (Adicionar política em linha).
5. Escolha a guia JSON e especifique o seguinte texto JSON, garantindo que você especifique o ARN completo do segredo do Secrets Manager que você criou na etapa 1.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:region:aws_account_id:secret:username_value-
u9bH6K"
            ]
        }
    ]
}
```

6. Escolha Review policy (Revisar política). Em Name (Nome) especifique `ECSSecretsTutorial` e escolha Create policy (Criar política).

Etapa 3: criar uma definição de tarefa do Amazon ECS

Você pode usar o console do Amazon ECS para criar uma definição de tarefa que faça referência a um segredo do Secrets Manager.

Para criar uma definição de tarefa que especifica um segredo

Use o console do IAM para atualizar a função de execução de tarefa com as permissões necessárias.

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, escolha Task Definitions (Definições de tarefa), Create new Task Definition (Criar nova definição de tarefa).
3. Na página Select launch type compatibility (Selecionar compatibilidade do tipo de inicialização), escolha EC2 e escolha Next step (Próxima etapa).
4. Escolha Configure via JSON (Configurar por meio de JSON) e insira o seguinte texto JSON de definição de tarefa, garantindo que você especifique o ARN completo do segredo do Secrets Manager que você criou na etapa 1 e a função do IAM de execução da tarefa que você atualizou na etapa 2. Escolha Salvar.

   Important

   O valor do nome do segredo na definição da tarefa deve corresponder ao nome especificado para o nome do segredo quando o segredo foi criado.

```
{
    "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole",
    "containerDefinitions": [
        {
            "entryPoint": [
                "sh",
                "-c"
            ],
            "portMappings": [
                {
                    "hostPort": 80,
                    "protocol": "tcp",
                    "containerPort": 80
                }
            ],
            "command": [
                "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
 ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html && httpd-
foreground\""
            ],
            "cpu": 10,
            "secrets": [
                {
                    "valueFrom":
 "arn:aws:secretsmanager:region:aws_account_id:secret:username_value-u9bH6K",
                    "name": "username_value"
                }
            ],
            "memory": 300,
            "image": "httpd:2.4",
```

```
            "essential": true,
            "name": "ecs-secrets-container"
        }
    ],
    "family": "ecs-secrets-tutorial"
}
```

5. Revise as configurações e escolha Create (Criar).

Etapa 4: criar um cluster do Amazon ECS

Você pode usar o console do Amazon ECS para criar um cluster que contém uma instância de contêiner na qual executar a tarefa. Se você tiver um cluster existente com pelo menos uma instância de contêiner registrada nele com os recursos disponíveis para executar uma instância da definição de tarefa criada para este tutorial, você poderá pular para a próxima etapa.

Para este tutorial, vamos criar um cluster com uma instância de contêiner `t2.micro` que usa a Amazon ECS-optimized Amazon Linux 2 AMI.

Para criar um cluster

Use o console do Amazon ECS para criar um cluster e registrar uma instância de contêiner nele.

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na barra de navegação, selecione a região que contém o segredo do Secrets Manager e a definição de tarefa do Amazon ECS que você criou.
3. No painel de navegação, escolha Clusters.
4. Na página Clusters, escolha Create Cluster (Criar cluster).
5. Em Select cluster compatibility (Selecionar a compatibilidade do cluster), escolha EC2 Linux + Networking (EC2 Linux + redes) e Next Step (Próxima etapa).
6. Na página Configure cluster (Configurar cluster), em Cluster name (Nome do cluster) insira `ecs-secrets-tutorial`.
7. Em EC2 instance type (Tipo de instância do EC2), escolha t2.micro.
8. Em Key pair (Par de chaves), escolha um par de chaves para adicionar à instância de contêiner.

   Important

   Um par de chaves é necessário para concluir o tutorial, portanto, se você ainda não tiver criado um par de chaves, siga o link para o console do EC2 para criar um.
9. Na seção Networking (Rede), configure a VPC para seu cluster. Selecione uma VPC existente ou escolha Create a new VPC (Criar uma nova VPC) a ser usada para o tutorial.
   a. (Opcional) Se você optar por criar uma nova VPC, em CIDR Block (Bloco CIDR), selecione um bloco CIDR para sua VPC. Para obter mais informações, consulte Sua VPC e suas sub-redes em Guia do usuário da Amazon VPC.
   b. Em Subnets, selecione as sub-redes a serem usadas para sua VPC. Você pode manter as configurações padrão ou modificá-las para atender às suas necessidades.
10. Em Container instance IAM role (Função do IAM de instância de contêiner), escolha a função do IAM de instância de contêiner existente ou selecione Create new role (Criar nova função) para ter uma criada para você.
11. Deixe todos os outros campos nos valores padrão e escolha Create (Criar).

Etapa 5: executar uma tarefa do Amazon ECS

Você pode usar o console do Amazon ECS para executar uma tarefa usando a definição de tarefa que você criou. Neste tutorial, executaremos uma tarefa usando o tipo de execução EC2 que usa o cluster que criamos na etapa anterior.

Para executar uma tarefa

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. No painel de navegação, selecione Task Definitions (Definições de tarefas) e selecione a definição de tarefa ecs-secrets-tutorial que criamos.
3. Selecione a última revisão da definição de tarefa e escolha Actions (Ações), Run Task (Executar tarefa).
4. Em Launch Type, selecione EC2.
5. Em Cluster, escolha o cluster ecs-secrets-tutorial que criamos na etapa anterior.
6. Em Task tagging configuration (Configuração de marcação de tarefa), desmarque Enable ECS managed tags (Habilitar tags gerenciadas pelo ECS). Elas são desnecessárias para os fins deste tutorial.
7. Revise as informações de sua tarefa e escolha Run Task.

   Note

   Se a sua tarefa mudar de `PENDING` para `STOPPED` ou se exibir o status `PENDING` e, em seguida, desaparecer das tarefas listadas, sua tarefa poderá ser interrompida devido a um erro. Para obter mais informações, consulte Como verificar se há erros em tarefas interrompidas (p. 310) na seção de solução de problemas.

Etapa 6: verificar

Você pode verificar se todas as etapas foram concluídas com êxito e a variável de ambiente foi criada corretamente em seu contêiner usando as etapas a seguir.

Para verificar se a variável de ambiente foi criada

1. Encontre o endereço IP público ou DNS para sua instância de contêiner.
   a. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
   b. Selecione o cluster `ecs-secrets-tutorial` que hospeda a instância de contêiner.
   c. Na página Cluster, escolha Instâncias do ECS.
   d. Na coluna Instância de contêiner, selecione a instância de contêiner à qual se conectar.
   e. Na página Instância de contêiner, registre o IP público ou DNS público para sua instância.
2. Se você estiver usando um computador Linux ou macOS, conecte-se à sua instância com o seguinte comando, substituindo o caminho para sua chave privada e o endereço público para sua instância:

```
$ ssh -i /path/to/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

   Para obter mais informações sobre como usar um computador Windows, consulte Conectar-se à instância Linux no Windows utilizando PuTTY no Guia do usuário do Amazon EC2 para instâncias do Linux.

Important

Para obter mais informações sobre problemas ao se conectar à sua instância, consulte Solucionar problemas para se conectar à instância no Guia do usuário do Amazon EC2 para instâncias do Linux.

3. Liste os contêineres em execução na instância. Anote o ID do contêiner `ecs-secrets-tutorial`.

```
docker ps
```

4. Conecte-se ao contêiner `ecs-secrets-tutorial` usando o ID do contêiner da saída da etapa anterior.

```
docker exec -it container_ID /bin/bash
```

5. Use o comando `echo` para imprimir o valor da variável de ambiente.

```
echo $username_value
```

Se o tutorial foi bem-sucedido, você deverá ver a saída a seguir:

```
password_value
```

Note

Como alternativa, você pode listar todas as variáveis de ambiente em seu contêiner usando o comando `env` (ou `printenv`).

Etapa 7: limpeza

Ao concluir este tutorial, você deve limpar os recursos associados para evitar cobranças por recursos não utilizados.

Para limpar os recursos

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Selecione o ecs-secrets-tutorial cluster (Cluster ecs-secrets-tutorial) que você criou.
3. Na página Cluster, escolha Delete Cluster (Excluir cluster).
4. Insira a frase de confirmação de exclusão do cluster e escolha Delete (Excluir). Isso levará alguns minutos, mas limpará todos os recursos do cluster do Amazon ECS.
5. Abra o console do IAM em https://console.aws.amazon.com/iam/.
6. No painel de navegação, selecione Roles.
7. Procure por `ecsTaskExecutionRole` na lista de funções e selecione-a.
8. Escolha Permissions (Permissões) e escolha o X ao lado de `ECSSecretsTutorial`. Escolha Remove (Remover) para confirmar a remoção da política em linha.
9. Abra o console do Secrets Manager em https://console.aws.amazon.com/secretsmanager/.
10. Selecione o segredo username_value que você criou e escolha Actions (Ações), Delete secret (Excluir segredo).

Tutorial: como criar um serviço usando uma implantação azul/verde

O Amazon ECS integrou as implantações azul/verde ao assistente Create Service (Criar serviço) no console do Amazon ECS. Para obter mais informações, consulte Criar um serviço (p. 148).

O tutorial a seguir mostra como criar um serviço do Amazon ECS contendo uma tarefa Fargate que usa o tipo de implantação azul/verde com a AWS CLI.

Pré-requisitos

Este tutorial pressupõe que você concluiu os seguintes pré-requisitos:

- A versão mais recente da AWS CLI está instalada e configurada. Para obter mais informações sobre como instalar ou fazer upgrade do AWS CLI, consulte Instalar o AWS Command Line Interface.
- As etapas em Configuração com o Amazon ECS (p. 7) foram concluídas.
- Seu usuário da AWS tem as permissões necessárias especificadas no exemplo de política Permissões do assistente de primeira execução do Amazon ECS (p. 202) do IAM.
- Você tem uma VPC e um grupo de segurança criados para uso. Para obter mais informações, consulte Tutorial: como criar uma VPC com sub-redes públicas e privadas para seus clusters.

Etapa 1: criar um Balanceador de carga de aplicações

Os serviços do Amazon ECS usando o tipo de implantação azul/verde exigem o uso de um Balanceador de carga de aplicações ou um Load balancer de rede. Este tutorial usa um Balanceador de carga de aplicações.

Para criar um Balanceador de carga de aplicações

1. Use o comando create-load-balancer para criar um Balanceador de carga de aplicações. Especifique duas sub-redes que não estejam na mesma Zona de disponibilidade, bem como um grupo de segurança.

```
aws elbv2 create-load-balancer --name bluegreen-alb \
--subnets subnet-abcd1234 subnet-abcd5678 --security-groups sg-abcd1234 --region us-east-1
```

   O resultado inclui o Nome de recurso da Amazon (ARN) do load balancer, com o seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/e5ba62739c16e642
```

2. Use o comando create-target-group para criar um grupo de destino. Esse grupo de destino roteará o tráfego para a tarefa original definida no seu serviço.

```
aws elbv2 create-target-group --name bluegreentarget1 --protocol HTTP --port 80 \
--target-type ip --vpc-id vpc-abcd1234 --region us-east-1
```

   A saída inclui o ARN do grupo de destino, com o seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/bluegreentarget1/209a844cd01825a4
```

3. Use o comando create-listener para criar um listener do load balancer com uma regra padrão que encaminha solicitações ao grupo de destino.

```
aws elbv2 create-listener --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 \
--protocol HTTP --port 80 \
--default-actions
 Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

A saída inclui o ARN do listener, com o seguinte formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/
e5ba62739c16e642/665750bec1b03bd4
```

Etapa 2: criar um cluster do Amazon ECS

Use o comando create-cluster para criar um cluster chamado `tutorial-bluegreen` a ser usado.

```
aws ecs create-cluster --cluster-name tutorial-bluegreen --region us-east-1
```

A saída inclui o ARN do cluster, com o seguinte formato:

```
arn:aws:ecs:region:aws_account_id:cluster/tutorial-bluegreen
```

Etapa 3: registrar uma definição de tarefa

Use o comando register-task-definition para registrar uma definição de tarefa compatível com o Fargate. Ele exige o uso do modo de rede `awsvpc`. Veja a seguir o exemplo de definição de tarefa usado para este tutorial.

Primeiro, crie um arquivo chamado `fargate-task.json` com os conteúdos a seguir. Certifique-se de usar o ARN para a sua função de execução da tarefa. Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

```
{
    "family": "tutorial-task-def",
        "networkMode": "awsvpc",
        "containerDefinitions": [
            {
                "name": "sample-app",
                "image": "httpd:2.4",
                "portMappings": [
                    {
                        "containerPort": 80,
                        "hostPort": 80,
                        "protocol": "tcp"
                    }
                ],
                "essential": true,
                "entryPoint": [
                    "sh",
                    "-c"
                ],
                "command": [
```

```
                    "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div
 style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!
</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></
html>' >  /usr/local/apache2/htdocs/index.html && httpd-foreground\""
                ]
            }
        ],
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "cpu": "256",
        "memory": "512",
        "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole"
}
```

Em seguida, registre a definição de tarefa usando o arquivo `fargate-task.json` criado.

```
aws ecs register-task-definition --cli-input-json file://fargate-task.json --region us-
east-1
```

Etapa 4: criar um serviço do Amazon ECS

Use o comando create-service para criar um serviço.

Primeiro, crie um arquivo chamado `service-bluegreen.json` com os conteúdos a seguir.

```
{
    "cluster": "tutorial-bluegreen",
    "serviceName": "service-bluegreen",
    "taskDefinition": "tutorial-task-def",
    "loadBalancers": [
        {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4",
            "containerName": "sample-app",
            "containerPort": 80
        }
    ],
    "launchType": "FARGATE",
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "CODE_DEPLOY"
    },
    "platformVersion": "LATEST",
    "networkConfiguration": {
       "awsvpcConfiguration": {
          "assignPublicIp": "ENABLED",
          "securityGroups": [ "sg-abcd1234" ],
          "subnets": [ "subnet-abcd1234", "subnet-abcd5678" ]
       }
    },
    "desiredCount": 1
}
```

Em seguida, crie o seu serviço usando o arquivo `service-bluegreen.json` que você criou.

```
aws ecs create-service --cli-input-json file://service-bluegreen.json --region us-east-1
```

A saída inclui o ARN do serviço, com o seguinte formato:

```
arn:aws:ecs:region:aws_account_id:service/service-bluegreen
```

Etapa 5: criar os recursos do AWS CodeDeploy

Use as seguintes etapas para criar seu aplicativo do CodeDeploy, o grupo de destino do Balanceador de carga de aplicações para o grupo de implantação do CodeDeploy e o grupo de implantação do CodeDeploy.

Para criar recursos do CodeDeploy

1. Use o comando create-application para criar um aplicativo do CodeDeploy. Especifique a plataforma de computação `ECS`.

   ```
   aws deploy create-application --application-name tutorial-bluegreen \
   --compute-platform ECS --region us-east-1
   ```

   A saída inclui o ID do aplicativo, com o seguinte formato:

   ```
   {
       "applicationId": "b8e9c1ef-3048-424e-9174-885d7dc9dc11"
   }
   ```

2. Use o comando create-target-group para criar um segundo grupo de destino do Balanceador de carga de aplicações, o qual será usado ao criar seu grupo de implantação do CodeDeploy.

   ```
   aws elbv2 create-target-group --name bluegreentarget2 --protocol HTTP --port 80 \
   --target-type ip --vpc-id "vpc-0b6dd82c67d8012a1" --region us-east-1
   ```

   A saída inclui o ARN para o grupo de destino, com o seguinte formato:

   ```
   arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
   bluegreentarget2/708d384187a3cfdc
   ```

3. Use o comando create-deployment-group para criar um grupo de implantação do CodeDeploy.

   Primeiro, crie um arquivo chamado `tutorial-deployment-group.json` com os conteúdos a seguir. Este exemplo usa o recurso que você criou.

   ```
   {
      "applicationName": "tutorial-bluegreen",
      "autoRollbackConfiguration": {
         "enabled": true,
         "events": [ "DEPLOYMENT_FAILURE" ]
      },
      "blueGreenDeploymentConfiguration": {
         "deploymentReadyOption": {
            "actionOnTimeout": "CONTINUE_DEPLOYMENT",
            "waitTimeInMinutes": 0
         },
         "terminateBlueInstancesOnDeploymentSuccess": {
            "action": "TERMINATE",
            "terminationWaitTimeInMinutes": 5
         }
      },
      "deploymentGroupName": "tutorial-bluegreen-dg",
      "deploymentStyle": {
         "deploymentOption": "WITH_TRAFFIC_CONTROL",
         "deploymentType": "BLUE_GREEN"
   ```

```
   },
   "loadBalancerInfo": {
      "targetGroupPairInfoList": [
        {
          "targetGroups": [
             {
                 "name": "bluegreentarget1"
             },
             {
                 "name": "bluegreentarget2"
             }
          ],
          "prodTrafficRoute": {
              "listenerArns": [
                  "arn:aws:elasticloadbalancing:region:aws_account_id:listener/
app/bluegreen-alb/e5ba62739c16e642/665750bec1b03bd4"
              ]
          }
        }
      ]
   },
   "serviceRoleArn": "arn:aws:iam::aws_account_id:role/ecsCodeDeployRole",
   "ecsServices": [
       {
           "serviceName": "service-bluegreen",
           "clusterName": "tutorial-bluegreen"
       }
   ]
}
```

Em seguida, cria um grupo de implantação do CodeDeploy.

```
aws deploy create-deployment-group --cli-input-json file://tutorial-deployment-
group.json --region us-east-1
```

A saída inclui o ID do grupo de implantação, com o seguinte formato:

```
{
    "deploymentGroupId": "6fd9bdc6-dc51-4af5-ba5a-0a4a72431c88"
}
```

Etapa 5: criar e monitorar uma implantação do CodeDeploy

Use as seguintes etapas para criar e fazer upload de um arquivo de especificação de aplicativo (arquivo AppSpec) e de uma implantação do CodeDeploy.

Como criar e monitorar uma implantação do CodeDeploy

1. Crie e faça upload de um arquivo AppSpec usando as etapas a seguir.

   a. Crie um arquivo denominado `appspec.yaml` com o conteúdo do grupo de implantação do CodeDeploy. Este exemplo usa os recursos criados por você anteriormente no tutorial.

      ```
      version: 0.0
      Resources:
        - TargetService:
            Type: AWS::ECS::Service
      ```

```
      Properties:
        TaskDefinition: "arn:aws:ecs:region:aws_account_id:task-definition/first-
run-task-definition:7"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
        PlatformVersion: "LATEST"
```

b. Use o comando s3 mb para criar um bucket do Amazon S3 para o arquivo AppSpec.

```
aws s3 mb s3://tutorial-bluegreen
```

c. Use o comando s3 cp para fazer upload do arquivo AppSpec para o bucket do Amazon S3.

```
aws s3 cp ./AppSpec.yaml s3://tutorial-bluegreen/appspec.yaml
```

2. Crie a implantação CodeDeploy usando as etapas a seguir.

a. Crie um arquivo denominado `create-deployment.json` com o conteúdo da implantação do CodeDeploy. Este exemplo usa os recursos criados por você anteriormente no tutorial.

```
{
    "applicationName": "tutorial-bluegreen",
    "deploymentGroupName": "tutorial-bluegreen-dg",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bucket": "tutorial-bluegreen",
            "key": "appspec.yaml",
            "bundleType": "YAML"
        }
    }
}
```

b. Use o comando create-deployment para criar uma implantação.

```
aws deploy create-deployment --cli-input-json file://create-deployment.json --
region us-east-1
```

A saída inclui o ID de implantação, com o seguinte formato:

```
{
    "deploymentId": "d-RPCR1U3TW"
}
```

c. Use o comando get-deployment-target para obter os detalhes da implantação, especificando o `deploymentId` da saída anterior.

```
aws deploy get-deployment-target --deployment-id "d-IMJU3A8TW" --target-
id tutorial-bluegreen:service-bluegreen --region us-east-1
```

Continue a recuperar os detalhes de implantação até que o status seja `Succeeded`, como mostra a saída a seguir.

```
{
    "deploymentTarget": {
        "deploymentTargetType": "ECSTarget",
        "ecsTarget": {
            "deploymentId": "d-RPCR1U3TW",
```

```
            "targetId": "tutorial-bluegreen:service-bluegreen",
            "targetArn": "arn:aws:ecs:region:aws_account_id:service/service-
bluegreen",
            "lastUpdatedAt": 1543431490.226,
            "lifecycleEvents": [
                {
                    "lifecycleEventName": "BeforeInstall",
                    "startTime": 1543431361.022,
                    "endTime": 1543431361.433,
                    "status": "Succeeded"
                },
                {
                    "lifecycleEventName": "Install",
                    "startTime": 1543431361.678,
                    "endTime": 1543431485.275,
                    "status": "Succeeded"
                },
                {
                    "lifecycleEventName": "AfterInstall",
                    "startTime": 1543431485.52,
                    "endTime": 1543431486.033,
                    "status": "Succeeded"
                },
                {
                    "lifecycleEventName": "BeforeAllowTraffic",
                    "startTime": 1543431486.838,
                    "endTime": 1543431487.483,
                    "status": "Succeeded"
                },
                {
                    "lifecycleEventName": "AllowTraffic",
                    "startTime": 1543431487.748,
                    "endTime": 1543431488.488,
                    "status": "Succeeded"
                },
                {
                    "lifecycleEventName": "AfterAllowTraffic",
                    "startTime": 1543431489.152,
                    "endTime": 1543431489.885,
                    "status": "Succeeded"
                }
            ],
            "status": "Succeeded",
            "taskSetsInfo": [
                {
                    "identifer": "ecs-svc/9223370493425779968",
                    "desiredCount": 1,
                    "pendingCount": 0,
                    "runningCount": 1,
                    "status": "ACTIVE",
                    "trafficWeight": 0.0,
                    "targetGroup": {
                        "name": "bluegreentarget1"
                    }
                },
                {
                    "identifer": "ecs-svc/9223370493423413672",
                    "desiredCount": 1,
                    "pendingCount": 0,
                    "runningCount": 1,
                    "status": "PRIMARY",
                    "trafficWeight": 100.0,
                    "targetGroup": {
                        "name": "bluegreentarget2"
                    }
                }
```

```
            ]
        }
    }
}
```

Etapa 6: limpeza

Ao concluir este tutorial, limpe os recursos associados a ele para evitar cobranças por recursos que você não está usando.

Limpando os recursos do tutorial

1. Use o comando delete-deployment-group para excluir o grupo de implantação do CodeDeploy.

```
aws deploy delete-deployment-group --application-name tutorial-bluegreen --deployment-group-name tutorial-bluegreen-dg --region us-east-1
```

2. Use o comando delete-application para excluir o aplicativo do CodeDeploy.

```
aws deploy delete-application --application-name tutorial-bluegreen --region us-east-1
```

3. Use o comando delete-service para excluir o serviço do Amazon ECS. Usar o sinalizador `--force` permite que você exclua um serviço, ainda que ele não tenha sido reduzido a zero tarefas.

```
aws ecs delete-service --service arn:aws:ecs:region:aws_account_id:service/service-bluegreen --force --region us-east-1
```

4. Use o comando delete-cluster para excluir o cluster do Amazon ECS.

```
aws ecs delete-cluster --cluster tutorial-bluegreen --region us-east-1
```

5. Use o comando s3 rm para excluir o arquivo AppSpec do bucket do Amazon S3.

```
aws s3 rm s3://tutorial-bluegreen/appspec.yaml
```

6. Use o comando s3 rb para excluir o bucket do Amazon S3.

```
aws s3 rb s3://tutorial-bluegreen
```

7. Use o comando delete-load-balancer para excluir o Balanceador de carga de aplicações.

```
aws elbv2 delete-load-balancer --load-balancer-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/
e5ba62739c16e642 --region us-east-1
```

8. Use o comando delete-target-group para excluir os dois grupos de destino do Balanceador de carga de aplicações.

```
aws elbv2 delete-target-group --target-group-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4 --region us-east-1
```

```
aws elbv2 delete-target-group --target-group-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/708d384187a3cfdc --region us-east-1
```

Tutorial: implantação contínua com o CodePipeline

Este tutorial ajuda você a criar um pipeline de implantação contínua (CD) completo e de ponta a ponta com o Amazon ECS, por meio do CodePipeline.

Pré-requisitos

Para você usar este tutorial para criar seu pipeline de CD, alguns recursos precisam estar em operação. Veja aqui estão os itens que você precisa para começar:

> Note
>
> Todos esses recursos devem ser criados na mesma região da AWS.

- Um repositório de controle de origem (este tutorial usa o CodeCommit) com seu Dockerfile e a origem do aplicativo. Para obter mais informações, consulte Criar um repositório do CodeCommit no Guia do usuário do AWS CodeCommit.
- Um repositório de imagens de docker (este tutorial usa o Amazon ECR) que contém uma imagem que você criou com o Dockerfile e a origem do aplicativo. Para obter mais informações, consulte Criar um repositório e Enviar uma imagem no Guia do usuário do Amazon Elastic Container Registry.
- Uma definição de tarefa do Amazon ECS que se refere à imagem de docker hospedada em seu repositório de imagens. Para obter mais informações, consulte Criar uma definição de tarefa no Amazon Elastic Container Service Developer Guide.
- Um cluster do Amazon ECS que executa um serviço que usa a definição de tarefa mencionada anteriormente. Para obter mais informações, consulte Criar um cluster e Criar um serviço no Amazon Elastic Container Service Developer Guide.

Assim que você atender a esses pré-requisitos, poderá continuar com o tutorial e criar seu pipeline de CD.

Etapa 1: Adicionar um arquivo de especificação de compilação ao repositório de origem

Este tutorial usa o CodeBuild para criar sua imagem do Docker e enviá-la para o Amazon ECR. Adicione um arquivo `buildspec.yml` ao repositório de código-fonte para informar ao CodeBuild como fazer isso. O exemplo de especificação de compilação abaixo faz o seguinte:

- Estágio pré-compilação:
  - Inicie sessão no Amazon ECR.
  - Defina a URI de repositório de sua imagem ECR e adicione uma tag de imagem com os primeiros sete caracteres do ID de confirmação do Git da origem.
- Estágio de compilação:
  - Crie a imagem de docker e marque-a como `latest` e com o ID de confirmação do Git.
- Estágio pós-compilação:
  - Envie por push a imagem para o repositório do ECR com ambas as tags.
  - Grave um arquivo denominado `imagedefinitions.json` na raiz da compilação que contém o nome do contêiner e a imagem e tag do serviço do Amazon ECS. O estágio de implantação do pipeline de CD usa essas informações para criar uma nova revisão da definição de tarefa do serviço e, em seguida, atualiza o serviço para usar a nova definição de tarefa. O arquivo `imagedefinitions.json` é necessário para o operador de trabalho ECS do CodeDeploy.

```
version: 0.2
```

```
phases:
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - aws --version
      - $(aws ecr get-login --region $AWS_DEFAULT_REGION --no-include-email)
      - REPOSITORY_URI=012345678910.dkr.ecr.us-west-2.amazonaws.com/hello-world
      - COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c 1-7)
      - IMAGE_TAG=${COMMIT_HASH:=latest}
  build:
    commands:
      - echo Build started on `date`
      - echo Building the Docker image...
      - docker build -t $REPOSITORY_URI:latest .
      - docker tag $REPOSITORY_URI:latest $REPOSITORY_URI:$IMAGE_TAG
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker images...
      - docker push $REPOSITORY_URI:latest
      - docker push $REPOSITORY_URI:$IMAGE_TAG
      - echo Writing image definitions file...
      - printf '[{"name":"hello-world","imageUri":"%s"}]' $REPOSITORY_URI:$IMAGE_TAG >
 imagedefinitions.json
artifacts:
    files: imagedefinitions.json
```

A especificação de compilação foi escrita para a definição de tarefa a seguir, usada pelo serviço do Amazon ECS para este tutorial. O valor `REPOSITORY_URI` corresponde ao repositório `image` (sem nenhuma tag de imagem) e o valor *`hello-world`* próximo do final do arquivo corresponde ao nome do contêiner na definição de tarefa do serviço.

```
{
  "taskDefinition": {
    "family": "hello-world",
    "containerDefinitions": [
      {
        "name": "hello-world",
        "image": "012345678910.dkr.ecr.us-west-2.amazonaws.com/hello-world:6a57b99",
        "cpu": 100,
        "portMappings": [
          {
            "protocol": "tcp",
            "containerPort": 80,
            "hostPort": 80
          }
        ],
        "memory": 128,
        "essential": true
      }
    ]
  }
}
```

Para adicionar um arquivo **`buildspec.yml`** ao repositório de origem

1. Abra um editor de texto e, em seguida, copie e cole a especificação de compilação acima em um novo arquivo.
2. Substitua o valor `REPOSITORY_URI` (*`012345678910.dkr.ecr.us-west-2.amazonaws.com/hello-world`*) pela URI do repositório do Amazon ECR (sem nenhuma tag de imagem) da imagem de docker. Substitua *`hello-world`* pelo nome do contêiner na definição de tarefa do serviço que se refere à imagem de docker.

3. Confirme e envie o arquivo `buildspec.yml` para o repositório de origem.

   a. Adicione o arquivo.

   ```
   git add .
   ```

   b. Confirme a alteração.

   ```
   git commit -m "Adding build specification."
   ```

   c. Envie a confirmação.

   ```
   git push
   ```

Etapa 2: Criar uma pipeline de implantação contínua

Use o assistente do CodePipeline para criar os estágios do pipeline e conectar o repositório de origem ao serviço do ECS.

Para criar o pipeline

1. Abra o console do CodePipeline em https://console.aws.amazon.com/codepipeline/.
2. Na página Bem-vindo, escolha Criar pipeline.

   Se esta for a primeira vez que você usa o CodePipeline, uma página de introdução será exibida em vez da página Welcome (Bem-vindo). Escolha Get Started Now.
3. Na página Step 1: Name, em Pipeline name, digite o nome do pipeline e escolha Next step. Para este tutorial, o nome do pipeline é hello-world.
4. Na página Step 2: Source (Etapa 2: origem), em Source provider (Fornecedor de origem), escolha CodeCommit.

   a. Em Repository name (Nome do repositório), escolha o nome do repositório do CodeCommit para usar como o local de origem do pipeline.

   b. Em Branch name, escolha a ramificação a ser usada e escolha Next step.
5. Na página Step 3: Build (Etapa 3: compilar), escolha CodeBuild e, em seguida, Create a new build project (Criar um novo projeto de compilação).

   a. Em Project name, escolha um nome exclusivo para seu projeto de compilação. Para este tutorial, o nome do projeto é hello-world.

   b. Para Operating system, selecione Ubuntu.

   c. Em Runtime (Tempo de execução), escolha Docker.

   d. Para Version, escolha aws/codebuild/docker:17.09.0 .

   e. Selecione Save build project.

   f. Escolha Próxima etapa.

      Note

      O assistente cria uma função de serviço do CodeBuild para o projeto de compilação, chamada code-build-***build-project-name***-service-role. Observe o nome da função ao adicionar permissões do Amazon ECR posteriormente.
6. Na página Step 4: Deploy (Etapa 4: implantar), em Deployment provider (Fornecedor de implantação), escolha Amazon ECS.

a. Em Cluster name (Nome do cluster), escolha o cluster do Amazon ECS no qual o serviço está em execução. Para este tutorial, o cluster é default.

b. Em Service name, escolha o serviço a ser atualizado e escolha Next step. Para este tutorial, o nome do serviço é hello-world.

7. Na página Step 5: Service Role, escolha Create role. Na página do console do IAM que descreve a função que será criada para você, escolha Allow (Permitir).
8. Escolha Próxima etapa.
9. Na página Step 6: Review, revise a configuração do pipeline e escolha Create pipeline para criá-lo.

   Note

   Agora que o pipeline foi criado, ele tentará passar pelos diferentes estágios de pipeline. No entanto, a função do CodeBuild criada pelo assistente não tem permissões para executar todos os comandos contidos no arquivo `buildspec.yml`. Por esse motivo, o estágio de compilação falha. A próxima seção adiciona as permissões para o estágio de compilação.

Etapa 3: Adicionar permissões do Amazon ECR para a função do CodeBuild

O assistente do CodePipeline criou uma função do IAM para o projeto de compilação do CodeBuild, chamada code-build-***build-project-name***-service-role. Para este tutorial, o nome é code-build-hello-world-service-role. Como o arquivo `buildspec.yml` faz chamadas para operações de API do Amazon ECR, a função deve ter uma política que conceda permissões para a realização dessas chamadas ao Amazon ECR. O procedimento a seguir ajuda você a anexar as permissões adequadas à função.

Para adicionar permissões do Amazon ECR à função do CodeBuild

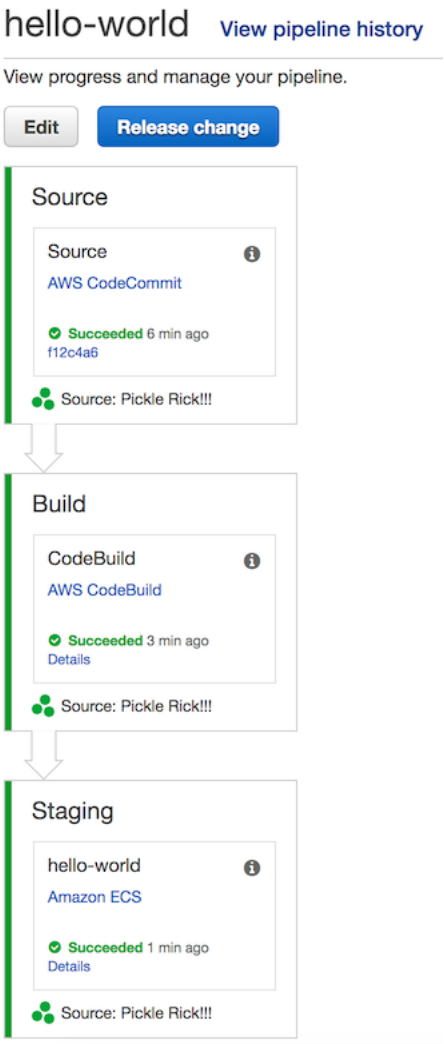
1. Abra o console do IAM em https://console.aws.amazon.com/iam/.
2. No painel de navegação à esquerda, escolha Roles.
3. Na caixa de pesquisa, digite code-build- e escolha a função que foi criada pelo assistente do CodePipeline. Para este tutorial, o nome da função é code-build-hello-world-service-role.
4. Na página Summary,. escolha Attach policy.
5. Selecione a caixa à esquerda da política AmazonEC2ContainerRegistryPowerUser e escolha Attach policy.

Etapa 4: Testar o pipeline

O pipeline deve ter tudo para realizar uma implantação contínua de ponta a ponta nativa da AWS. Agora, teste a funcionalidade enviando uma alteração de código ao repositório de origem.

Para testar o pipeline

1. Faça uma alteração no código no repositório de origem configurado, confirme e envie a alteração.
2. Abra o console do CodePipeline em https://console.aws.amazon.com/codepipeline/.
3. Escolha o pipeline na lista.
4. Observe a evolução do pipeline pelos respectivos estágios. O pipeline deve chegar ao fim para o serviço do Amazon ECS executar a imagem de docker que foi criada na alteração de código.


hello-world
View pipeline history
View progress and manage your pipeline.
Edit
Release change
Source
Source
AWS CodeCommit
Succeeded 6 min ago
f12c4a6
Source: Pickle Rick!!!
Build
CodeBuild
AWS CodeBuild
Succeeded 3 min ago
Details
Source: Pickle Rick!!!
Staging
hello-world
Amazon ECS
Succeeded 1 min ago
Details
Source: Pickle Rick!!!

Solução de problemas do Amazon ECS

Talvez você precise solucionar problemas com os load balancers, as tarefas, os serviços ou as instâncias de contêiner. Este capítulo ajuda a encontrar informações de diagnóstico do agente de contêiner do Amazon ECS, o daemon do Docker na instância de contêiner e o log de eventos de serviço no console do Amazon ECS.

Tópicos

- Solução de problemas de inicialização do assistente da primeira execução (p. 310)
- Como verificar se há erros em tarefas interrompidas (p. 310)
- Mensagens de evento de serviço (p. 312)
- Valor de memória ou CPU inválido especificado (p. 314)
- Erro "Não foi possível obter a imagem do contêiner" (p. 314)
- Como solucionar problemas de load balancers de serviço (p. 316)

Solução de problemas de inicialização do assistente da primeira execução

O seguinte erro pode impedir que o assistente da primeira execução do Amazon ECS crie seu cluster.

VpcLimitExceeded

Você poderá receber um erro `VpcLimitExceeded` ao tentar concluir o assistente da primeira execução do Amazon ECS. Nesse caso, você atingiu o limite do número de VPCs que podem ser criadas em uma região. Ao criar sua conta da AWS, há limites padrão do número de VPCs que podem ser executadas em cada Região. Para mais informações, consulte Limites da Amazon VPC.

Para resolver esse problema, você tem as seguintes opções:

- Solicite um aumento do limite de serviço da VPC por região. Para mais informações, consulte Limites da Amazon VPC.
- Exclua qualquer VPC não utilizada em sua conta. Para obter mais informações, consulte Trabalhar com VPCs e sub-redes.

Important

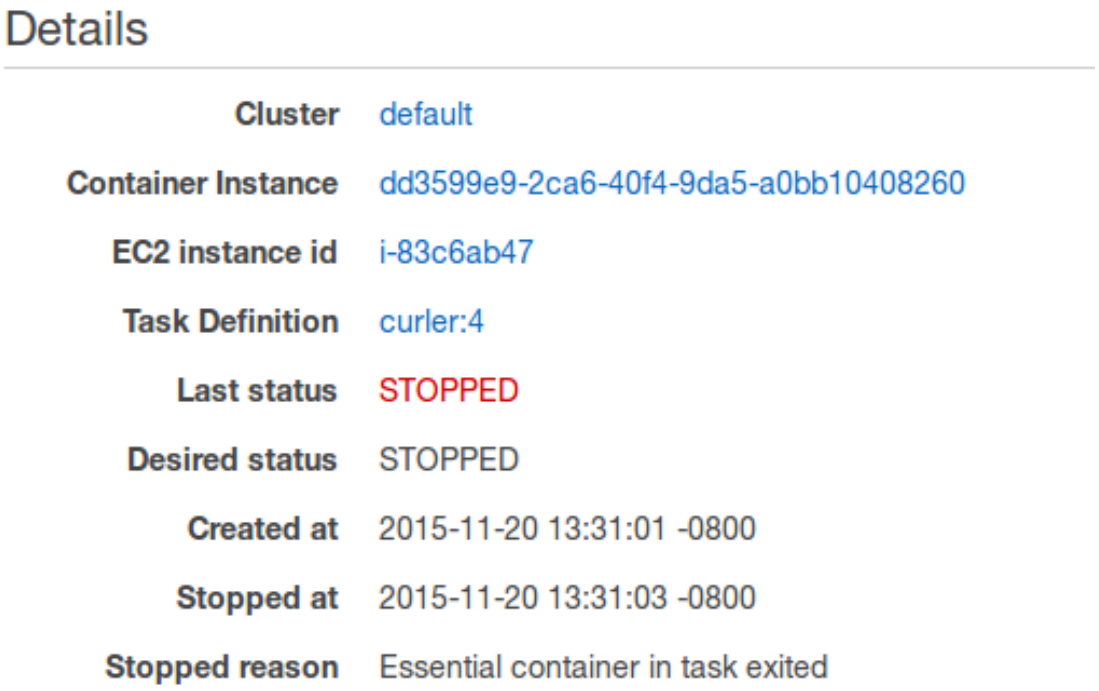
Qualquer recurso do Amazon ECS criado com êxito durante o assistente da primeira execução antes de aparecer esse erro pode ser excluído antes de executar o assistente novamente.

Como verificar se há erros em tarefas interrompidas

Se você tiver problemas ao iniciar uma tarefa, sua tarefa poderá ser interrompida devido a um erro. Por exemplo, você executa a tarefa, ela exibe um status `PENDING` e, em seguida, desaparece. Você pode visualizar erros como esse no console do Amazon ECS exibindo a tarefa interrompida e a inspecionando em busca de mensagens de erro.

Para verificar se há erros em tarefas interrompidas

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na página Clusters, selecione o cluster no qual a tarefa interrompida reside.
3. Na página Cluster: ***`clustername`***, escolha Tasks (Tarefas).
4. No cabeçalho da tabela Desired task status (Status da tarefa desejada), escolha Stopped (Interrompida) e, em seguida, selecione a tarefa interrompida a ser inspecionada. As tarefas interrompidas mais recentes são listadas primeiro.
5. Na seção Details (Detalhes), inspecione o campo Stopped reason (Motivo da interrupção) para ver o motivo pelo qual a tarefa foi interrompida.



Alguns motivos possíveis e suas explicações são listados abaixo:

Falha de tarefas nas verificações de integridade no (elb elb-name)

> A tarefa atual falhou na verificação de integridade do Elastic Load Balancing para o load balancer associado ao serviço da tarefa. Para obter mais informações, consulte Como solucionar problemas de load balancers de serviço (p. 316).

Ação de escalabilidade iniciada por (ID de implantação)

> Quando você reduzir a contagem desejada de um serviço estável, algumas tarefas devem ser interrompidas para alcançar o número desejado. Tarefas que são interrompidas por serviços de redução têm esse motivo de interrupção.

EC2 de host (`id` da instância) interrompido/encerrado

> Se você interromper ou encerrar uma instância de contêiner com tarefas em execução, as tarefas receberão esse motivo de interrupção.

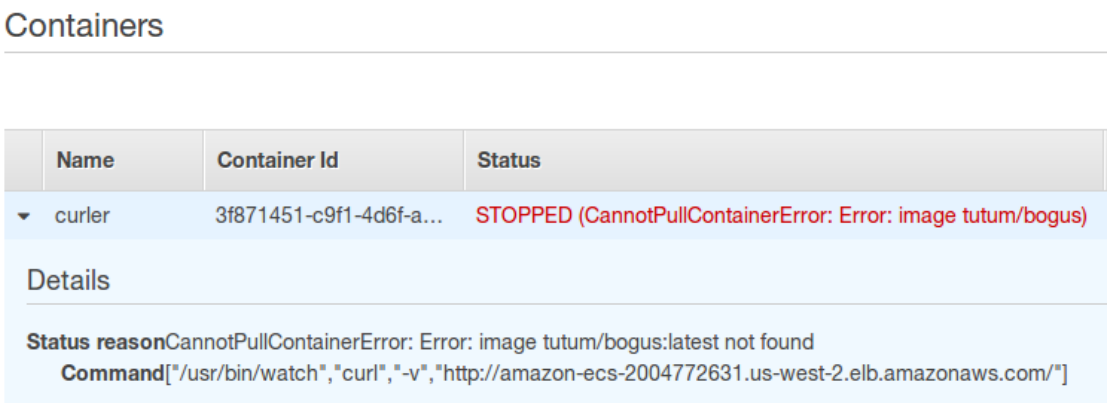
O cancelamento do registro de instâncias de contêiner forçado pelo usuário

> Se você forçar o cancelamento do registro de uma instância de contêiner com tarefas em execução, as tarefas receberão esse motivo de interrupção.

Contêiner essencial na tarefa encerrado

Se um contêiner marcado como `essential` nas definições de tarefa for fechado ou desativado, isso pode fazer com que uma tarefa pare. Quando a saída de um contêiner essencial é a causa de uma tarefa interrompida, o Step 6 (p. 312) pode fornecer mais informações de diagnóstico sobre a causa do contêiner interrompido.

6. Se você tiver um contêiner que foi interrompido, expanda-o e inspecione a linha Status reason (Motivo do status) para ver o que causou a mudança no estado da tarefa.



No exemplo anterior, o nome da imagem de contêiner não pode ser encontrada. Isso pode acontecer se você digitar incorretamente o nome da imagem.

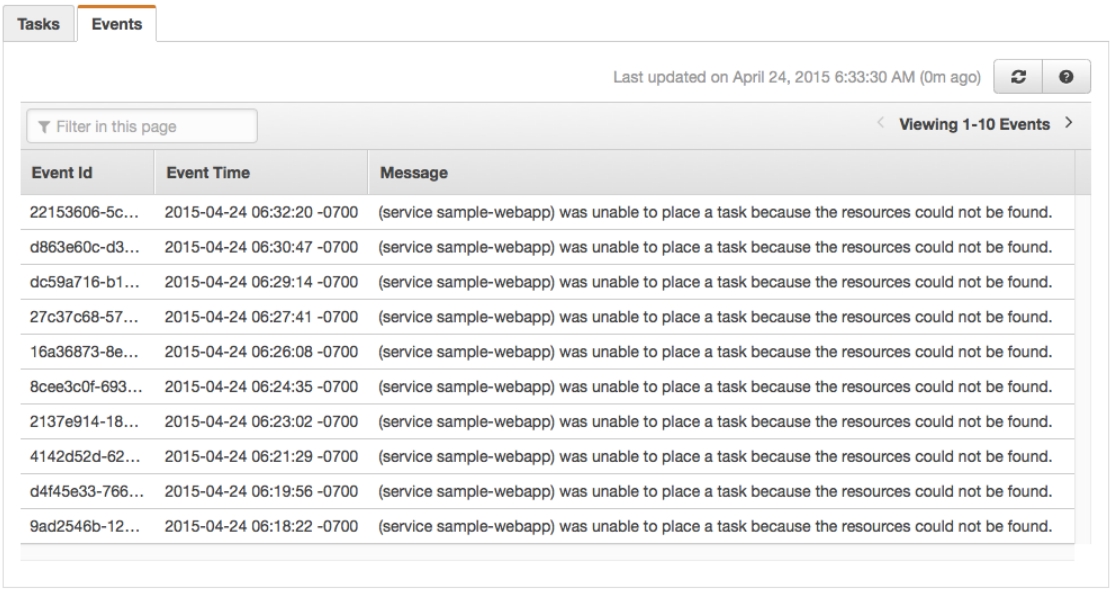
Mensagens de evento de serviço

Caso esteja solucionando um problemas com um serviço, o primeiro lugar em que você deve procurar informações de diagnóstico é o log de eventos do serviço.

Ao visualizar mensagens de evento do serviço no console do Amazon ECS, as mensagens de evento do serviço duplicadas são omitidas até que a causa seja resolvida ou se passem seis horas. Se a causa não for resolvida, você receberá outra mensagem de evento de serviço depois que se passarem seis horas.

Para verificar o log de eventos do serviço no console do Amazon ECS

1. Abra o console do Amazon ECS em https://console.aws.amazon.com/ecs/.
2. Na página Clusters, selecione o cluster no qual o serviço reside.
3. Na página Cluster: ***`clustername`***, selecione o serviço a ser inspecionado.
4. Na página Service: ***`servicename`***, selecione Events (Eventos).



5. Examine na coluna Message (Mensagem) erros ou outras informações úteis.

Mensagens de evento de serviço

Veja a seguir exemplos de mensagens de eventos de serviço que você pode ver no console:

- O serviço (*service-name*) (instância *instance-id*) sem integridade em (elb *elb-name*) por causa de (motivo de falha na instância em, pelo menos, o número de verificações de integridade UnhealthyThreshold consecutivas.) (p. 313)
- O serviço (*service-name*) não consegue iniciar tarefas com êxito de maneira consistente. (p. 313)

O serviço (*service-name*) (instância *instance-id*) sem integridade em (elb *elb-name*) por causa de (motivo de falha na instância em, pelo menos, o número de verificações de integridade UnhealthyThreshold consecutivas.)

O serviço é registrado com um load balancer, e as verificações de integridade do load balancer apresentam falhas. Para obter mais informações, consulte Como solucionar problemas de load balancers de serviço (p. 316).

O serviço (*service-name*) não consegue iniciar tarefas com êxito de maneira consistente.

Esse serviço contém tarefas que deixaram de ser iniciadas após tentativas consecutivas. Nesse ponto, o programador de serviço começa a aumentar incrementalmente o tempo entre as novas tentativas. Você deve solucionar o motivo pelo qual suas tarefas falham ao iniciar. Para obter mais informações, consulte Lógica de controle de serviço (p. 164).

Depois que o serviço estiver atualizado, por exemplo, com uma definição de tarefa atualizada, o programador de serviços retomará o comportamento normal.

Valor de memória ou CPU inválido especificado

Ao registrar uma tarefa, se você especificar um valor `cpu` ou `memory` inválido, receberá o seguinte erro:

```
An error occurred (ClientException) when calling the RegisterTaskDefinition operation:
 Invalid 'cpu' setting for task. For more information, see the Troubleshooting section of
 the Amazon ECS Developer Guide.
```

Para resolver esse problema, você deve especificar um valor compatível para a CPU e a memória em sua definição da tarefa.

O valor `cpu` pode ser expressado em unidades de CPU ou vCPUs em uma definição de tarefa, mas é convertido em um inteiro que indica as unidades de CPU quando a definição da tarefa for registrada. Se você estiver usando o tipo de inicialização EC2, os valores compatíveis estarão entre unidades de CPU `128` (vCPUs `0.125`) e unidades de CPU `10240` (vCPUs `10`). Se estiver usando o tipo de inicialização Fargate, você deverá usar um dos valores na tabela a seguir, que determina o intervalo de valores compatíveis para o parâmetro `memory`.

O valor `memory` pode ser expressado em MiB ou GB em uma definição de tarefa, mas é convertido em um inteiro que indica o MiB quando a definição da tarefa for registrada. Se você estiver usando o tipo de inicialização EC2, você deverá especificar um inteiro. Se estiver usando o tipo de inicialização Fargate, você deverá usar um dos valores na tabela a seguir, que determina o intervalo de valores compatíveis para o parâmetro `cpu`.

Os valores de CPU e memória da tarefa para tarefas Fargate são os indicados a seguir.

CPU value	Memory value (MiB)
256 (.25 vCPU)	512 (0.5GB), 1024 (1GB), 2048 (2GB)
512 (.5 vCPU)	1024 (1GB), 2048 (2GB), 3072 (3GB), 4096 (4GB)
1024 (1 vCPU)	2048 (2GB), 3072 (3GB), 4096 (4GB), 5120 (5GB), 6144 (6GB), 7168 (7GB), 8192 (8GB)
2048 (2 vCPU)	Between 4096 (4GB) and 16384 (16GB) in increments of 1024 (1GB)
4096 (4 vCPU)	Between 8192 (8GB) and 30720 (30GB) in increments of 1024 (1GB)

Erro "Não foi possível obter a imagem do contêiner"

Os seguintes erros do Docker indicam que, ao criar uma tarefa, a imagem do contêiner especificada não pôde ser recuperada.

A conexão atingiu o tempo limite

Quando uma tarefa Fargate é iniciada, sua interface de rede elástica requer uma rota para a internet a fim de obter imagens do contêiner. Se você receber um erro semelhante ao indicado a seguir ao iniciar uma tarefa, será porque uma rota para a internet não existe:

```
CannotPullContainerError: API error (500): Get https://111122223333.dkr.ecr.us-
east-1.amazonaws.com/v2/: net/http: request canceled while waiting for connection"
```

Para resolver esse problema, você pode:

Imagem não encontrada

Ao especificar uma imagem do Amazon ECR na definição do contêiner, você deve usar o ARN completo ou o URI do repositório ECR junto com o nome da imagem no repositório. Se o repositório ou a imagem não for encontrada, você receberá o seguinte erro:

```
CannotPullContainerError: API error (404): repository 111122223333.dkr.ecr.us-
east-1.amazonaws.com/<repo>/<image> not found
```

Para resolver esse problema, verifique o ARN ou o URI do repositório e o nome da imagem. Além disso, certifique-se de que o acesso apropriado foi configurado usando a função do IAM para execução de tarefas. Para obter mais informações sobre a função de execução de tarefas, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).

Espaço em disco insuficiente

Se o volume raiz da instância de contêiner tiver espaço em disco insuficiente durante a obtenção da imagem do contêiner, você verá uma mensagem de erro semelhante à seguinte:

```
CannotPullContainerError: write /var/lib/docker/tmp/GetImageBlob111111111: no space
 left on device
```

Para resolver esse problema, libere espaço em disco.

Se estiver usando a Amazon ECS-optimized AMI, você poderá usar o seguinte comando para recuperar os 20 maiores arquivos no sistema de arquivos:

```
du -Sh / | sort -rh | head -20
```

Exemplos de resultado:

```
5.7G    /var/lib/docker/
containers/50501b5f4cbf90b406e0ca60bf4e6d4ec8f773a6c1d2b451ed8e0195418ad0d2
1.2G    /var/log/ecs
594M    /var/lib/docker/devicemapper/mnt/
c8e3010e36ce4c089bf286a623699f5233097ca126ebd5a700af023a5127633d/rootfs/data/logs
...
```

Em alguns casos, como neste exemplo acima, o volume raiz pode ser preenchido por um contêiner em execução. Se o contêiner estiver usando o driver de log `json-file` padrão sem um limite `max-size`, o arquivo de log poderá ser responsável pela maioria do espaço usado. Você pode usar o comando `docker ps` para verificar qual contêiner está usando o espaço mapeando o nome do diretório da saída acima para o ID do contêiner. Por exemplo:

```
CONTAINER ID   IMAGE                              COMMAND             CREATED
 STATUS              PORTS                              NAMES
50501b5f4cbf   amazon/amazon-ecs-agent:latest   "/agent"            4 days ago
 Up 4 days                                             ecs-agent
```

Por padrão, ao usar o driver de log `json-file`, o Docker registra a saída padrão (e o erro padrão) de todos os contêineres e a grava em arquivos usando o formato JSON. Você pode definir o `max-size` como um driver de log, que impede que o arquivo de log ocupe muito espaço. Para obter mais informações, consulte Configurar drivers de registro em log na documentação do Docker.

Este é um trecho de definição de contêiner que mostra como usar essa opção:

```
{
```

```
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "256m"
    }
}
```

Uma alternativa caso os logs de contêiner estejam ocupando muito espaço em disco é usar o driver de log `awslogs`. O driver de log `awslogs` envia os logs para CloudWatch, que libera o espaço em disco que seria usado nos logs de contêiner na instância de contêiner. Para obter mais informações, consulte Como usar o driver de log awslogs (p. 64).

Como solucionar problemas de load balancers de serviço

Os serviços do Amazon ECS podem registrar tarefas com um load balancer do Elastic Load Balancing. Erros de configuração do load balancer são causas comuns de tarefas interrompidas. Caso as tarefas interrompidas tenham sido iniciadas por serviços que usam um load balancer, leve em consideração as possíveis causas a seguir.

> **Important**
>
> As verificações de integridade do contêiner não são compatíveis com tarefas que fazem parte de um serviço configurado para usar um Classic Load Balancer. O programador de serviços do Amazon ECS ignora tarefas em um estado `UNHEALTHY` que estiverem atrás de um Classic Load Balancer.

Grupo de segurança da instância de contêiner

Caso o contêiner seja mapeado para a porta 80 na instância de contêiner, o grupo de segurança da instância de contêiner deve permitir o tráfego de entrada na porta 80 para que as verificações de integridade do load balancer sejam aprovadas.

Load balancer do Elastic Load Balancing não configurado para todas as zonas de disponibilidade

O load balancer deve ser configurado para usar todas as zonas de disponibilidade em uma região, ou pelo menos todas as zonas de disponibilidade nas quais as instâncias de contêiner residem. Caso um serviço use um load balancer e inicie uma tarefa em uma instância de contêiner que resida em uma zona de disponibilidade para a qual o load balancer não está configurado para usar, a tarefa jamais será aprovada na verificação de integridade e será encerrada.

Verificação de integridade do load balancer do Elastic Load Balancing mal configurada

Os parâmetros de verificação de integridade do load balancer podem ser excessivamente restritivos ou apontar para recursos que não existem. Caso não seja considerada íntegra, a instância de contêiner é removida do load balancer. Não se esqueça de verificar se os parâmetros a seguir estão configurados corretamente para o load balancer de serviço.

Ping Port

O valor Ping Port de uma verificação de integridade do load balancer é a porta nas instâncias de contêiner que o load balancer verifica para determinar se ela é íntegra. Se essa porta estiver mal configurada, o balanceador de carga provavelmente cancelará o registro da instância de contêiner. Essa porta deve ser configurada a fim de usar o valor `hostPort` para o contêiner na definição de tarefa do serviço que você está usando com a verificação de integridade.

Ping Path

Este valor costuma ser definido como `index.html`, mas caso o serviço não responda a essa solicitação, a verificação de integridade falhará. Caso o contêiner não tenha um arquivo `index.html`, você pode defini-lo como `/` para segmentar o URL base da instância de contêiner.

Tempo limite de resposta

Este é o tempo que o contêiner tem para retornar uma resposta para o ping de verificação de integridade. Caso o valor seja menor que o tempo necessário a uma resposta, a verificação de integridade falhará.

Intervalo de verificação de integridade

Este é o tempo entre os pings de verificação de integridade. Quanto menor for o intervalo de verificação de integridade, mais rapidamente a instância de contêiner poderá atingir Unhealthy Threshold (Limite não íntegro).

Limite não íntegro

Este é o número de vezes em que a verificação de integridade pode falhar até a instância de contêiner ser considerada não íntegra. Caso você tenha um limite não íntegro de 2 e um intervalo de verificação de integridade de 30 segundos, a tarefa tem 60 segundos para responder ao ping de verificação de integridade até ser considerada não íntegra. Você pode aumentar o limite não íntegro ou o intervalo de verificação de integridade para dar às tarefas mais tempo para responder.

Não foi possível atualizar o serviço *`servicename`*: o nome do contêiner do load balancer ou a porta foi alterada na definição da tarefa

Caso o serviço use um load balancer, a configuração do load balancer definido para o serviço quando criado não pode ser alterada. Se você atualizar a definição de tarefa para o serviço, o contêiner e a porta de contêiner especificados quando o serviço foi criado deverão permanecer na definição da tarefa.

Para alterar o load balancer, o nome do contêiner, ou a porta do contêiner associada a uma configuração do load balancer de serviço, você deve criar um novo serviço.

Histórico do documento

A tabela a seguir descreve as principais atualizações e os novos recursos do Guia do usuário do Amazon ECS para AWS Fargate. Também atualizamos a documentação com frequência para abordar os comentários enviados por você.

Alteração	Descrição	Data
FireLens para Amazon ECS	O FireLens para Amazon ECS está em pré-visualização pública. O FireLens para Amazon ECS permite usar parâmetros de definição de tarefa a fim de rotear logs para um serviço da AWS ou destino de um parceiro para armazenamento e análise de logs. Para obter mais informações, consulte Rotear logs personalizados (p. 70).	30 de agosto de 2019
CloudWatch Container Insights	O CloudWatch Container Insights já está disponível. Ele permite coletar, agregar e resumir métricas e logs de seus aplicativos e microsserviços em contêineres. Para obter mais informações, consulte CloudWatch Container Insights do Amazon ECS (p. 186).	30 de agosto de 2019
Expansão da região AWS Fargate	O AWS Fargate com o Amazon ECS foi expandido para a região Ásia-Pacífico (Hong Kong).	06 de agosto de 2019
Registro de vários grupos de destino com um serviço	Adicionado suporte para especificar vários grupos de destino em uma definição de serviço. Para obter mais informações, consulte Registro de vários grupos de destino com um serviço (p. 126).	30 de julho de 2019
CloudWatch Container Insights	O Amazon ECS adicionou suporte para o CloudWatch Container Insights. Para obter mais informações, consulte CloudWatch Container Insights do Amazon ECS (p. 186).	9 de julho de 2019
Permissões no nível do recurso para serviços e conjuntos de tarefas do Amazon ECS	O Amazon ECS expandiu o suporte às permissões no nível do recurso de para serviços e conjuntos de tarefas do Amazon ECS. Para obter mais informações, consulte Como o Amazon Elastic Container Service funciona com o IAM (p. 196).	27 de junho de 2019
Atualização da plataforma AWS Fargate versão 1.3.0	A partir de 1º de maio de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte ao driver de log `splunk`, além do driver de log `awslogs`. Para obter mais informações, consulte Armazenamento e registro em log (p. 44).	1º de maio de 2019
Atualização da plataforma AWS Fargate versão 1.3.0	A partir de 1º de maio de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte à referência a dados confidenciais na configuração de log de um contêiner usando o parâmetro de definição de contêiner `secretOptions`. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).	1º de maio de 2019
Atualização da plataforma AWS Fargate versão 1.3.0	A partir de 2 de abril de 2019, todas as novas tarefas do Fargate que forem iniciadas oferecerão suporte à injeção de dados confidenciais em seus contêineres. Isso é feito armazenando seus dados confidenciais em	2 de abril de 2019

Alteração	Descrição	Data
	segredos do AWS Secrets Manager ou em parâmetros do Parameter Store do AWS Systems Manager e fazendo referência a eles na definição do contêiner. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73).	
Atualização da plataforma AWS Fargate versão 1.3.0	A partir de 27 de março de 2019, todas as novas tarefa do Fargate que forem iniciadas poderão usar parâmetros adicionais de definição de tarefas que permitem que você defina uma configuração de proxy, dependências para inicialização e desligamento de contêiner, além de um valor de tempo limite de início e interrupção por contêiner. Para obter mais informações, consulte Configuração do proxy (p. 56), Dependência de contêiner (p. 51) e Tempos limite de contêiner (p. 52).	27 de março de 2019
O Amazon ECS apresenta o tipo de implantação externa	O tipo de implantação externa permite que você use qualquer controlador de implantação de terceiros para o controle total do processo de implantação para um serviço do Amazon ECS. Para obter mais informações, consulte Implantação externa (p. 111).	27 de março de 2019
O Amazon ECS apresenta a API `PutAccountSettingDefault`	O Amazon ECS apresenta a API `PutAccountSettingDefault` que permite ao usuário definir o status de aceitação do formato padrão de ARN/ID para todos os usuários e funções do IAM na conta. Anteriormente, definir o status de aceitação padrão da conta exigia o uso do usuário raiz. Para obter mais informações, consulte Nomes de recursos da Amazon (ARNs) e IDs (p. 86).	8 de fevereiro de 2019
VPC endpoints de interface (AWS PrivateLink)	Adicionado suporte para a configuração de VPC endpoints de interface desenvolvidos pelo AWS PrivateLink. Isso permite criar uma conexão privada entre sua VPC e o Amazon ECS sem exigir acesso pela Internet por meio de uma instância NAT, de uma conexão VPN ou do AWS Direct Connect. Para obter mais informações, consulte VPC endpoints da interface (AWS PrivateLink). 26 de dezembro de 2018	

Alteração	Descrição	Data
Plataforma AWS Fargate versão 1.3.0	Nova versão da plataforma AWS Fargate lançada, que contém: • Suporte adicionado para o uso dos parâmetros do Parameter Store do AWS Systems Manager para injetar dados confidenciais em seus contêineres. Para obter mais informações, consulte Especificação de dados confidenciais (p. 73). • Adicionada a reciclagem de tarefas para as tarefas Fargate, que é o processo de atualização das tarefas que compõem um serviço do Amazon ECS. Para obter mais informações, consulte Reciclagem de tarefas Fargate (p. 97). Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	17 de dezembro de 2018
Expansão da região AWS Fargate	O AWS Fargate com o Amazon ECS expandiu para as regiões da Ásia-Pacífico (Mumbai) e Canadá (Central).	07 de dezembro de 2018
Implantações azuis/verdes do Amazon ECS	O Amazon ECS adicionou suporte para implantações azuis/verdes usando o CodeDeploy. Esse tipo de implantação permite que você verifique uma nova implantação de um serviço antes de enviar tráfego de produção para ele. Para obter mais informações, consulte Implantação azul/verde com o CodeDeploy (p. 108).	27 de novembro de 2018
Marcação de recursos	O Amazon ECS adicionou suporte para adicionar tags de metadados aos seus serviços, definições de tarefa, tarefas, clusters e instâncias de contêiner. Para obter mais informações, consulte Recursos e tags (p. 165).	15 de novembro de 2018
Expansão da região AWS Fargate	O AWS Fargate com o Amazon ECS foi expandido para as regiões Oeste dos EUA (Norte da Califórnia) e Ásia-Pacífico (Seul). Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	07 de novembro de 2018

Alteração	Descrição	Data
Service limits atualizados	Os seguintes service limits foram atualizados: • O número de tarefas que usam o tipo de inicialização Fargate por região, por conta, foi elevado de `20` para `50`. • O número de endereços IP públicos para tarefas que usam o tipo de inicialização Fargate foi elevado de `20` para `50`. Para obter mais informações, consulte Limites de serviço do Amazon ECS (p. 277).	31 de outubro de 2018
Expansão da região AWS Fargate	O AWS Fargate com o Amazon ECS foi expandido para a região UE (Londres). Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	26 de outubro de 2018
Suporte de autenticação de registro privado para o Amazon ECS usando tarefas AWS Fargate	O Amazon ECS introduziu suporte para tarefas Fargate usando autenticação de registro privado usando AWS Secrets Manager. Esse recurso permite que você armazene suas credenciais de forma segura e, então, referencie-as em sua definição de contêiner, o que permite que suas tarefas usem imagens privadas. Para obter mais informações, consulte Autenticação de registro privado para tarefas (p. 71).	10 de setembro de 2018
CLI do Amazon ECS v1.8.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Adicionado suporte para volumes do Docker em arquivos de composição do Docker. • Adicionado suporte para restrições e estratégias de realização de tarefas em arquivos de composição do Docker. • Adicionado suporte para autenticação do registro privado em arquivos de composição do Docker. • Adicionado suporte para `--force-update` no `compose up` a fim de forçar reinicialização de tarefas. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	7 de setembro de 2018
Expansão da região de descoberta do serviço do Amazon ECS	A descoberta do serviço do Amazon ECS expandiu suporte para as regiões Ásia-Pacífico (Cingapura), Ásia-Pacífico (Sydney), Ásia-Pacífico (Tóquio), UE (Frankfurt) e UE (Londres). Para obter mais informações, consulte Descoberta de serviço (p. 137).	30 de agosto de 2018

Alteração	Descrição	Data
Tarefas programadas com suporte a tarefas Fargate	O Amazon ECS introduziu suporte para tarefas programadas para o tipo de inicialização Fargate. Para obter mais informações, consulte Tarefas programadas (`cron`) (p. 93).	28 de agosto de 2018
Expansão da região AWS Fargate	O AWS Fargate com o Amazon ECS foi expandido para as regiões da UE (Frankfurt), Ásia-Pacífico (Cingapura) e Ásia-Pacífico (Sydney). Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	19 de julho de 2018
CLI do Amazon ECS v1.7.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Adicionado suporte para os contêineres `healthcheck` e `devices` nos arquivos de composição do Docker. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	18 de julho de 2018
Estratégias do programador de serviço do Amazon ECS adicionadas	O Amazon ECS introduziu o conceito de estratégias de programador de serviço. Há duas estratégias de programador de serviços disponíveis: • `REPLICA` — A estratégia de programação de réplica posiciona e mantém o número desejado de tarefas no seu cluster. Por padrão, o programador de serviços distribui as tarefas nas zonas de disponibilidade. Você pode usar estratégias de posicionamento de tarefas e restrições para personalizar as decisões de posicionamento de tarefas. Para obter mais informações, consulte Réplica (p. 100). • `DAEMON` — A estratégia de programação do daemon implantará exatamente uma tarefa em cada instância de contêiner ativa que atender a todas as restrições de posicionamento de tarefas que você especificar no seu cluster. Ao usar essa estratégia, não há necessidade de especificar um número desejado de tarefas, uma estratégia de posicionamento de tarefas ou usar políticas de Auto Scaling do serviço. Para obter mais informações, consulte Daemon (p. 100). Note As tarefas Fargate não são compatíveis com a estratégia de programação do `DAEMON`. Para obter mais informações, consulte Conceitos do programador de serviço (p. 99).	12 de junho de 2018

Alteração	Descrição	Data
CLI do Amazon ECS v1.6.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Suporte adicionado para a Sintaxe do arquivo de composição do Docker versão 3. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	5 de junho de 2018
Expansão da região AWS Fargate	AWS Fargate com o Amazon ECS se expandiu para as regiões Leste dos EUA (Ohio), Oeste dos EUA (Oregon) e Oeste da UE (Irlanda). Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	26 de abril de 2018
CLI do Amazon ECS v1.5.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Adicionado suporte à CLI do ECS para recuperar automaticamente a última AMI estável otimizada para Amazon ECS consultando a API do Parameter Store do Systems Manager durante o processo de criação de recurso de cluster. Isso requer que a conta de usuário sendo utilizada tenha as permissões do Systems Manager. • Adicionado suporte para os parâmetros `shm_size` e `tmpfs` nos arquivos de composição. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	19 de abril de 2018
Verificação de download da CLI do Amazon ECS	Adicionado novo método de assinatura PGP para verificar o arquivo de instalação da CLI do Amazon ECS. Para obter mais informações, consulte Como instalar a CLI do Amazon ECS (p. 248).	5 de abril de 2018
Versão da plataforma AWS Fargate	Nova versão da plataforma AWS Fargate lançada, que contém: • Adicionado o suporte para Endpoint de metadados de tarefas do Amazon ECS (p. 272). • Adicionado o suporte para Verificação de integridade (p. 40). • Adicionado o suporte para Descoberta de serviço (p. 137) Para obter mais informações, consulte Versões de plataforma do AWS Fargate (p. 20).	26 de março de 2018

Alteração	Descrição	Data
Descoberta de serviço do Amazon ECS	Inclusão de integração com o Route 53 para oferecer suporte à descoberta de serviço do Amazon ECS. Para obter mais informações, consulte Descoberta de serviço (p. 137).	22 de março de 2018
CLI do Amazon ECS v1.4.2	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • AMI atualizada para `amzn-ami-2017.09.k-amazon-ecs-optimized`. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	20 de março de 2018
CLI do Amazon ECS v1.4.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Inclusão de suporte à região us-gov-west-1. • Adição do sinalizador `--force-deployment` para o comando do serviço composto. • Inclusão de suporte a `aws_session_token` nos perfis do ECS. • AMI atualizada para `amzn-ami-2017.09.j-amazon-ecs-optimized`. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	09 de março de 2018
Verificações de integridade do contêiner	Inclusão de suporte às verificações de integridade do Docker nas definições de contêiner. Para obter mais informações, consulte Verificação de integridade (p. 40).	08 de março de 2018
Endpoint de metadados de tarefas do Amazon ECS	A partir da versão 1.17.0 do agente de contêiner do Amazon ECS, vários metadados de tarefas e estatísticas do Docker estão disponíveis para as tarefas que usam o modo de rede `awsvpc` em um endpoint HTTP fornecido pelo agente de contêiner do Amazon ECS. Para obter mais informações, consulte Endpoint de metadados de tarefas do Amazon ECS (p. 272).	8 de fevereiro de 2018
Auto Scaling de serviços do Amazon ECS usando políticas de rastreamento de destino	Adição de suporte ao Auto Scaling de serviços do ECS usando políticas de rastreamento de destino no console do Amazon ECS. Para obter mais informações, consulte Políticas de escalabilidade de rastreamento de destino (p. 130). Remoção do tutorial anterior de escalabilidade em etapas do assistente de primeira execução do ECS. Ele foi substituído pelo novo tutorial de rastreamento de destino.	8 de fevereiro de 2018

Alteração	Descrição	Data
CLI do Amazon ECS v1.3.0	Nova versão da CLI do Amazon ECS emitida, que adicionou o seguinte recurso: • Capacidade de criar clusters vazios com o comando up. • Adição do sinalizador `--health-check-grace-period` para o comando up do serviço composto. • AMI atualizada para `amzn-ami-2017.09.g-amazon-ecs-optimized`. Para obter mais informações, consulte Referência de linha de comando do Amazon ECS no Amazon Elastic Container Service Developer Guide.	19 de janeiro de 2018
Novo comportamento do programador de serviços	Informações atualizadas sobre o comportamento de tarefas de serviços que não puderam ser iniciadas. Documentada nova mensagem de evento de serviço que é acionada quando uma tarefa de serviço tem falhas consecutivas. Para obter mais informações sobre esse comportamento atualizado, consulte Conceitos de serviços adicionais (p. 100).	11 de janeiro de 2018
CPU e memória em nível de tarefa	Adicionado suporte para especificação de CPU e memória em nível de tarefa nas definições de tarefas. Para obter mais informações, consulte TaskDefinition.	12 de dezembro de 2017
Integração do CodePipeline do console do Amazon ECS	Adicionada integração do Amazon ECS com o CodePipeline. O CodePipeline é compatível com o Amazon ECS como uma opção de implantação para ajudar a configurar pipelines de implantação. Para obter mais informações, consulte Tutorial: implantação contínua com o CodePipeline (p. 305).	12 de dezembro de 2017
Função de execução de tarefas	O agente de contêiner do Amazon ECS faz chamadas para as ações de API do Amazon ECS em seu nome. Dessa forma, exige uma política e uma função do IAM para o serviço saber que o agente pertence a você. As ações a seguir estão cobertas pela tarefa função de execução: • Chamadas para o Amazon ECR obter a imagem de contêiner • Chamadas para o CloudWatch armazenar os logs do aplicativo do contêiner Para obter mais informações, consulte Função do IAM da execução de tarefas do Amazon ECS (p. 228).	7 de dezembro de 2017

Alteração	Descrição	Data
CLI do Amazon ECS com suporte para Fargate v1.1.0	Nova versão da CLI do Amazon ECS emitida, que adicionou os seguintes recursos: • Suporte para redes de tarefas • Suporte para AWS Fargate • Suporte para visualizar dados do CloudWatch Logs a partir de uma tarefa Para obter mais informações, consulte ECS CLI changelog.	29 de novembro de 2017
GA do AWS Fargate	Adicionado suporte para iniciar serviços do Amazon ECS usando o tipo de inicialização Fargate. Para obter mais informações, consulte Tipos de inicialização Amazon ECS (p. 59).	29 de novembro de 2017

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the AWS General Reference.