



Manual do usuário

AWS Batch



AWS Batch: Manual do usuário

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas e imagens comerciais da Amazon não podem ser usadas em conexão com nenhum produto ou serviço que não seja da Amazon, de qualquer maneira que possa causar confusão entre os clientes, que menospreze ou desacredite a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é o AWS Batch?	1
Você é um usuário iniciante do AWS Batch?	3
Serviços relacionados	4
Como acessar o AWS Batch	4
Componentes do AWS Batch	5
Ambiente computacional	5
Filas de tarefas	6
Definições de trabalho	6
Tarefas	6
Política de agendamento	6
Recursos consumíveis	7
Ambiente de serviço	7
Trabalhos de serviço	7
Configurar o AWS Batch	8
Criar conta de IAM e usuário administrativo	8
Inscreva-se para uma Conta da AWS	8
Criar um usuário com acesso administrativo	9
Criar perfil do IAM	10
Criar um par de chaves	11
Crie uma VPC	13
Criar um grupo de segurança	14
Instalar o AWS CLI	16
Tutoriais de conceitos básicos	17
Conceitos básicos do Amazon EC2 usando o assistente	17
Visão geral	17
Pré-requisitos	18
Etapa 1: criar um ambiente computacional	18
Etapa 2: criar uma fila de trabalhos	19
Etapa 3: Criar uma definição de trabalho	20
Etapa 4: Criar um trabalho	20
Etapa 5: analisar e criar	20
Etapa 6: ver a saída do trabalho	21
Etapa 7: limpar os recursos do tutorial	21
Recursos adicionais	22

Conceitos básicos de orquestração do Fargate usando o assistente	22
Visão geral do	22
Pré-requisitos	23
Etapa 1: criar um ambiente computacional	23
Etapa 2: criar uma fila de trabalhos	24
Etapa 3: criar uma definição de tarefa	24
Etapa 4: Criar um trabalho	25
Etapa 5: analisar e criar	26
Etapa 6: ver a saída do trabalho	26
Etapa 7: limpar os recursos do tutorial	26
Recursos adicionais do	27
Introdução ao Fargate AWS Batch e ao Fargate usando o AWS CLI	27
Pré-requisitos	27
Criar um perfil de execução do IAM	28
Crie um ambiente de computação	29
Crie uma fila de trabalhos	30
Crie uma definição de trabalho	31
Enviar e monitorar um trabalho	32
Ver a saída de trabalho	33
Limpar os recursos	34
Transição para o ambiente de produção	35
Próximas etapas	36
Conceitos básicos do Amazon EKS	37
Visão geral	38
Pré-requisitos	39
Etapa 1: criar o cluster do Amazon EKS para AWS Batch	40
Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch	40
Etapa 3: criar um ambiente computacional do Amazon EKS	44
Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação	46
Etapa 5: Criar uma definição de trabalho	47
Etapa 6: Enviar um trabalho	48
Etapa 7: ver a saída do trabalho	48
Etapa 8: (Opcional) Enviar um trabalho com substituições	49
Etapa 9: limpar os recursos do tutorial	50
Recursos adicionais	51
Conceitos básicos do AWS Batch em clusters privados do Amazon EKS	51

Visão geral	38
Pré-requisitos	54
Etapa 1: criar seu cluster EKS para AWS Batch	55
Etapa 2: preparando seu cluster EKS para o AWS Batch	57
Etapa 3: criar um ambiente computacional do Amazon EKS	60
Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação	62
Etapa 5: criar um Amazon ECR com cache pull through	63
Etapa 6: registrar uma definição de trabalho	64
Etapa 7: enviar um trabalho para execução	65
Etapa 8: ver a saída do trabalho	65
Etapa 9: (Opcional) Enviar um trabalho com substituições	65
Etapa 10: limpar os recursos do tutorial	66
Recursos adicionais	51
Solução de problemas	67
Começando a usar AWS Batch a SageMaker IA	68
Visão geral do	68
Pré-requisitos	69
Etapa 1: criar uma função de execução de SageMaker IA	70
Etapa 2: criar seu ambiente de serviço	71
Etapa 3: Crie sua fila SageMaker de trabalhos	73
Etapa 4: criar e enviar um trabalho de treinamento	74
Etapa 5: monitorar o status do trabalho	76
Etapa 6: ver a saída do trabalho	78
Etapa 7: limpar os recursos do tutorial	81
Recursos adicionais do	81
O painel do widget AWS Batch	83
Adicionar o widget Fila de trabalho única	83
Como adicionar o widget do CloudWatch Container Insights	84
Adicionar o widget de logs de trabalho	84
Ambientes computacionais para AWS Batch	86
Ambientes de computação gerenciados	86
Consideração ao criar trabalhos paralelos de vários nós	89
Ambientes de computação não gerenciados	89
Crie um ambiente de computação	91
Tutorial: criar um ambiente de computação gerenciado usando recursos do Fargate	92
Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EC2	94

Tutorial: criar um ambiente de computação não gerenciado usando recursos do Amazon EC2	103
Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EKS ..	104
Tutorial: Crie um ambiente computacional não gerenciado usando os recursos do Amazon EKS	110
Recurso: modelo de ambiente de computação	111
Tabela de computação do tipo de instância	113
Atualizar um ambiente computacional	116
Estratégias de atualização do ambiente computacional	117
Escolhendo a estratégia de atualização correta	118
Executar atualizações de escalabilidade	120
Realizar atualizações da infraestrutura	122
Realizar atualizações azul/verde	127
Recursos de computação de AMIs	132
Especificação da AMI do recurso de computação	135
Tutorial: criar uma AMI de recurso de computação	136
Usar uma AMI de workload de GPU	140
Depreciação do Amazon Linux	146
Descontinuação da AMI Amazon EKS Amazon Linux 2	146
Descontinuação da AMI Amazon ECS Amazon Linux 2	147
Usar modelos de execução do Amazon EC2	147
Modelos de execução padrão e substituídos	149
Dados de usuário do Amazon EC2 em modelos de execução	150
Referência: exemplos de modelos de inicialização	152
Configuração do serviço de metadados de instância (IMDS)	154
Cenários de configuração	154
Configurações do EC2	156
Como migrar do ECS AL2 para o ECS AL2023	157
Estratégias de alocação de tipos de instâncias	159
Gerenciamento de memória	161
Reservar memória do sistema	162
Tutorial: visualizar memória de recursos de computação	163
Considerações sobre memória e vCPU para AWS Batch no Amazon EKS	163
Ambientes de computação Fargate	169
Quando usar o Fargate	169
Definições de trabalho no Fargate	170

Filas de trabalhos no Fargate	172
Ambientes de computação no Fargate	172
Ambientes de computação Amazon EKS	173
Amazon EKS	176
AMI padrão do Amazon EKS	176
Ambientes AMI mistos	178
Versões do Kubernetes com suporte	179
Atualizar a versão Kubernetes do ambiente de computação	180
Responsabilidade compartilhada dos nós Kubernetes	180
Execute um DaemonSet em nós AWS Batch gerenciados	181
Personalize os modelos de lançamento do Amazon EKS	182
Como fazer o upgrade do EKS AL2 para o EKS AL2023	186
Ambientes de serviço	189
O que são ambientes de serviço	189
Como os ambientes de serviço funcionam com outros AWS Batch componentes	190
Práticas recomendadas para ambientes de serviço	190
Estados do ambiente de serviço e ciclo de vida	191
Definições do estado do ambiente de serviço	192
Criar um ambiente de serviço	194
Pré-requisitos	194
Atualizar um ambiente de serviço	196
Excluir um ambiente de serviço	198
Pré-requisitos de exclusão	198
Filas de tarefas	201
Crie uma fila de trabalhos	201
Criar uma fila de trabalhos Amazon EC2	202
Criar uma fila de trabalhos Fargate	203
Criar uma fila de trabalhos Amazon EKS	204
Criar uma fila de trabalhos do SageMaker	205
Modelo de fila de trabalhos	208
Exibir uma fila de trabalhos	209
Visualizar informações sobre a fila de trabalhos	209
Excluir uma fila de trabalhos	211
Políticas de agendamento de compartilhamento justo	212
Usar identificadores de compartilhamento	213
Usar políticas de agendamento	214

Usar o agendamento de compartilhamento justo	214
Tutorial: criar uma política de agendamento	215
Referência: modelo de política de agendamento	217
Agendamento com reconhecimento de recurso	218
Criar recursos consumíveis	219
Especificar recursos para um trabalho	220
Verificar o uso de recursos	221
Atualizar quantidades de recursos em uso	223
Encontrar trabalhos que exigem um recurso consumível	224
Excluir um recurso consumível	225
Definições de trabalho	226
Criar uma definição de tarefa de nó único	226
Tutorial: criar uma definição de trabalho de nó único nos recursos do Amazon EC2	227
Criar uma definição de trabalho de nó único nos recursos do Fargate	233
Criar uma definição de trabalho de nó único nos recursos do Amazon EKS	239
Criar uma definição de trabalho de nó único com vários contêineres nos recursos do Amazon EC2	244
Criar uma definição de trabalho paralelo de vários nós	251
Tutorial: criar uma definição de trabalho em paralelo de vários nós nos recursos do Amazon EC2	252
Referência: modelo de definição de trabalho que usa ContainerProperties	259
Parâmetros de definição de trabalho para ContainerProperties	267
Crie definições de trabalhos usando EcsProperties	312
Definições de trabalho ContainerProperties versus EcsProperties	312
Alterações gerais nas APIs do AWS Batch	313
Definições de trabalho de vários contêineres para o Amazon ECS	314
Definições de trabalho de vários contêineres para o Amazon EKS	315
Referência: cenários de trabalho do AWS Batch usando EcsProperties	316
Usar o driver de log awslogs.	322
Opções do driver de log awslogs no tipo de dados AWS Batch JobDefiniton	323
Especificar uma configuração de log na definição de trabalho	326
Especificar dados confidenciais	327
Uso do Secrets Manager	328
Usar o Systems Manager Parameter Store	336
Autenticação de registro privado para trabalhos	340
Permissões do IAM necessárias para a autenticação de registro privado	341

Tutorial: criar um segredo para a autenticação de registro privado	342
Volumes Amazon EFS	343
Considerações de volume Amazon EFS	344
Uso dos pontos de acesso do Amazon EFS	345
Especificação de um sistema de arquivos do Amazon EFS na definição de trabalho	346
Exemplo de definição de trabalhos	349
Variáveis de ambiente	349
Substituição de parâmetros	350
Testar funcionalidade de GPU	351
Trabalho em paralelo de vários nós	352
Tarefas	354
Tutorial: enviar um trabalho	355
Trabalhos de serviço	358
Cargas úteis de trabalho de serviço	359
Enviar um trabalho de serviço	360
Status de serviço do trabalho	362
Estratégias de nova tentativa de trabalho de serviço	363
Monitorar trabalhos de serviço em uma fila	366
Encerrar trabalhos de serviço	368
Estados do trabalho	368
Variáveis de ambiente de trabalho	372
Repetições de trabalho automatizadas	373
Dependências do trabalho	375
Tempos limite de trabalho	375
Trabalhos do Amazon EKS	377
Tutorial: mapear um trabalho em execução para um pod e um nó	378
Tutorial: mapear um pod em execução de volta ao seu trabalho	379
Trabalhos em paralelo de vários nós	380
Variáveis de ambiente	382
Grupos de nós	382
Ciclo de vida do trabalho	383
Considerações sobre o ambiente de computação	384
Trabalhos paralelos de vários nós no Amazon EKS	385
Execução de trabalhos de MNP	385
Criar uma definição de trabalho de GPU do Amazon EKS	387
Enviar um trabalho MNP do Amazon EKS	390

Substituir uma definição de trabalho MNP do Amazon EKS	390
Trabalhos de matriz	391
Exemplo de um fluxo de trabalho de matriz	394
Usando o índice de trabalhos de matriz	397
Executar trabalhos de GPU	403
Criar um cluster do Kubernetes baseado em GPU no Amazon EKS	408
Criar uma definição de trabalho de GPU do Amazon EKS	410
Executar um trabalho de GPU em seu Cluster do Amazon EKS	411
Exibir trabalhos em uma fila de trabalho	411
Pesquisar por trabalhos em uma fila de trabalho	412
Pesquisar AWS Batch empregos (AWS Console)	412
Pesquisar e filtrar AWS Batch vagas (AWS CLI)	414
Modos de rede para AWS Batch trabalhos	416
Exibir registros de trabalhos em CloudWatch Registros	417
Revise as informações do AWS Batch trabalho	418
Segurança em AWS Batch	420
Gerenciamento de Identidade e Acesso	421
Público	421
Autenticação com identidades	421
Gerenciar o acesso usando políticas	423
Como AWS Batch funciona com o IAM	425
Exemplos de políticas baseadas em identidade	429
AWS políticas gerenciadas	432
Políticas do IAM, perfis e permissões	437
Estrutura da política do IAM	438
Recurso: exemplos de políticas	442
Recurso: política AWS Batch gerenciada	452
AWS Batch Função de execução do IAM	452
Permissões compatíveis em nível de recurso	454
Tutorial: criar o perfil de execução do IAM	454
Tutorial: verificar a perfil de execução do IAM	455
Usar perfis vinculados a serviço	456
Perfil de instância do Amazon ECS	470
Perfil de frota spot Amazon EC2	473
EventBridge IAM role (Função do IAM)	477
Criar uma nuvem privada virtual	478

Crie uma VPC	479
Próximas etapas	479
Endpoints da VPC	480
Considerações	480
Como criar um endpoint de interface	482
Criar uma política de endpoint	483
Validação de conformidade	484
Segurança da infraestrutura	484
Prevenção do problema "confused deputy" entre serviços	485
Exemplo: perfil para acessar apenas um ambientes de computação	486
Exemplo: perfil para acessar vários ambientes de computação	486
CloudTrail	487
AWS Batch informações em CloudTrail	488
Referência: Entendendo as entradas do arquivo de AWS Batch log	489
Solucionar problemas do IAM AWS Batch	490
Não tenho autorização para executar uma ação no AWS Batch	491
Não estou autorizado a realizar iam: PassRole	491
Quero permitir que pessoas fora da minha AWS conta acessem meus AWS Batch recursos	492
AWS Step Functions	493
Tutorial: visualizar detalhes da máquina de estado	493
Tutorial: editar uma máquina de estado	494
Tutorial: executar uma máquina de estado	494
Amazon EventBridge	496
AWS BatchEventos do	497
Eventos de alteração de estado do trabalho	497
Eventos bloqueados da fila de trabalhos	499
Eventos de alteração de estado do trabalho de serviço	500
Eventos bloqueados da fila de trabalhos de serviço	502
Tutorial: usar notificações de usuário da AWS com o AWS Batch	503
Trabalhos AWS Batch como destinos do EventBridge	504
Tutorial: criar um trabalho programado	505
Tutorial: criar uma regra com um padrão de evento	508
Tutorial: transformador de entrada de passagem	510
Tutorial: recepcionar eventos do trabalho do AWS Batch	513
Pré-requisitos	513

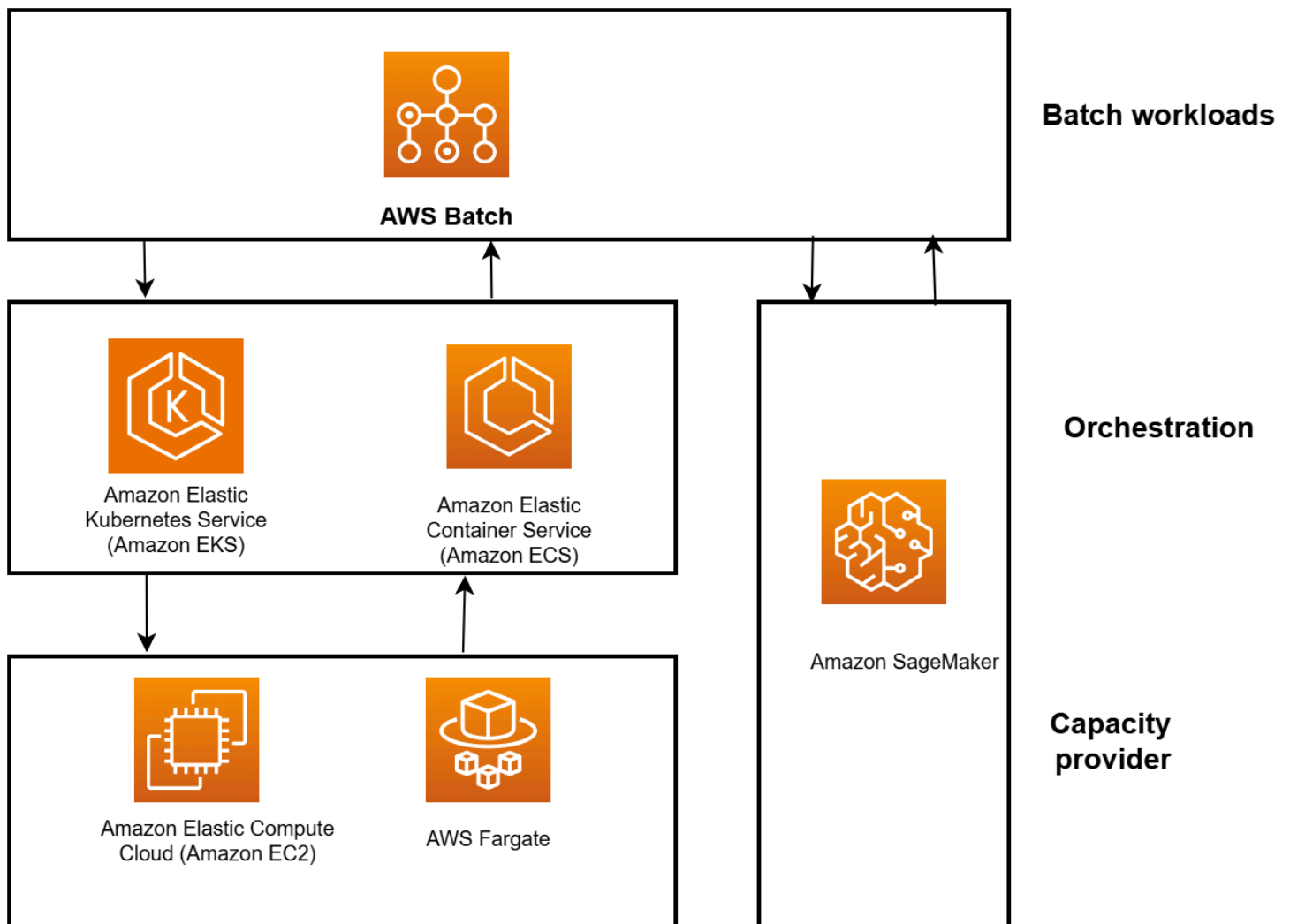
Tutorial: criar a função do Lambda	514
Tutorial: registrar a regra do evento	515
Tutorial: testar sua configuração	517
Tutorial: Como enviar alertas do Amazon Simple Notification Service para eventos de trabalho que falharam	517
Pré-requisitos	517
Tutorial: criar e assinar um tópico do Amazon SNS	518
Tutorial: registrar uma regra de evento	518
Tutorial: testar sua regra	520
Regra alternativa: fila de trabalhos em lote bloqueada	520
Elastic Fabric Adapter	522
Monitorar AWS Batch	525
CloudWatch Registros	526
Tutorial: Adicionar uma política do CloudWatch Logs IAM	526
Instalar e configurar o CloudWatch agente	528
Tutorial: Exibir CloudWatch registros	529
CloudWatch Container Insights	531
Ativar o Container Insights	531
Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS	532
Pré-requisitos	533
Instalar o complemento	533
Marcar com tag os recursos do	534
Conceitos Básicos de Tags	534
Marcar com tag os recursos do	535
Restrições de tags	536
Tutorial: gerenciar etiquetas usando o console	537
Adicionar etiquetas a um recurso individual na criação	537
Adicionar e excluir tags em um recurso individual	537
Gerenciar etiquetas usando a CLI ou a API	538
Melhores práticas	540
Quando utilizar AWS Batch	540
Lista de verificação para execução em escala	541
Otimize contêineres e AMIs	542
Escolha a opção direita de recurso de ambiente de computação	543
Amazon EC2 Sob Demanda, ou Amazon EC2 Spot	544
Use as práticas recomendadas do Amazon EC2 Spot para AWS Batch	545

Erros Comuns e Solução de Problemas	547
Solução de problemas	551
AWS Batch	552
Configuração ideal do tipo de instância para receber atualizações automáticas da família de instâncias	553
Ambiente de computação do INVALID	553
Trabalhos presos no status RUNNABLE	556
Instâncias spot sem tags na criação	562
As instâncias spot não estão diminuindo	562
Não é possível recuperar segredos do Secrets Manager	564
Não é possível substituir os requisitos de recursos de definição de trabalho	564
Mensagem de erro ao atualizar a configuração desiredvCpus	566
AWS Batch no Amazon EKS	566
Ambiente de computação do INVALID	567
AWS Batch no Amazon EKS, o trabalho está preso no status RUNNABLE	570
AWS Batch no Amazon EKS, o trabalho está preso no status STARTING	571
Verifique se o aws-auth ConfigMap está configurado corretamente	572
As permissões ou vinculações do RBAC não estão configuradas corretamente	573
Recursos: Service Quotas	576
Histórico do documento	578
.....	dlxxxv

O que é o AWS Batch?

AWS Batch O ajuda a executar workloads de computação em lote na Nuvem AWS. A computação em lote é uma maneira comum para desenvolvedores, cientistas e engenheiros acessarem grandes quantidades de recursos de computação. O AWS Batch remove as tarefas gerais de configuração e gerenciamento da infraestrutura necessária, semelhante ao software de computação em lote tradicional. Esse serviço pode fornecer recursos com eficiência em resposta a trabalhos enviados para eliminar restrições de capacidade, reduzir os custos de computação e entregar resultados rapidamente.

Como um serviço totalmente gerenciado, o AWS Batch ajuda a executar workloads de computação em lote de qualquer escala. O AWS Batch provisiona automaticamente recursos de computação e otimiza a distribuição da workload com base na quantidade e na escala das workloads. Com o AWS Batch, não há necessidade de instalar ou gerenciar o software de computação em lote, para que você possa concentrar seu tempo na análise de resultados e na resolução de problemas.

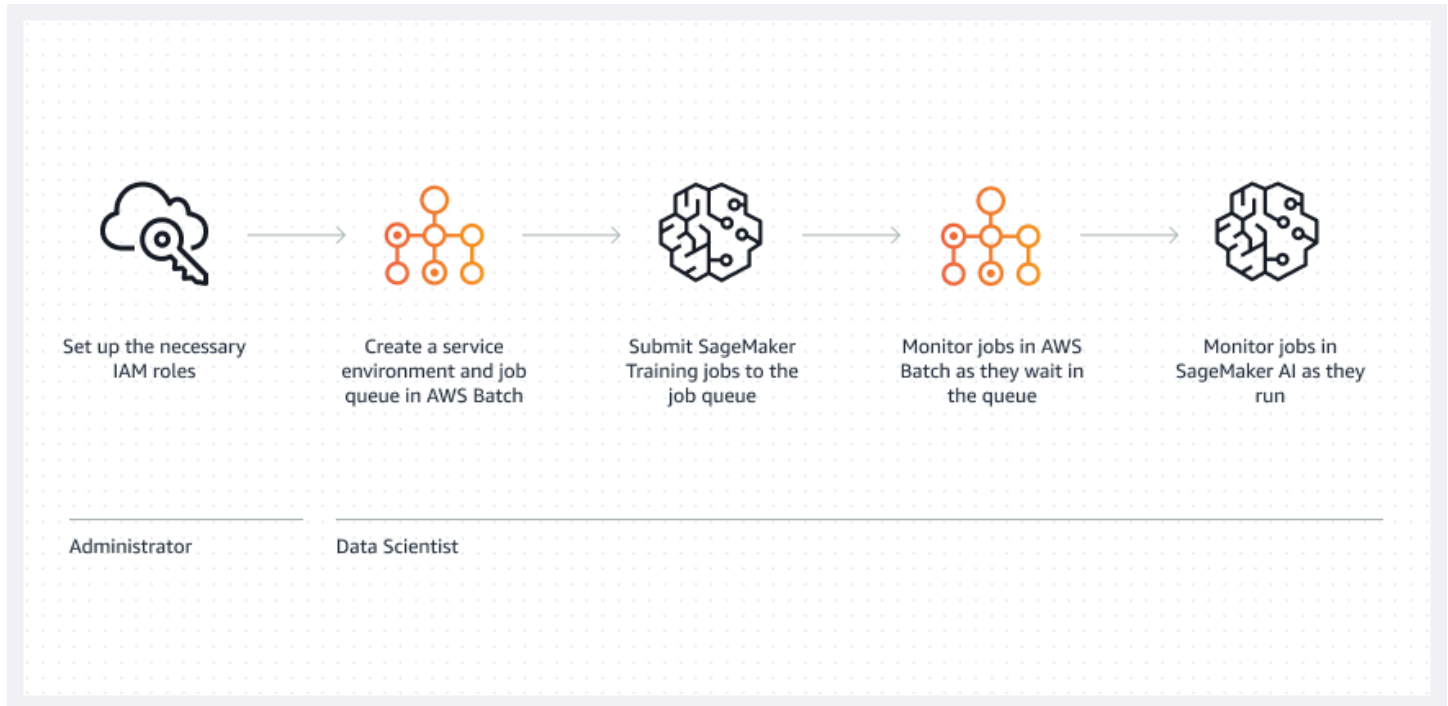


AWS Batch fornece toda a funcionalidade necessária para executar workloads de alta escala e com uso intensivo de computação, além dos serviços gerenciados de orquestração de contêineres AWS, Amazon ECS e Amazon EKS. AWS Batch é capaz de escalar a capacidade de computação em instâncias do Amazon EC2 e dos recursos do Fargate.

AWS Batch fornece um serviço totalmente gerenciado para workloads em lote e fornece os recursos operacionais para otimizar esses tipos de workloads em termos de throughput, velocidade, eficiência de recursos e custo.

AWS Batch também ativa o enfileiramento de trabalhos do SageMaker Training, permitindo que cientistas de dados e engenheiros de ML enviem trabalhos de treinamento com prioridades para filas configuráveis. Esse recurso garante que os workloads de ML sejam executados automaticamente assim que os recursos estiverem disponíveis, eliminando a necessidade de coordenação manual e melhorando a utilização dos recursos.

Para workloads de machine learning, AWS Batch fornece recursos de enfileiramento para trabalhos do SageMaker Training. Você pode configurar filas com políticas específicas para otimizar o custo, o desempenho e a alocação de recursos para seus workloads de treinamento de ML.



Isso fornece um modelo de responsabilidade compartilhada em que os administradores configuram a infraestrutura e as permissões, enquanto os cientistas de dados podem se concentrar em enviar e monitorar seus workloads de treinamento de ML. Os trabalhos são automaticamente colocados em fila e executados com base nas prioridades configuradas e na disponibilidade dos recursos.

Você é um usuário iniciante do AWS Batch?

Se estiver usando o AWS Batch pela primeira vez, recomendamos que você leia as seguintes seções para começar:

- [Componentes do AWS Batch](#)
- [Criar conta de IAM e usuário administrativo](#)
- [Configurar o AWS Batch](#)
- [Introdução aos AWS Batch tutoriais](#)
- [Começando a usar AWS Batch a SageMaker IA](#)

Serviços relacionados

AWS Batch é um serviço de computação em lote totalmente gerenciado que planeja, programa e executa seus workloads de ML, simulação e análise em lotes em contêineres em toda a gama de ofertas de computação AWS, como Amazon ECS, Amazon EKS, AWS Fargate e instâncias spot ou sob demanda. Para obter mais informações sobre cada serviço de computação gerenciado, consulte:

- [Guia do usuário do Amazon EC2](#)
- [AWS Fargate Guia do desenvolvedor do](#)
- [Guia do usuário do Amazon EKS](#)
- [Amazon SageMaker AI Guia do desenvolvedor do](#)

Como acessar o AWS Batch

Você pode acessar o AWS Batch usando o seguinte:

AWS Batch Console do

A interface da web na qual você cria e gerencia recursos.

AWS Command Line Interface

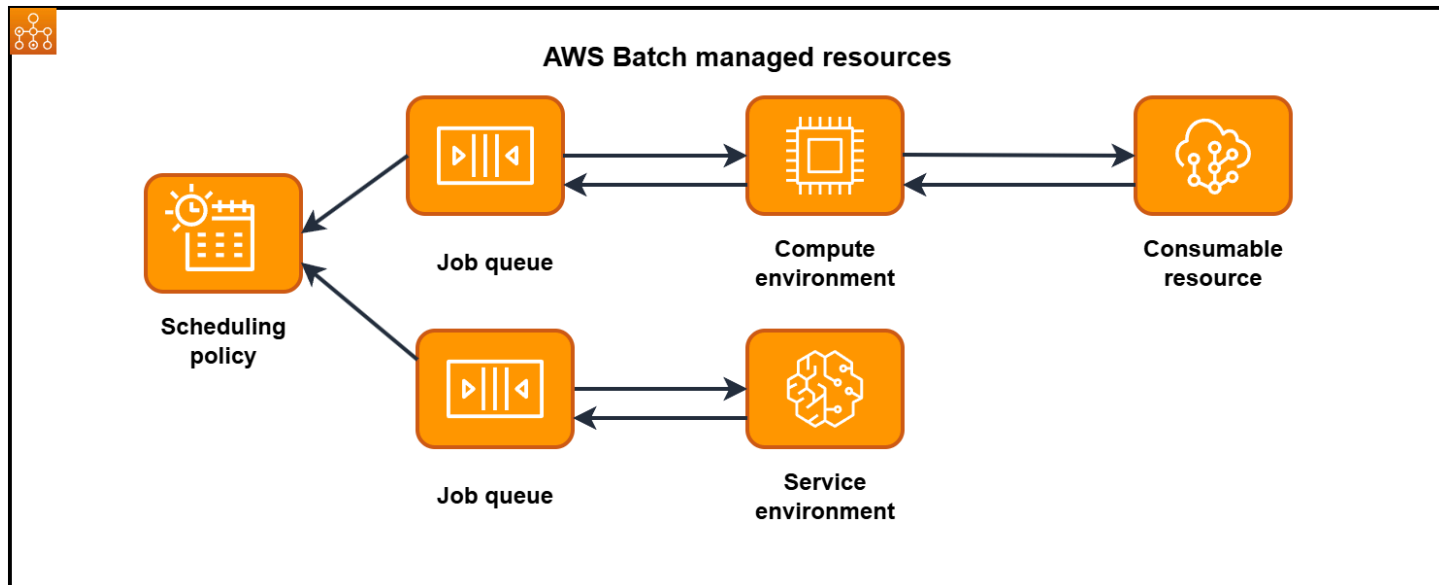
Interagir com o Serviços da AWS usando comandos no shell da linha de comando. A AWS Command Line Interface é compatível com Windows, macOS e Linux. Para obter mais informações sobre a AWS CLI, consulte o [Guia do usuário da AWS Command Line Interface](#). Você encontra os comandos do AWS Batch na [AWS CLI Command Reference](#).

AWS SDKs

Se você preferir criar aplicações usando APIs específicas de uma linguagem em vez de enviar uma solicitação via HTTP ou HTTPS, use as bibliotecas, código de exemplo, tutoriais e outros recursos para desenvolvedores de software fornecidos pela AWS. Essas bibliotecas fornecem funções básicas que automatizam tarefas como a assinatura criptografada das solicitações, novas tentativas de solicitações e tratamento das respostas de erro. Essas funções tornam seu começo mais eficiente. Para obter mais informações, consulte [Ferramentas para criar na AWS](#).

Componentes do AWS Batch

O AWS Batch simplifica a execução de trabalhos em lote em várias zonas de disponibilidade dentro de uma região. Você pode criar ambientes computacionais do AWS Batch em uma VPC nova ou existente. Depois que um ambiente computacional estiver ativo e associado a uma fila de trabalho, você poderá fazer as definições de trabalho que especificam quais imagens de contêineres do Docker executarão seus trabalhos. As imagens de contêiner são armazenadas em registros de contêiner e extraídas deles, que podem existir dentro ou fora de sua infraestrutura da AWS.



Ambiente computacional

Um ambiente computacional é um conjunto de recursos computacionais gerenciados ou não gerenciados que são usados para executar trabalhos. Com ambientes de computação gerenciados, você pode especificar o tipo de computação desejado (Fargate ou EC2) em vários níveis de detalhe. Você pode configurar ambientes computacionais que usem um determinado tipo de instância EC2, um modelo específico, como `c5.2xlarge` ou `m5.10xlarge`. Ou você pode optar apenas por especificar que deseja usar os tipos de instância mais recentes. Também é possível especificar os números mínimo, desejado e máximo de vCPUs para o ambiente com a quantia que você está disposto a pagar por uma instância spot como uma porcentagem do preço da instância sob demanda e um conjunto de destino de sub-redes da VPC. O AWS Batch inicia, gerencia e encerra com eficiência os tipos de computação conforme necessário. Você também pode gerenciar seus próprios ambientes computacionais. Nesse caso, você é responsável por configurar e escalar as instâncias em um cluster do Amazon ECS que o AWS Batch cria para você. Para obter mais informações, consulte [Ambientes computacionais para AWS Batch](#).

Filas de tarefas

Quando enviar um trabalho do AWS Batch, você o enviará para uma fila de trabalho específica, onde ele fica até que seja programado em um ambiente computacional. Você pode associar um ou mais ambientes de computação a uma fila de trabalhos. Você também pode atribuir valores de prioridade para esses ambientes de computação e até mesmo para as próprias filas de trabalhos. Por exemplo, você pode ter uma fila de alta prioridade à qual você envia trabalhos urgentes e uma fila de baixa prioridade para trabalhos que podem ser executados a qualquer momento quando os recursos computacionais forem mais baratos. Para obter mais informações, consulte [Filas de tarefas](#).

Definições de trabalho

As definições de trabalhos especificam como os trabalhos devem ser executados. Você pode pensar em uma definição de trabalho como um esquema para os recursos em seu trabalho. Você pode fornecer ao seu trabalho um perfil do IAM para fornecer acesso a outros recursos da AWS. Você também especifica os requisitos de memória e CPU. A definição de trabalho também pode controlar as propriedades do contêiner, as variáveis de ambiente e pontos de montagem para armazenamento persistente. Muitas das especificações em uma definição de trabalho podem ser substituídas especificando-se novos valores ao enviar trabalhos individuais. Para obter mais informações, consulte [Definições de trabalho](#)

Tarefas

Uma unidade de trabalho (como um script de shell, um executável do Linux ou uma imagem de contêiner do Docker) que você envia para o AWS Batch. Ela tem um nome e é executada como um aplicativo em contêineres no AWS Fargate ou em recursos do Amazon EC2 no seu ambiente de computação usando parâmetros que você especifica em uma definição de trabalho. Os trabalhos podem fazer referência a outros trabalhos por nome ou por ID e podem ser dependentes da conclusão bem-sucedida de outros trabalhos ou da disponibilidade dos [recursos](#) que você especificar. Para obter mais informações, consulte [Tarefas](#).

Política de agendamento

Você pode usar políticas de agendamento para configurar como os recursos de computação em uma fila de trabalhos serão alocados entre usuários ou workloads. Usando políticas de agendamento de compartilhamento justo, você pode atribuir diferentes identificadores de compartilhamento a workloads ou usuários. O agendador de trabalhos AWS Batch usará como padrão uma estratégia

FIFO (por ordem de entrada). Para obter mais informações, consulte [Políticas de agendamento de compartilhamento justo](#).

Recursos consumíveis

Um recurso consumível é um recurso necessário para executar seus trabalhos, como um token de licença de terceiros, largura de banda de acesso ao banco de dados, a necessidade de chamadas de controle de utilização para uma API de terceiros e assim por diante. Você especifica os recursos consumíveis necessários para a execução de um trabalho e o Batch leva essas dependências de recursos em consideração ao agendar um trabalho. Você pode reduzir a subutilização dos recursos computacionais alocando somente os trabalhos que têm todos os recursos necessários disponíveis. Para obter mais informações, consulte [Agendamento com reconhecimento de recurso](#).

Ambiente de serviço

Um ambiente de serviço define como o AWS Batch se integra ao SageMaker para execução de trabalhos. Os ambientes de serviço permitem ao AWS Batch enviar e gerenciar trabalhos no SageMaker, ao mesmo tempo em que fornecem os recursos de enfileiramento, agendamento e gerenciamento de prioridades do AWS Batch. Os ambientes de serviço definem limites de capacidade para tipos de serviços específicos, como trabalhos do SageMaker Training. Os limites de capacidade controlam o máximo de recursos que podem ser usados por trabalhos de serviço no ambiente. Para obter mais informações, consulte [Ambientes de serviço para AWS Batch](#).

Trabalhos de serviço

Um trabalho de serviço é uma unidade de trabalho que você envia ao AWS Batch para execução em um ambiente de serviço. Os trabalhos de serviço utilizam os recursos de enfileiramento e agendamento AWS Batch enquanto delegam a execução real ao serviço externo. Por exemplo, trabalhos do SageMaker Training enviados como trabalhos de serviço são colocados em fila e priorizados pelo AWS Batch, mas a execução do trabalho do SageMaker Training ocorre dentro da infraestrutura do SageMaker AI. Essa integração permite que cientistas de dados e engenheiros de ML se beneficiem do gerenciamento automatizado de workload do AWS Batch e do enfileiramento prioritário para seus workloads do SageMaker AI Training. Os trabalhos de serviço podem referenciar outros trabalhos por nome ou ID e dar suporte às dependências do trabalho. Para obter mais informações, consulte [Empregos de serviços em AWS Batch](#).

Configurar o AWS Batch

Se você já tiver se cadastrado no Amazon Web Services (AWS) e usado o Amazon Elastic Compute Cloud (Amazon EC2) ou Amazon Elastic Container Service (Amazon ECS), poderá usá-lo em breve. AWS Batch O processo de configuração para os dois serviços é semelhante. Isso ocorre porque AWS Batch usa instâncias de contêiner do Amazon ECS em seus ambientes computacionais. Para usar a AWS CLI com o AWS Batch, você deve usar uma versão da AWS CLI que ofereça suporte aos recursos mais recentes do AWS Batch. Se você sentir falta do suporte a um determinado recurso do AWS Batch na AWS CLI, atualize para a versão mais recente. Para obter mais informações, consulte <http://aws.amazon.com/cli/>.

Note

Como o AWS Batch usa componentes do Amazon EC2, você usa o console do Amazon EC2 para muitas dessas etapas.

Conclua as tarefas a seguir para configuração do AWS Batch.

Tópicos

- [Criar conta di IAM e usuário administrativo](#)
- [Criar funções do IAM para seus ambientes de computação e instâncias de contêiner](#)
- [Criar um par de chaves para suas instâncias](#)
- [Crie uma VPC](#)
- [Criar um grupo de segurança](#)
- [Instalar o AWS CLI](#)

Criar conta di IAM e usuário administrativo

Para começar, você precisa criar uma AWS conta e um único usuário que normalmente receba direitos administrativos. Para isso, conclua os tutoriais a seguir:

Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra a <https://portal.aws.amazon.com/billing/inscrição>.
2. Siga as instruções online.

Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando <https://aws.amazon.com/e> escolhendo Minha conta.

Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS Centro de Identidade do AWS IAM, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

Proteja seu Usuário raiz da conta da AWS

1. Faça login [Console de gerenciamento da AWS](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Fazer login como usuário-raiz](#) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

Criar um usuário com acesso administrativo

1. Habilita o Centro de Identidade do IAM.

Para obter instruções, consulte [Habilitar o Centro de Identidade do AWS IAM](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. No Centro de Identidade do IAM, conceda o acesso administrativo a um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Criar um conjunto de permissões](#) no Guia do usuário do Centro de Identidade do AWS IAM .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

Para obter instruções, consulte [Adicionar grupos](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Criar funções do IAM para seus ambientes de computação e instâncias de contêiner

Os ambientes de computação e instâncias de contêiner do AWS Batch exigem credenciais de conta da Conta da AWS para fazer chamadas para outras APIs da AWS em seu nome. Crie um perfil do AWS Identity and Access Management que fornece essas credenciais para seus ambientes de computação e instâncias de contêiner e, em seguida, associar esse perfil aos seus ambientes de computação.

Note

Para verificar se o Conta da AWS tem as permissões necessárias, consulte [Configuração inicial do serviço IAM para sua conta](#).

Os perfis de instância de contêiner e ambiente de computação do AWS Batch são criados para você automaticamente na primeira execução do console. Portanto, se você pretende usar o console do AWS Batch, pode avançar para a próxima seção. Se você planeja usar a AWS CLI, conclua os procedimentos em [Usar perfis vinculados a serviço do AWS Batch](#), [Perfil de instância do Amazon ECS](#) e [Tutorial: criar o perfil de execução do IAM](#) antes de criar seu primeiro ambiente computacional.

Criar um par de chaves para suas instâncias

A AWS usa criptografia de chave pública para proteger as informações de login da instância. Uma instância do Linux, como uma instância de contêiner do AWS Batch não possui senha a ser usada para acesso SSH. Você usa um par de chaves para fazer login na sua instância com segurança. Você especifica o nome do par de chaves ao criar o ambiente de computação e fornece a chave privada ao fazer login usando SSH.

Se ainda não tiver criado um par de chaves, você poderá criar um usando o console do Amazon EC2. Observe que, se quiser iniciar instâncias em várias Regiões da AWS, crie um par de chaves em cada região. Para obter mais informações sobre regiões, consulte [Regiões e zonas de disponibilidade](#) no Manual do usuário do Amazon EC2.

Para criar um par de chaves

1. Abra o console do Amazon EC2 em <https://console.aws.amazon.com/ec2/>.
2. Na barra de navegação, selecione uma Região da AWS para o par de chaves. Você pode selecionar qualquer região que esteja disponível, independentemente de sua localização: No entanto, pares de chaves são específicos de uma região. Por exemplo, se você planeja executar uma instância na região US West (Oregon), deve criar um par de chaves para a instância na mesma região.
3. No painel de navegação, escolha Pares de chaves, Criar par de chaves.
4. Na caixa de diálogo Criar par de chaves, para Nome do par de chaves, insira um nome para o novo par de chaves e escolha Criar. Escolha um nome que seja fácil de memorizar, como

seu nome de usuário, seguido por `-key-pair` mais o nome da região. Por exemplo, `eu-par de chaves-uswest2`.

5. O arquivo de chave privada é baixado automaticamente pelo navegador. O nome do arquivo base é o especificado por você como sendo o nome do par de chaves, e a extensão do nome do arquivo é `.pem`. Salve o arquivo de chave privada em um lugar seguro.

 Important

Esta é a única chance de você salvar o arquivo de chave privada. Será necessário fornecer o nome do par de chaves ao executar uma instância e a chave privada correspondente sempre que for estabelecida uma conexão com a instância.

6. Se você usar um cliente de SSH em um computador Mac ou Linux para se conectar à instância do Linux, use o seguinte comando para definir as permissões do arquivo de chave privada. Dessa forma, só você pode lê-lo.

```
$ chmod 400 your_user_name-key-pair-region_name.pem
```

Para obter mais informações, consulte [Pares de chaves do Amazon EC2](#) no Guia do usuário do Amazon EC2.

Para se conectar à instância usando o par de chaves

Para se conectar à instância do Linux em um computador no qual o Mac ou o Linux esteja em execução, especifique o arquivo `.pem` ao cliente SSH com a opção `-i` e o caminho para a chave privada. Para se conectar à instância do Linux em um computador no qual o Windows esteja em execução, você pode usar MindTerm ou PuTTY. Se você planeja usar PuTTY, precisa instalá-lo e usar o procedimento a seguir para converter o arquivo `.pem` em um arquivo `.ppk`.

(Opcional) Para se preparar para se conectar a uma instância do Linux a partir do Windows usando PuTTY

1. Baixe e instale o PuTTY em <http://www.chiark.greenend.org.uk/~sgtatham/putty/>. Instale o pacote inteiro.
2. Inicie o PuTTYgen (por exemplo, no menu Start, escolha All Programs, PuTTY, and PuTTYgen).
3. Em Tipo de chave a ser gerada, escolha RSA. Se você estiver usando uma versão mais antiga do PuTTYgen, escolha SSH-2 RSA.

Parameters

Type of key to generate:

☒ RSA ☐ DSA ☐ ECDSA ☐ Ed25519 ☐ SSH-1 (RSA)

Number of bits in a generated key:

- Escolha Load. Por padrão, o PuTTYgen exibe somente arquivos com a extensão .ppk. Para localizar o arquivo .pem, escolha a opção para exibir arquivos de todos os tipos.

File name:

PuTTY Private Key Files (*.ppk) ▼

PuTTY Private Key Files (*.ppk)

All Files (*.*)

- Selecione o arquivo de chave privada que você criou no procedimento anterior e escolha Abrir. Escolha OK para descartar a caixa de diálogo de confirmação.
- Escolha Save private key. A PuTTYgen exibe um aviso sobre salvar a chave sem uma senha. Escolha Sim.
- Especifique o mesmo nome da chave usado para o par de chaves. O PuTTY adiciona automaticamente a extensão de arquivo .ppk.

Crie uma VPC

Com a Amazon Virtual Private Cloud (Amazon VPC) você pode iniciar recursos da AWS em uma rede virtual definida por você. É altamente recomendável ativar as instâncias de contêiner em um VPC.

Caso tenha uma VPC padrão, você também pode ignorar esta seção e avançar à próxima tarefa, [Criar um grupo de segurança](#). Para determinar se você tem uma VPC padrão, consulte [Plataformas compatíveis no console do Amazon EC2](#) no Manual do usuário do Amazon EC2

Para obter mais informações sobre como criar VPCs, consulte [Criação de uma VPC](#) no Guia do usuário da Amazon VPC. Consulte a tabela a seguir para determinar quais opções selecionar.

Opção	Valor	
Recursos a criar	Somente VPC	
Name	Forneça um nome opcional para a sua VPC.	

Opção	Valor	
Bloco IPv4 CIDR	Entrada manual IPv4 CIDR	
	O tamanho do bloco CIDR deve ter um tamanho entre /16 e /28.	
Bloco IPv6 CIDR	Sem bloco IPv6 CIDR	
Localção	Padrão	

Para saber mais sobre a Amazon VPC, consulte [O Que é Amazon VPC?](#) no Guia de usuário Amazon VPC.

Criar um grupo de segurança

Os security groups funcionam como um firewall para instâncias de contêiner de ambiente de computação associadas, controlando o tráfego de entrada e de saída no nível de instância de contêiner. Um grupo de segurança só pode ser usado na VPC na qual ele é criado.

É possível adicionar regras a um grupo de segurança que permite a você se conectar à instância de contêiner do endereço IP usando SSH. Você também pode adicionar regras que permitam o acesso HTTP e HTTPS de entrada e saída de qualquer lugar. Adicione eventuais regras a portas abertas exigidas pelas tarefas.

Caso pretenda ativar instâncias de contêiner em várias regiões, você precisa criar um grupo de segurança em cada região. Para obter mais informações, consulte [Regiões e Zonas de Disponibilidade](#) no Guia do usuário do Amazon EC2.

Note

O endereço IP público do computador local é necessário e pode ser obtido por meio de um serviço. Por exemplo, fornecemos o seguinte serviço: <http://checkip.amazonaws.com/> ou <https://checkip.amazonaws.com/>. Para localizar outro serviço que forneça o endereço IP, use a frase de busca "qual é o meu endereço IP". Caso esteja se conectando por meio de um provedor de serviços de Internet (ISP) ou atrás de um firewall sem um endereço IP estático, descubra o intervalo de endereços IP usado por computadores cliente.

Para criar um grupo de segurança usando o console

1. Abra o console da Amazon VPC em <https://console.aws.amazon.com/vpc/>.
2. No painel de navegação, selecione Grupos de segurança.
3. Escolha Create security group (Criar grupo de segurança).
4. Insira um nome e uma descrição para o grupo de segurança. Não é possível alterar o nome e a descrição de um grupo de segurança após ele ser criado.
5. Em VPC, escolha a VPC.
6. Por padrão, novos grupos de segurança começam com apenas uma regra de saída que permite que todo o tráfego deixe as instâncias. Adicione regras para permitir qualquer tráfego de entrada ou para restringir o tráfego de saída.

As instâncias de contêiner do AWS Batch não exigem que nenhuma porta de entrada esteja aberta. No entanto, talvez você queira adicionar uma regra SSH. No entanto, você pode adicionar uma regra SSH para fazer login na instância de contêiner e examinar os contêineres em trabalhos com comandos do Docker. Se desejar que sua instância de contêiner hospede um trabalho que executa um servidor da Web, você também pode adicionar regras para HTTP. Conclua as seguintes etapas para adicionar essas regras de security group opcionais.

Na guia Entrada, crie as seguintes regras e escolha Criar:

- Escolha Add Rule. Para Tipo, escolha HTTP. Para Origem, escolha Qualquer lugar (0.0.0.0/0).
- Escolha Add Rule. Para Tipo, escolha SSH. Para Source, escolha Custom IP e especifique o endereço IP público do seu computador ou rede na notação Roteamento sem classe entre domínios (CIDR). Se sua empresa alocar endereços de um intervalo, especifique o intervalo inteiro, como 203.0.113.0/24. Para especificar um único endereço IP em notação CIDR, escolha Meu IP. Isso adiciona o prefixo de roteamento /32 ao endereço IP público.

 Note

Por motivos de segurança, não recomendamos que você permita acesso SSH de todos os endereços IP (0.0.0.0/0) à sua instância, exceto para fins de teste e somente por um curto período.

7. É possível adicionar etiquetas agora ou pode adicioná-las mais tarde. Para adicionar uma tag, escolha Add new tag, e insira a chave e o valor da tag.

8. Escolha Create grupo de segurança.

Para criar um grupo de segurança usando a linha de comando, consulte [>create-security-group](#) (AWS CLI)

Para obter mais informações sobre os grupos de segurança, consulte [Como trabalhar com grupos de segurança](#).

Instalar o AWS CLI

Para usar a AWS CLI com o AWS Batch, 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 [Interface da linha de comando da AWS](#) no Guia do usuário do AWS Command Line Interface.

Introdução aos AWS Batch tutoriais

Você pode usar o assistente de AWS Batch primeira execução para começar AWS Batch rapidamente. Depois de concluir os pré-requisitos, você pode usar o assistente de primeira execução para criar um ambiente de computação, uma definição de trabalhos e uma fila de trabalhos.

Você também pode enviar um exemplo de job “Hello World” usando o assistente AWS Batch de primeira execução para testar sua configuração. Se você já tem uma imagem do Docker na qual deseja iniciar AWS Batch, pode usar essa imagem para criar uma definição de tarefa.

Depois, você pode usar o assistente de AWS Batch primeira execução para criar um ambiente computacional, uma fila de trabalhos e enviar um exemplo de trabalho do Hello World.

Conceitos básicos da orquestração do Amazon EC2 usando o assistente

O Amazon Elastic Compute Cloud (Amazon EC2) oferece capacidade de computação escalável na Nuvem AWS. O uso do Amazon EC2 elimina a necessidade de investir em hardware inicialmente, portanto, você pode desenvolver e implantar aplicativos com mais rapidez.

É possível usar o Amazon EC2 para executar quantos servidores virtuais forem necessários, configurar a segurança e as redes e gerenciar o armazenamento. O Amazon EC2 permite aumentar ou reduzir a escala verticalmente para lidar com alterações nos requisitos ou com picos em popularidade, reduzindo sua necessidade de prever o tráfego.

Visão geral

Este tutorial demonstra como configurar o AWS Batch com o assistente para configurar o Amazon EC2 e executar o Hello World.

Público-alvo

Este tutorial foi desenvolvido para administradores de sistemas e desenvolvedores responsáveis pela configuração, teste e implantação do AWS Batch.

Atributos usados

Este tutorial mostra como usar o assistente do console AWS Batch para:

- Criar e configurar um ambiente computacional do Amazon EC2
- Crie uma fila de trabalhos.
- Crie uma definição de trabalho
- Criar e enviar um trabalho para execução
- Visualizar a saída do trabalho no CloudWatch

Tempo necessário

Levará entre 10 e 15 minutos, aproximadamente, para concluir este tutorial.

Restrições regionais

Não há restrições de país ou regionais associadas ao uso desta solução.

Custos pelo uso de recursos

Não há cobrança para criar uma conta da AWS. Contudo, ao implementar essa solução, pode haver alguns ou todos os custos listados na tabela a seguir.

Descrição	Custo (dólares dos EUA)
Instância do Amazon EC2	Você paga por cada instância do Amazon EC2 que é criada. Para obter mais informações sobre a definição de preço, consulte Definição de preço do Amazon EC2 .

Pré-requisitos

Antes de começar:

- Crie uma Conta da AWS se ainda não tiver um.
- Crie o [Perfil de instância `ecsInstanceRole`](#).

Etapa 1: criar um ambiente computacional

Important

Para começar da forma mais simples e rápida possível, este tutorial inclui etapas com configurações padrão. Antes de criar para uso em produção, recomendamos que você se

familiarize com todas as configurações e implante com as configurações que atendam aos seus requisitos.

Para criar um ambiente de computação para uma orquestração do Amazon EC2, faça o seguinte:

1. Abra o [assistente de primeira execução do console do AWS Batch](#).
2. Para Configurar trabalho e tipo de orquestração, escolha Amazon Elastic Compute Cloud (Amazon EC2).
3. Escolha Próximo.
4. Na seção Configuração do ambiente de computação, em Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
5. Em Perfil de instância, escolha um perfil de instância existente que tenha as permissões de IAM necessárias anexadas. Esse perfil de instância permite que as instâncias de contêiner do Amazon ECS no seu ambiente computacional façam chamadas para as operações de APIs AWS necessárias. Para obter mais informações, consulte [Perfil de instância do Amazon ECS](#).

O nome padrão do Perfil de instância é `ecsInstanceRole`.

6. Em Configuração da instância, você pode deixar as configurações padrão.
7. Em Configuração de rede, use sua VPC padrão para o Região da AWS.
8. Escolha Próximo.

Etapa 2: criar uma fila de trabalhos

Uma fila de trabalhos armazena os trabalhos enviados até que o Scheduler do AWS Batch execute o trabalho em um recurso de computação no seu ambiente de computação. Para obter mais informações, consulte [Filas de tarefas](#)

Para criar uma fila de trabalhos para uma orquestração do Amazon EC2, faça o seguinte:

1. Em Configuração da fila de trabalhos, em Nome, especifique um nome exclusivo para sua fila de trabalhos. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
2. É possível deixar o valor padrão para todas as outras opções de configuração.

3. Escolha Próximo.

Etapa 3: Criar uma definição de trabalho

As definições de tarefas do AWS Batch especificam como os trabalhos devem ser executados. Embora cada tarefa deva fazer referência a uma definição de tarefa, muitos parâmetros que são especificados na definição de tarefa podem ser substituídos em runtime.

Para criar a definição de trabalho:

1. Em Criar uma definição de trabalho
 - a. em Nome, especifique um nome exclusivo para a sua fila de trabalhos. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
 - b. Em Comando - opcional, você pode mudar `hello world` para uma mensagem personalizada ou deixá-la como está.
2. É possível deixar o valor padrão para todas as outras opções de configuração.
3. Escolha Próximo.

Etapa 4: Criar um trabalho

Para criar um trabalho, faça o seguinte:

1. Na seção Configuração do trabalho, em Nome, especifique um nome exclusivo para o trabalho. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
2. É possível deixar o valor padrão para todas as outras opções de configuração.
3. Escolha Next (Próximo).

Etapa 5: analisar e criar

Na página Revisar e criar, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar recursos.

1. Em Revisar e criar, escolha Criar recursos.

2. Uma janela se abre quando o AWS Batch começa a alocar seus recursos. Depois de concluído, escolha Ir para o painel. No painel, você deve ver todos os recursos alocados e se o trabalho está no estado `Runnable`. Seu trabalho está agendado para ser executado e deve ser concluído em 2 a 3 minutos.

Etapa 6: ver a saída do trabalho

Para ver o resultado do trabalho, faça o seguinte:

1. No painel de navegação, escolha Trabalhos.
2. No menu suspenso Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
3. A tabela de Trabalhos lista todos os seus trabalhos e qual é o status atual deles. Depois que o Status do trabalho for Bem-sucedido, escolha o Nome do trabalho para ver os detalhes do trabalho.
4. No painel Detalhes, escolha Nome do fluxo de log. O console do CloudWatch para o trabalho será aberto e deverá haver um evento com a Mensagem do `hello world` ou sua mensagem personalizada.

Etapa 7: limpar os recursos do tutorial

Você será cobrado pela instância do Amazon EC2 enquanto ela estiver ativada. É possível excluir a instância para não incorrer mais em cobranças.

Para excluir os recursos criados, faça o seguinte:

1. No painel de navegação, escolha Filas de trabalhos.
2. Na tabela Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
3. Escolha Disable. Depois que o Estado da fila de trabalhos estiver desativado, você poderá escolher Excluir.
4. Depois que a fila de trabalhos for excluída, no painel de navegação, escolha Ambientes computacionais.
5. Escolha o ambiente computacional que você criou para este tutorial e, em seguida, escolha Desativar. Pode levar de 1 a 2 minutos para que o ambiente computacional seja desativado.
6. Quando o Estado do ambiente computacional estiver desativado, escolha Excluir. Pode levar de 1 a 2 minutos para que o ambiente computacional seja excluído.

Recursos adicionais

Depois de concluir o tutorial, você pode querer explorar os seguintes tópicos:

- Explore os componentes principais do AWS Batch. Para obter mais informações, consulte [Componentes do AWS Batch](#).
- Saiba mais sobre os diferentes [Ambientes computacionais](#) disponíveis no AWS Batch.
- Saiba mais sobre as [Filas de trabalhos](#) e suas diferentes opções de agendamento.
- Saiba mais sobre as [Definições de trabalho](#) e suas diferentes opções de configuração.
- Saiba mais sobre os diferentes tipos de [Trabalho](#).

Introdução AWS Batch e orquestração do Fargate usando o Wizard

AWS O Fargate inicia e dimensiona a computação de acordo com os requisitos de recursos que você especifica para o contêiner. Com o Fargate, você não precisa provisionar em excesso nem pagar por servidores adicionais. Para obter mais informações, consulte [Fargate](#).

Visão geral do

Este tutorial demonstra como fazer a configuração AWS Batch com o Assistente para configurar o AWS Fargate e executá-lo. Hello World

Público-alvo

Este tutorial foi desenvolvido para administradores de sistemas e desenvolvedores responsáveis pela configuração, teste e implantação do AWS Batch.

Características usadas

Este tutorial mostra como usar o assistente do AWS Batch console para:

- Crie e configure um ambiente AWS computacional Fargate
- Crie uma fila de trabalhos.
- Crie uma definição de trabalho
- Criar e enviar um trabalho para execução
- Visualize a saída do trabalho em CloudWatch

Tempo necessário

A conclusão deste tutorial deve levar cerca de 10 a 15 minutos.

Restrições regionais

Não há restrições de país ou regionais associadas ao uso desta solução.

Custos de uso de recursos

Não há cobrança pela criação de uma AWS conta. Contudo, ao implementar essa solução, pode haver alguns ou todos os custos listados na tabela a seguir.

Description	Custo (dólares dos EUA)
O preço é baseado na vCPU, memória, sistemas operacionais, arquitetura da CPU e recursos de armazenamento solicitados para a tarefa ou o pod.	Para obter mais informações sobre a definição de preço, consulte Definição de preços do Fargate .

Pré-requisitos

Antes de começar

- Crie uma Conta da AWS se você não tiver uma.
- Crie o perfil de execução da tarefa. Se ainda não criou o [Perfil de execução de tarefa](#), você pode criá-lo como parte deste tutorial.

Etapa 1: criar um ambiente computacional

Important

Para começar da forma mais simples e rápida possível, este tutorial inclui etapas com configurações padrão. Antes de criar para uso em produção, recomendamos que você se familiarize com todas as configurações e implante com as configurações que atendam aos seus requisitos.

Para criar um ambiente de computação para uma orquestração do Fargate, faça o seguinte:

1. Abra o [assistente de primeira execução do console do AWS Batch](#).

2. Em Configurar tipo de trabalho e orquestração, escolha Fargate.
3. Escolha Próximo.
4. Na seção Configuração do ambiente de computação, em Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
5. É possível deixar o valor padrão para todas as outras opções de configuração.
6. Escolha Próximo.

Etapa 2: criar uma fila de trabalhos

Uma fila de trabalhos armazena seus trabalhos enviados até que o AWS Batch Agendador execute o trabalho em um recurso em seu ambiente computacional. Para criar uma fila de trabalhos:

Para criar uma fila de trabalhos para uma orquestração do Fargate, faça o seguinte:

1. Na seção Configuração da fila de trabalhos, em Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
2. Em Prioridade, insira 900 para a fila de trabalhos.
3. É possível deixar o valor padrão para todas as outras opções de configuração.
4. Escolha Próximo.

Etapa 3: criar uma definição de tarefa

Para criar a definição de trabalho:

1. Na seção Configuração geral:
 - Na seção Configuração geral, em Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
2. Na seção Configuração da plataforma Fargate:
 - a. Ative Atribuir IP público para atribuir um endereço IP público. Você precisa de um IP público para baixar a imagem do contêiner, a menos que tenha configurado um repositório de imagens privado.

- b. Para Função de execução, escolha uma função de execução de tarefas que permita que agentes do Amazon Elastic Container Service (Amazon ECS) AWS façam chamadas em seu nome. Escolha `ecsTaskExecutionFunção` ou `BatchEcsTaskExecutionRole`.

Para criar um Perfil de execução, escolha Criar um perfil de execução. No modal Criar perfil do IAM, selecione Criar perfil do IAM.

- i. O console do IAM tem a configuração de permissão já configurada para criar o perfil de execução.
- ii. Em Tipo de entidade confiável, verifique se o serviço AWS está selecionado.
- iii. Em Serviço ou caso de usuário, verifique se o Elastic Container Service está selecionado.
- iv. Escolha Próximo.
- v. Para políticas de permissões, verifique se a Amazon ECSTask ExecutionRolePolicy está selecionada.
- vi. Escolha Próximo.
- vii. Em Nome, revise e crie, verifique se o nome da função é `BatchEcsTaskExecutionRole`.
- viii. Selecione Criar perfil.
- ix. No AWS Batch console, escolha o botão Atualizar ao lado de Função de execução. Escolha a função `BatchEcsTaskExecutionRole` de execução.

3. Na seção Configuração de Contêiner:

- Em Comando, você pode mudar `hello world` para uma mensagem personalizada ou deixá-la como está.

4. É possível deixar o valor padrão para todas as outras opções de configuração.

5. Escolha Próximo.

Etapa 4: Criar um trabalho

Para criar uma trabalho do Fargate, faça o seguinte:

1. Na seção Configuração do trabalho, em Nome, especifique um nome exclusivo para o trabalho. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
2. É possível deixar o valor padrão para todas as outras opções de configuração.

3. Escolha Próximo.

Etapa 5: analisar e criar

Na página Revisar e criar, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar recursos.

Etapa 6: ver a saída do trabalho

Para ver o resultado do trabalho, faça o seguinte:

1. No painel de navegação, escolha Trabalhos.
2. No menu suspenso Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
3. A tabela de Trabalhos lista todos os seus trabalhos e qual é o status atual deles. Depois que o Status do trabalho for Bem-sucedido, escolha o Nome do trabalho para ver os detalhes do trabalho.
4. No painel Detalhes, escolha Nome do fluxo de log. O CloudWatch console do Job será aberto e deverá haver um evento com a Mensagem de `hello world` ou sua mensagem personalizada.

Etapa 7: limpar os recursos do tutorial

Você será cobrado pela EC2 instância da Amazon enquanto ela estiver ativada. É possível excluir a instância para não incorrer mais em cobranças.

Para excluir os recursos criados, faça o seguinte:

1. No painel de navegação, escolha Filas de trabalhos.
2. Na tabela Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
3. Escolha Disable. Depois que o Estado da fila de trabalhos estiver desativado, você poderá escolher Excluir.
4. Depois que a fila de trabalhos for excluída, no painel de navegação, escolha Ambientes computacionais.
5. Escolha o ambiente computacional que você criou para este tutorial e, em seguida, escolha Desativar. Pode levar de 1 a 2 minutos para que o ambiente computacional seja desativado.
6. Quando o Estado do ambiente computacional estiver desativado, escolha Excluir. Pode levar de 1 a 2 minutos para que o ambiente computacional seja excluído.

Recursos adicionais do

Depois de concluir o tutorial, você pode querer explorar os seguintes tópicos:

- Saiba mais sobre as [Práticas recomendadas](#).
- Explore os componentes AWS Batch principais. Para obter mais informações, consulte [Componentes do AWS Batch](#).
- Saiba mais sobre os diferentes [Ambientes computacionais](#) disponíveis no AWS Batch.
- Saiba mais sobre as [Filas de trabalhos](#) e suas diferentes opções de agendamento.
- Saiba mais sobre as [Definições de trabalho](#) e suas diferentes opções de configuração.
- Saiba mais sobre os diferentes tipos de [Trabalho](#).

Introdução ao Fargate AWS Batch e ao Fargate usando o AWS CLI

Este tutorial demonstra como configurar AWS Batch com AWS Fargate orquestração e executar uma tarefa simples “Hello World” usando o AWS Command Line Interface ().AWS CLI Você aprenderá como criar ambientes computacionais, filas de trabalho, definições de trabalhos e enviar trabalhos para o AWS Batch.

Tópicos

- [Pré-requisitos](#)
- [Criar um perfil de execução do IAM](#)
- [Crie um ambiente de computação](#)
- [Crie uma fila de trabalhos](#)
- [Crie uma definição de trabalho](#)
- [Enviar e monitorar um trabalho](#)
- [Ver a saída de trabalho](#)
- [Limpar os recursos](#)
- [Transição para o ambiente de produção](#)
- [Próximas etapas](#)

Pré-requisitos

Antes de iniciar este tutorial, certifique-se de ter o seguinte.

1. AWS CLI A. Caso ainda seja necessário instalá-la, consulte o [guia de instalação da AWS CLI](#). Você também pode [usar AWS CloudShell](#), o que inclui AWS CLI o.
2. Configurei seu AWS CLI com as credenciais apropriadas. Caso ainda não tenha realizado a configuração das credenciais, execute o comando `aws configure`.
3. Familiaridade básica com interfaces de linha de comando e conceitos de containerização.
4. [Como AWS Batch funciona com o IAM](#) para criar e gerenciar AWS Batch recursos, funções do IAM e recursos de VPC em seu. Conta da AWS
5. Um ID de sub-rede e um ID de grupo de segurança de uma VPC em seu. Conta da AWS Caso não tenha uma VPC, você pode [criar uma](#). Para obter mais informações sobre como usar o AWS CLI para recuperar esses recursos IDs, consulte [describe-subnets](#) e na Referência de Comandos. [describe-security-groups](#) AWS CLI

Tempo necessário: aproximadamente 15 a 20 minutos para concluir este tutorial.

Custo: este tutorial usa os recursos computacionais do Fargate. O custo estimado para concluir este tutorial é inferior a 0,01 USD, supondo que você siga as instruções de limpeza para excluir recursos imediatamente após a conclusão. O preço do Fargate é baseado nos recursos de vCPU e memória consumidos, cobrados por segundo com um mínimo de 1 minuto. Para obter informações atuais sobre a definição de preço, consulte [Preços do AWS Fargate](#).

Criar um perfil de execução do IAM

AWS Batch requer uma função de execução que permita que agentes do Amazon Elastic Container Service (Amazon ECS) AWS façam chamadas de API em seu nome. Essa função é necessária para que as tarefas do Fargate extraiam imagens de contêineres e gravem registros na Amazon. CloudWatch

Criar um documento de política de confiança

Primeiro, crie uma política de confiança para permitir que o serviço de tarefas do Amazon ECS assumo o perfil.

```
cat > batch-execution-role-trust-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
```

```
"Effect": "Allow",
"Principal": {
  "Service": "ecs-tasks.amazonaws.com"
},
"Action": "sts:AssumeRole"
}
]
}
EOF
```

Crie a função de execução

O comando a seguir cria um perfil do IAM chamado `BatchEcsTaskExecutionRoleTutorial` usando a política de confiança que você acabou de criar.

```
aws iam create-role \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --assume-role-policy-document file://batch-execution-role-trust-policy.json
```

Anexe a política necessária

Anexe a política AWS gerenciada que fornece as permissões necessárias para a execução de tarefas do Amazon ECS.

```
aws iam attach-role-policy \
  --role-name BatchEcsTaskExecutionRoleTutorial \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

A função agora está pronta para ser usada na execução de AWS Batch tarefas do Fargate.

Crie um ambiente de computação

Um ambiente computacional define os recursos computacionais nos quais seus trabalhos em lotes serão executados. Neste tutorial, você criará um ambiente computacional Fargate gerenciado que provisiona e dimensiona automaticamente os recursos com base nos requisitos do trabalho.

Criar o ambiente de computação

O comando a seguir cria um ambiente computacional Fargate. Substitua o exemplo de sub-rede e grupo de segurança pelo seu próprio, de acordo IDs com o. [Pré-requisitos](#)

```
aws batch create-compute-environment \  
  --compute-environment-name my-fargate-compute-env \  
  --type MANAGED \  
  --state ENABLED \  
  --compute-resources type=FARGATE,maxvCpus=128,subnets=subnet-  
a123456b,securityGroupIds=sg-a12b3456
```

O exemplo a seguir mostra a aparência da saída quando o comando é executado com sucesso.

```
{  
  "computeEnvironmentName": "my-fargate-compute-env",  
  "computeEnvironmentArn": "arn:aws:batch:us-west-2:123456789012:compute-environment/  
my-fargate-compute-env"  
}
```

Aguarde até que o ambiente computacional esteja pronto

Verifique o status do seu ambiente computacional para garantir que ele esteja pronto antes de continuar.

```
aws batch describe-compute-environments \  
  --compute-environments my-fargate-compute-env \  
  --query 'computeEnvironments[0].status'
```

```
"VALID"
```

Quando o status é exibido como VALID, seu ambiente computacional está pronto para aceitar trabalhos.

Crie uma fila de trabalhos

Uma fila de trabalhos armazena os trabalhos enviados até que o AWS Batch agendador os execute em recursos em seu ambiente computacional. Os trabalhos são processados em ordem de prioridade na fila.

Criar a fila de trabalhos

O comando a seguir cria uma fila de trabalhos com prioridade 900 que usa seu ambiente computacional Fargate.

```
aws batch create-job-queue \  
  --job-queue-name my-fargate-job-queue \  
  --state ENABLED \  
  --priority 900 \  
  --compute-environment-order order=1,computeEnvironment=my-fargate-compute-env
```

O exemplo a seguir mostra a aparência da saída quando o comando é executado com sucesso.

```
{  
  "jobQueueName": "my-fargate-job-queue",  
  "jobQueueArn": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue"  
}
```

Verificar se a fila de trabalhos está pronta

Verifique se sua fila de trabalhos está no estado ENABLED e pronta para aceitar trabalhos.

```
aws batch describe-job-queues \  
  --job-queues my-fargate-job-queue \  
  --query 'jobQueues[0].state' "ENABLED"
```

Crie uma definição de trabalho

As definições de trabalhos especificam como os trabalhos devem ser executados, incluindo a imagem do Docker a ser usada, requisitos de recursos e outros parâmetros. Para o Fargate, você usará os requisitos de recursos em vez dos parâmetros tradicionais de vCPU e memória.

Criar a definição de trabalho

O comando a seguir cria uma definição de trabalho que executa um comando simples “hello world” usando a imagem do contêiner busybox. 123456789012Substitua pelo seu Conta da AWS ID real e substitua o exemplo Região da AWS pelo seu.

```
aws batch register-job-definition \  
  --job-definition-name my-fargate-job-def \  
  --type container \  
  --platform-capabilities FARGATE \  
  --container-properties '{  
    "image": "busybox",
```

```
    "resourceRequirements": [
      {"type": "VCPU", "value": "0.25"},
      {"type": "MEMORY", "value": "512"}
    ],
    "command": ["echo", "hello world"],
    "networkConfiguration": {
      "assignPublicIp": "ENABLED"
    },
    "executionRoleArn": "arn:aws:iam::123456789012:role/
BatchEcsTaskExecutionRoleTutorial"
  },
{
  "jobDefinitionName": "my-fargate-job-def",
  "jobDefinitionArn": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1",
  "revision": 1
}'
```

A definição do trabalho especifica 0.25 vCPU e 512 MB de memória, que são os recursos mínimos para uma tarefa do Fargate. A configuração `assignPublicIp` está ativada para que o contêiner possa extrair a imagem do busybox do Docker Hub.

Enviar e monitorar um trabalho

Agora que você tem todos os componentes necessários, você pode enviar um trabalho para a fila e monitorar seu progresso.

Envie um trabalho

O comando a seguir envia um trabalho para sua fila usando a definição de trabalho que você criou.

```
aws batch submit-job \
  --job-name my-hello-world-job \
  --job-queue my-fargate-job-queue \
  --job-definition my-fargate-job-def
```

O exemplo a seguir mostra a aparência da saída quando o comando é executado com sucesso.

```
{
  "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
  "jobName": "my-hello-world-job",
```

```
"jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a"
}
```

Anote o jobId que retornou na resposta, pois você o usará para monitorar o progresso do trabalho.

Monitorar o status do trabalho

Use o ID do trabalho para verificar o status do trabalho. O trabalho progredirá em vários estados: SUBMITTED, PENDING, RUNNABLE, STARTING, RUNNING e, finalmente, SUCCEEDED ou FAILED.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

O exemplo a seguir mostra a aparência da saída quando o comando é executado com sucesso.

```
{
  "jobs": [
    {
      "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
      "jobName": "my-hello-world-job",
      "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a",
      "jobQueue": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-queue",
      "status": "SUCCEEDED",
      "createdAt": 1705161908000,
      "jobDefinition": "arn:aws:batch:us-west-2:123456789012:job-definition/my-fargate-job-def:1"
    }
  ]
}
```

Quando o status for exibido como SUCCEEDED, seu trabalho foi concluído com sucesso.

Ver a saída de trabalho

Depois que seu trabalho for concluído, você poderá visualizar sua saída no Amazon CloudWatch Logs.

Obter o nome do fluxo de logs

Primeiro, recupere o nome do fluxo de log a partir dos detalhes do trabalho. Substitua o ID de trabalho de exemplo por seu próprio.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'jobs[0].attempts[0].containers[0].logStreamName' \
  --output text
```

```
my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Ver os logs de trabalho

Use o nome do fluxo de registros para recuperar a saída do trabalho em CloudWatch Logs.

```
aws logs get-log-events \
  --log-group-name /aws/batch/job \
  --log-stream-name my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a \
  --query 'events[*].message' \
  --output text
```

A saída mostra “hello world”, confirmando que seu trabalho foi executado com sucesso.

Limpar os recursos

Para evitar cobranças contínuas, limpe os recursos que você criou neste tutorial. Você deve excluir os recursos na ordem correta devido às dependências.

Desativar e excluir a fila de trabalhos

Primeiro, desative a fila de trabalhos e, em seguida, exclua-a.

```
aws batch update-job-queue \
  --job-queue my-fargate-job-queue \
  --state DISABLED
```

```
aws batch delete-job-queue \
  --job-queue my-fargate-job-queue
```

Desativar e excluir o ambiente computacional

Depois que a fila de trabalhos for excluída, desative e exclua o ambiente computacional.

```
aws batch update-compute-environment \
```

```
--compute-environment my-fargate-compute-env \  
--state DISABLED
```

```
aws batch delete-compute-environment \  
--compute-environment my-fargate-compute-env
```

Limpar o perfil do IAM

Remova o anexo da política e exclua a perfil do IAM.

```
aws iam detach-role-policy \  
--role-name BatchEcsTaskExecutionRoleTutorial \  
--policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
aws iam delete-role \  
--role-name BatchEcsTaskExecutionRoleTutorial
```

Remover arquivos temporários

Exclua o arquivo de política de confiança que você criou.

```
rm batch-execution-role-trust-policy.json
```

Todos os recursos foram limpos com sucesso.

Transição para o ambiente de produção

Este tutorial foi desenvolvido para ajudar você a entender como AWS Batch funciona com o Fargate. Em implantações de produção, considere os seguintes requisitos adicionais:

Considerações sobre segurança:

- Criar grupos de segurança dedicados com o mínimo de acesso necessário em vez de usar grupos de segurança padrão
- Usar sub-redes privadas com o NAT Gateway em vez da atribuição de IP público para contêineres
- Armazenar imagens de contêiner no Amazon ECR em vez de usar repositórios públicos
- Implemente endpoints VPC para comunicação de AWS serviços a fim de evitar tráfego na Internet

Considerações sobre a arquitetura:

- Implementar em várias zonas de disponibilidade para alta disponibilidade
- Implementar estratégias de repetição de trabalhos e filas de mensagens não entregues para lidar com erros
- Usar várias filas de trabalhos com prioridades diferentes para o gerenciamento do workload
- Configurar políticas de ajuste de escala automático com base na profundidade da fila e na utilização de recursos
- Implementar monitoramento e alertas sobre falhas no trabalho e utilização de recursos

Considerações operacionais:

- Configure CloudWatch painéis e alarmes para monitoramento
- Implementar trilhas de auditoria e registro em log adequadas
- Use CloudFormation ou AWS CDK para infraestrutura como código
- Estabelecer procedimentos de backup e recuperação de desastres

Para obter uma orientação abrangente sobre arquiteturas prontas para produção, consulte a [Estrutura bem-arquitetada AWS](#) e as [Práticas recomendadas de segurança AWS](#).

Próximas etapas

Agora que você concluiu este tutorial, você pode explorar AWS Batch recursos mais avançados:

- [Filas de tarefas](#) — Saiba mais sobre agendamento de filas de trabalhos e gerenciamento de prioridades
- [Definições de trabalho](#) — Explore configurações avançadas de definição de trabalhos, incluindo variáveis de ambiente, volumes e estratégias de repetição
- [Ambientes computacionais para AWS Batch](#) — Entenda os diferentes tipos de ambientes computacionais e opções de escalabilidade
- [Trabalhos em paralelo de vários nós](#) — Execute trabalhos que abrangem vários nós de computação
- [Trabalhos de matriz](#) — Envie um grande número de trabalhos similares de forma eficiente
- [Práticas recomendadas do AWS Batch](#) — Aprenda técnicas de otimização para workloads de produção

Conceitos básicos do AWS Batch no Amazon EKS

O AWS Batch no Amazon EKS é um serviço gerenciado para programar e escalar workloads em lotes em clusters do Amazon EKS existentes. O AWS Batch não cria, administra nem executa operações de ciclo de vida de seus clusters do Amazon EKS em seu nome. A orquestração do AWS Batch aumenta e diminui os nós gerenciados pelo AWS Batch e executa pods nesses nós.

O AWS Batch não toca em nós, grupos de nós com ajuste de escala automático ou ciclos de vida de pods que não estejam associados a ambientes de computação do AWS Batch em seu cluster do Amazon EKS. Para o AWS Batch operar de forma eficaz, sua [perfil vinculada ao serviço](#) precisa de permissões de controle de acesso com base em perfil (RBAC) do Kubernetes em seu cluster do Amazon EKS existente. Para obter mais informações, consulte [Usar autorização RBAC](#) na documentação do Kubernetes.

O AWS Batch requer um namespace do Kubernetes em que possa incluir pods como trabalhos do AWS Batch. Recomendamos um namespace dedicado para isolar os pods do AWS Batch das outras workloads do cluster.

Depois do AWS Batch receber acesso RBAC e de um namespace ser estabelecido, você pode associar esse cluster do Amazon EKS a um ambiente de computação do AWS Batch usando a operação de API [CreateComputeEnvironment](#). Uma fila de trabalhos pode ser associada a esse novo ambiente de computação do Amazon EKS. Os trabalhos do AWS Batch são enviados para a fila de trabalhos com base em uma definição de trabalho do Amazon EKS usando a operação da API [SubmitJob](#). O AWS Batch então inicia os nós gerenciados do AWS Batch e coloca os trabalhos da fila de trabalhos como pods do Kubernetes no cluster do EKS associado a um ambiente de computação do AWS Batch.

As seções a seguir abordam como se preparar para usar o AWS Batch no Amazon EKS.

Sumário

- [Visão geral](#)
- [Pré-requisitos](#)
- [Etapa 1: criar o cluster do Amazon EKS para AWS Batch](#)
- [Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch](#)
- [Etapa 3: criar um ambiente computacional do Amazon EKS](#)
- [Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação](#)

- [Etapa 5: Criar uma definição de trabalho](#)
- [Etapa 6: Enviar um trabalho](#)
- [Etapa 7: ver a saída do trabalho](#)
- [Etapa 8: \(Opcional\) Enviar um trabalho com substituições](#)
- [Etapa 9: limpar os recursos do tutorial](#)
- [Recursos adicionais](#)

Visão geral

Este tutorial demonstra como configurar AWS Batch com um Amazon EKS usando o AWS CLI, `kubectl` e `eksctl`.

Público-alvo

Este tutorial foi desenvolvido para administradores de sistemas e desenvolvedores responsáveis pela configuração, teste e implantação do AWS Batch.

Atributos usados

Este tutorial mostra como usar o AWS CLI para:

- Criar e configurar um ambiente computacional do Amazon EKS
- Crie uma fila de trabalhos.
- Crie uma definição de trabalho
- Criar e enviar um trabalho para execução
- Enviar um trabalho com substituições

Tempo necessário

Levará entre 30 e 40 minutos, aproximadamente, para concluir este tutorial.

Restrições regionais

Não há restrições de país ou regionais associadas ao uso desta solução.

Custos pelo uso de recursos

Não há cobrança para criar uma conta da AWS. Contudo, ao implementar essa solução, pode haver alguns ou todos os custos listados na tabela a seguir.

Descrição	Custo (dólares dos EUA)
Você é cobrado por hora do cluster	Varia de acordo com a instância, consulte os preços do Amazon EKS

Pré-requisitos

Antes de iniciar este tutorial, você deve instalar e configurar as ferramentas e os recursos a seguir necessários para criar e gerenciar os recursos do AWS Batch e do Amazon EKS.

- **AWS CLI** – Uma ferramenta de linha de comando para trabalhar com os serviços AWS, incluindo o Amazon EKS. Este guia exige que você use a versão 2.8.6 ou superior, ou a versão 1.26.0 ou superior. Para obter mais informações, consulte [Como instalar, atualizar e desinstalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface. Depois de instalar a AWS CLI, recomendamos que você também a configure. Para obter mais informações, consulte [Configuração rápida com o aws configure](#) no Manual do usuário do AWS Command Line Interface.
- **kubectl**: uma ferramenta de linha de comando para trabalhar com clusters do Kubernetes. Este guia requer que você use a versão 1.23 ou superior. Para obter mais informações, consulte [Instalar ou atualizar o kubectl](#) no Guia do usuário do Amazon EKS.
- **eksctl** – uma ferramenta de linha de comando para trabalhar com clusters do Amazon EKS que automatiza várias tarefas individuais. Este guia requer que você use a versão 0.115.0 ou superior. Para obter mais informações, consulte [Instalar ou atualizar o eksctl](#) no Guia do usuário do Amazon EKS.
- **Permissões obrigatórias do IAM**: a entidade principal de segurança do IAM que você está usando deve ter permissões para trabalhar com perfis do IAM do Amazon EKS e perfis vinculados a serviço, o CloudFormation e uma VPC com recursos relacionados. Para obter mais informações, consulte [Ações, recursos e chaves de condição do Amazon Elastic Kubernetes Service](#) e [Usar perfis vinculados a serviço](#), no Guia do usuário do IAM. Você deve concluir todas as etapas deste manual como o mesmo usuário.
- **Permissões**: os usuários que chamam a operação [CreateComputeEnvironment](#) da API para criar um ambiente de computação que usa recursos do Amazon EKS precisam de permissões para a operação `eks:DescribeCluster` da API.
- **Número Conta da AWS** — Você precisa saber seu ID Conta da AWS. Siga as instruções em [Visualizar seu ID Conta da AWS](#).

- (Opcional) CloudWatch — Para examinar os detalhes de [\(Opcional\) Enviar um trabalho com substituições](#), o registro em log deve ser configurado. Para obter mais informações, consulte [Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS](#).

Etapa 1: criar o cluster do Amazon EKS para AWS Batch

Important

Para começar da forma mais simples e rápida possível, este tutorial inclui etapas com configurações padrão. Antes de criar para uso em produção, recomendamos que você se familiarize com todas as configurações e implante com as configurações que atendam aos seus requisitos.

Depois de instalar os pré-requisitos, você precisará criar seu cluster usando o `eksctl`. Criar o cluster pode levar de 10 a 15 minutos.

```
$ eksctl create cluster --name my-cluster-name --region region-code
```

No comando anterior, substitua:

- Substitua *my-cluster-name* pelo nome que você deseja usar para seu cluster.
- Substitua *region-code* por Região da AWS para criar o cluster, por exemplo, em `us-west-2`.

O nome e a região do cluster são necessários para um momento posterior deste tutorial.

Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch

Todas as etapas são necessárias.

1. Crie um namespace dedicado para trabalhos do AWS Batch

Use o `kubectl` para criar um namespace.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
```

```
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Resultado:

```
namespace/my-aws-batch-namespace created
```

2. Habilite o acesso por meio do controle de acesso com base em perfil (RBAC)

Use o `kubectl` para criar uma função do Kubernetes para o cluster do AWS Batch para permitir observar nós e pods e vincular a função. Você deve fazer isso uma vez para cada cluster do EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
- apiGroups: [""]
  resources: ["namespaces"]
  verbs: ["get"]
- apiGroups: [""]
  resources: ["nodes"]
  verbs: ["get", "list", "watch"]
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["get", "list", "watch"]
- apiGroups: [""]
  resources: ["events"]
  verbs: ["list"]
- apiGroups: [""]
  resources: ["configmaps"]
```

```

    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Resultado:

```

clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created

```

3. Crie uma função do com escopo de namespace do Kubernetes para o AWS Batch para gerenciar e fazer o ciclo de vida dos pods e vincule-os. Você deve fazer isso uma vez para cada namespace exclusivo.

```
$ namespace=my-aws-batch-namespace
```

```

$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
- apiGroups: [""]
  resources: ["pods"]

```

```

    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Resultado:

```

role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created

```

4. Atualize o mapa de configuração do Kubernetes aws-auth para mapear as permissões RBAC anteriores para a função vinculada ao serviço do AWS Batch.

No comando a seguir, substitua:

- Substitua **<your-account-number>** pelo seu número da Conta da AWS.

```

$ eksctl create iamidentitymapping \
  --cluster my-cluster-name \
  --arn "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" \
  --username aws-batch

```


Resultado:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" to auth ConfigMap
```

Note

O caminho `aws-service-role/batch.amazonaws.com/` foi removido do ARN so perfil vinculado ao serviço. Isso ocorre devido a um problema com o mapa de configuração `aws-auth`. Para obter mais informações, consulte [Perfis com caminhos não funcionam quando o caminho é incluído em seu ARN no aws-authconfigmap](#).

Etapa 3: criar um ambiente computacional do Amazon EKS

Ambientes de computação do AWS Batch definem parâmetros de recursos de computação para atender às suas necessidades de workload em lote. Em um ambiente de computação gerenciado, o AWS Batch ajuda a gerenciar os tipos de capacidade e instância dos recursos de computação (nós do Kubernetes) dentro do seu cluster do Amazon EKS. Isso se baseia na especificação do recurso de computação que você define ao criar o ambiente de computação. Você pode usar instâncias sob demanda do EC2 ou instâncias spot do EC2.

Agora que a função `AWSServiceRoleForBatch` vinculada ao serviço tem acesso ao seu cluster do Amazon EKS, você pode criar recursos do AWS Batch. Primeiro, crie um ambiente computacional que aponte para seu cluster do Amazon EKS.

- Para o subnets executar `eksctl get cluster my-cluster-name` para obter as sub-redes usadas pelo cluster.
- Para o parâmetro `securityGroupIds` você pode usar o mesmo grupo de segurança do cluster do Amazon EKS. Esse comando recupera o ID do grupo de segurança do cluster.

```
$ aws eks describe-cluster \
  --name my-cluster-name \
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- O `instanceRole` é criado quando você cria o cluster. Para encontrar a lista `instanceRole` de todas as entidades que usam a política `AmazonEKSEWorkerNodePolicy`:

```
$ aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/AmazonEKSWorkerNodePolicy
```

O nome do perfil da política contém o nome do cluster que você criou `eksctl-my-cluster-name`-nodegroup-example.

Para encontrar o `instanceRole` e executar o comando a seguir:

```
$ aws iam list-instance-profiles-for-role --role-name eksctl-my-cluster-name-nodegroup-example
```

Resultado:

```
INSTANCEPROFILES      arn:aws:iam::<your-account-number>:instance-profile/eks-04cb2200-94b9-c297-8dbe-87f12example
```

Para mais informações, consulte [Criar o perfil do IAM do nó do Amazon EKS](#) e [Habilitar o acesso da entidade principal do IAM ao seu cluster](#) no Guia do usuário do Amazon EKS. Se você estiver usando redes de pods, consulte [Como configurar o plugin CNI da Amazon VPC para o Kubernetes para usar perfis do IAM para contas de serviço](#) no Guia do usuário do Amazon EKS.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:region-code:your-account-number:cluster/my-cluster-name",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
      "m5"
    ],
  },
}
```

```

    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
}
EOF

```

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Observações

- A manutenção de um ambiente de computação do Amazon EKS é uma responsabilidade compartilhada. Para obter mais informações, consulte [Responsabilidade compartilhada dos nós Kubernetes](#).

Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação

Important

É importante confirmar se o ambiente de computação está íntegro antes de continuar. A operação da API [DescribeComputeEnvironments](#) pode ser usada para fazer isso.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Confirme se o parâmetro status não está INVALID. Se estiver, veja o parâmetro statusReason para saber a causa. Para obter mais informações, consulte [Solução de problemas AWS Batch](#).

Os trabalhos enviados para essa nova fila de trabalhos são executados como pods em nós gerenciados do AWS Batch que se juntaram ao cluster do Amazon EKS associado ao seu ambiente computacional.

```
$ cat <<EOF > ./batch-eks-job-queue.json
```

```
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Etapa 5: Criar uma definição de trabalho

A definição de trabalho a seguir instrui o pod a dormir por 60 segundos.

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    },
    "metadata": {
      "labels": {
```

```
        "environment": "test"
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Observações

- Há considerações sobre os parâmetros `cpu` e `memory`. Para obter mais informações, consulte [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#).

Etapa 6: Enviar um trabalho

Execute o comando AWS CLI a seguir para enviar um novo Trabalho.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
  --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
```

Para verificar o status de um trabalho:

```
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Observações

- Para obter mais informações sobre como executar trabalhos em recursos do Amazon EKS, consulte [Trabalhos do Amazon EKS](#).

Etapa 7: ver a saída do trabalho

Para ver o resultado do trabalho, faça o seguinte:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Trabalhos.

3. No menu suspenso Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
4. A tabela de Trabalhos lista todos os seus trabalhos e qual é o status atual deles. Depois que o Status do trabalho for Bem-sucedido, escolha o Nome do trabalho, *My-Eks-JQ1*, para visualizar os detalhes do trabalho.
5. No painel Detalhes, os horários Iniciado em e Concluído em devem ter um minuto de intervalo.

Etapa 8: (Opcional) Enviar um trabalho com substituições

Esse trabalho substitui o comando passado para o contêiner. O AWS Batch limpa agressivamente os pods após a conclusão dos trabalhos para reduzir a carga para o Kubernetes. Para examinar os detalhes de um trabalho, o log deve ser configurado. Para obter mais informações, consulte [Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS](#).

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job --cli-input-json file:///./submit-job-override.json
```

Observações

- Para melhorar a visibilidade dos detalhes das operações, ative o log do ambiente de gerenciamento Amazon EKS. Para obter mais informações, consulte [Logs do ambiente de gerenciamento do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- A sobrecarga de Daemonsets e kubelets afeta os recursos de vCPU e memória disponíveis, especificamente a escalabilidade e o posicionamento do trabalho. Para obter mais informações, consulte [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#).

Para ver o resultado do trabalho, faça o seguinte:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Trabalhos.
3. No menu suspenso Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
4. A tabela de Trabalhos lista todos os seus trabalhos e qual é o status atual deles. Depois que o Status do trabalho for Bem-sucedido, escolha o Nome do trabalho para ver os detalhes do trabalho.
5. No painel Detalhes, escolha Nome do fluxo de log. O console do CloudWatch para o trabalho será aberto e deverá haver um evento com a Mensagem do hello world ou sua mensagem personalizada.

Etapa 9: limpar os recursos do tutorial

Você será cobrado pela instância do Amazon EC2 enquanto ela estiver ativada. É possível excluir a instância para não incorrer mais em cobranças.

Para excluir os recursos criados, faça o seguinte:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Filas de trabalhos.
3. Na tabela Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
4. Escolha Disable. Depois que o Estado da fila de trabalhos estiver desativado, você poderá escolher Excluir.
5. Depois que a fila de trabalhos for excluída, no painel de navegação, escolha Ambientes computacionais.

6. Escolha o ambiente computacional que você criou para este tutorial e, em seguida, escolha Desativar. Pode levar de 1 a 2 minutos para que o ambiente computacional seja desativado.
7. Quando o Estado do ambiente computacional estiver desativado, escolha Excluir. Pode levar de 1 a 2 minutos para que o ambiente computacional seja excluído.

Recursos adicionais

Depois de concluir o tutorial, você pode querer explorar os seguintes tópicos:

- Saiba mais sobre as [Práticas recomendadas](#).
- Explore os componentes principais do AWS Batch. Para obter mais informações, consulte [Componentes do AWS Batch](#).
- Saiba mais sobre os diferentes [Ambientes computacionais](#) disponíveis no AWS Batch.
- Saiba mais sobre as [Filas de trabalhos](#) e suas diferentes opções de agendamento.
- Saiba mais sobre as [Definições de trabalho](#) e suas diferentes opções de configuração.
- Saiba mais sobre os diferentes tipos de [Trabalho](#).

Conceitos básicos do AWS Batch em clusters privados do Amazon EKS

O AWS Batch é um serviço gerenciado que orquestra workloads em lote nos seus clusters do Amazon Elastic Kubernetes Service (Amazon EKS). Isso inclui enfileiramento, monitoramento de dependências, tentativas e prioridades gerenciadas de tarefas, gerenciamento de pods e nós em escala. Esse recurso conecta seu cluster privado existente do Amazon EKS com o AWS Batch para executar seus trabalhos em grande escala. É possível usar o [eksctl](#) (uma interface de linha de comando para o Amazon EKS), o console do AWS ou a [AWS Command Line Interface](#) para criar um cluster privado do Amazon EKS com todos os outros recursos necessários.

Por padrão, os [clusters privados do Amazon EKS](#) não têm acesso de entrada/saída à Internet, e você só pode acessar o servidor de API pela VPC ou por uma rede conectada. Endpoints da VPC Amazon são usados para permitir o acesso privado a outros serviços da AWS. O `eksctl` oferece suporte à criação de clusters totalmente privados usando uma Amazon VPC e sub-redes pré-existent. O `eksctl` também cria endpoints da Amazon VPC na Amazon VPC fornecida e modifica as tabelas de rotas para as sub-redes fornecidas.

Cada sub-rede deve ter uma tabela de rotas explícita associada a ela, pois o `eksctl` não modifica a tabela de rotas principal. O [cluster](#) deve extrair imagens de um registro de contêiner que esteja na Amazon VPC. Você também pode criar um Amazon Elastic Container Registry na Amazon VPC e copiar para ele as imagens de contêiner para que os nós as extraia dali. Para obter mais informações, consulte [Copiar uma imagem de contêiner de um repositório para outro](#). Para começar a usar repositórios privados do Amazon ECR, consulte [Repositórios privados do Amazon ECR](#).

Opcionalmente, é possível criar uma [regra de cache de pullthrough](#) com o Amazon ECR. Depois que uma regra de cache de pullthrough é criada para um registro público externo, é possível extrair uma imagem desse registro público externo usando seu URI (Identificador de recursos uniformes) de registro privado do Amazon ECR. Em seguida, o Amazon ECR cria um repositório e armazena a imagem no cache. Quando uma imagem em cache é extraída usando o URI de registro privado do Amazon ECR, o Amazon ECR verifica o registro remoto para ver se há uma nova versão da imagem e atualiza seu registro privado até uma vez a cada 24 horas.

Sumário

- [Visão geral](#)
- [Pré-requisitos](#)
- [Etapa 1: criar seu cluster EKS para AWS Batch](#)
- [Etapa 2: preparando seu cluster EKS para o AWS Batch](#)
- [Etapa 3: criar um ambiente computacional do Amazon EKS](#)
- [Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação](#)
- [Etapa 5: criar um Amazon ECR com cache pull through](#)
- [Etapa 6: registrar uma definição de trabalho](#)
- [Etapa 7: enviar um trabalho para execução](#)
- [Etapa 8: ver a saída do trabalho](#)
- [Etapa 9: \(Opcional\) Enviar um trabalho com substituições](#)
- [Etapa 10: limpar os recursos do tutorial](#)
- [Recursos adicionais](#)
- [Solução de problemas](#)

Visão geral

Este tutorial demonstra como configurar AWS Batch com um Amazon EKS privado usando o AWS CloudShell, `kubectl` e `eksctl`.

Público-alvo

Este tutorial foi desenvolvido para administradores de sistemas e desenvolvedores responsáveis pela configuração, teste e implantação do AWS Batch.

Atributos usados

Este tutorial mostra como usar o AWS CLI para:

- Usar o Amazon Elastic Container Registry (Amazon ECR) para armazenar imagens de contêiner
- Criar e configurar um ambiente computacional do Amazon EKS
- Crie uma fila de trabalhos.
- Crie uma definição de trabalho
- Criar e enviar um trabalho para execução
- Enviar um trabalho com substituições

Tempo necessário

Levará entre 40 e 50 minutos, aproximadamente, para concluir este tutorial.

Restrições regionais

Não há restrições de país ou regionais associadas ao uso desta solução.

Custos pelo uso de recursos

Não há cobrança para criar uma conta da AWS. Contudo, ao implementar essa solução, pode haver alguns ou todos os custos listados na tabela a seguir.

Descrição	Custo (dólares dos EUA)
Você é cobrado por hora do cluster	Varia de acordo com a instância, consulte os preços do Amazon EKS
Instância do Amazon EC2	Você paga por cada instância do Amazon EC2 que é criada. Para obter mais informaçõ

Descrição	Custo (dólares dos EUA)
	es sobre a definição de preço, consulte Definição de preço do Amazon EC2 .

Pré-requisitos

Este tutorial usa o AWS CloudShell, que é um shell pré-autenticado baseado em navegador que você pode iniciar diretamente do Console de gerenciamento da AWS. Isso permite o acesso ao cluster quando ele não tem mais acesso público à Internet. O AWS CLI, `kubectl` e `eksctl` podem já estar instalados como parte do AWS CloudShell. Para obter mais informações sobre o AWS CloudShell, consulte o [AWS CloudShell Guia do usuário](#). Uma alternativa ao AWS CloudShell é conectar-se à VPC do seu cluster ou a uma [rede conectada](#).

Para executar comandos `kubectl`, você precisará de acesso privado ao cluster do Amazon EKS. Isso significa que todo o tráfego para o servidor de API do cluster deve vir da VPC do cluster ou de uma rede conectada.

- **AWS CLI** – Uma ferramenta de linha de comando para trabalhar com os serviços AWS, incluindo o Amazon EKS. Este guia exige que você use a versão 2.8.6 ou superior, ou a versão 1.26.0 ou superior. Para obter mais informações, consulte [Como instalar, atualizar e desinstalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface. Depois de instalar a AWS CLI, recomendamos que você também a configure. Para obter mais informações, consulte [Configuração rápida com o `aws configure`](#) no Manual do usuário do AWS Command Line Interface.
- **`kubectl`**: uma ferramenta de linha de comando para trabalhar com clusters do Kubernetes. Este guia requer que você use a versão 1.23 ou superior. Para obter mais informações, consulte [Instalar ou atualizar o `kubectl`](#) no Guia do usuário do Amazon EKS.
- **`eksctl`** – uma ferramenta de linha de comando para trabalhar com clusters do Amazon EKS que automatiza várias tarefas individuais. Este guia requer que você use a versão 0.115.0 ou superior. Para obter mais informações, consulte [Instalar ou atualizar o `eksctl`](#) no Guia do usuário do Amazon EKS.
- **Permissões**: os usuários que chamam a operação [CreateComputeEnvironment](#) da API para criar um ambiente computacional que usa recursos do Amazon EKS precisam de permissões para a operação `eks:DescribeCluster` e `eks:ListClusters` da API. Você pode anexar a política

gerenciada [AWSBatchfullAccess](#) à sua conta de usuário seguindo as instruções [Como adicionar e remover permissões de identidade IAM](#) no Guia do usuário do IAM.

- InstanceRole — Você precisa criar um InstanceRole para seus nós do Amazon EKS que tenha as políticas AmazonEKSWorkerNodePolicy e AmazonEC2ContainerRegistryPullOnly. Para obter instruções sobre como criar o InstanceRole, consulte [Criar o perfil do IAM do nó Amazon EKS](#). Você precisará do ARN do InstanceRole.
- ID Conta da AWS — Você precisa saber seu ID Conta da AWS. Siga as instruções em [Visualizar seu ID Conta da AWS](#).
- (Opcional) CloudWatch — Para examinar os detalhes de [\(Opcional\) Enviar um trabalho com substituições](#), o registro em log deve ser configurado. Para obter mais informações, consulte [Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS](#).

Etapa 1: criar seu cluster EKS para AWS Batch

Important

Para começar da forma mais simples e rápida possível, este tutorial inclui etapas com configurações padrão. Antes de criar para uso em produção, recomendamos que você se familiarize com todas as configurações e implante com as configurações que atendam aos seus requisitos.

Recomendamos que você use `eksctl` e o seguinte arquivo de configuração para criar seu cluster. Para configurar manualmente seu cluster, siga as instruções em [Implantar clusters privados com acesso limitado à Internet](#) no Guia do usuário do Amazon EKS.

1. Abra o [Console AWS CloudShell](#) e defina a região como `us-east-1`. Para o resto do tutorial, certifique-se de usar o `us-east-1`.
2. Crie um cluster EKS privado na região `us-east-1` usando o arquivo de configuração de amostra `eksctl`. Salve o arquivo `yaml` em seu ambiente AWS CloudShell e dê a ele o nome `clusterConfig.yaml`. Você pode alterar `my-test-cluster` com o nome que deseja usar para seu cluster.

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
metadata:
  name: my-test-cluster
```

```

    region: us-east-1
availabilityZones:
  - us-east-1a
  - us-east-1b
  - us-east-1c
managedNodeGroups:
  - name: ng-1
    privateNetworking: true
privateCluster:
  enabled: true
  skipEndpointCreation: false

```

3. Crie seus recursos com o comando: `eksctl create cluster -f clusterConfig.yaml`. A criação do cluster pode levar de 10 a 15 minutos.
4. Depois que o cluster terminar de ser criado, você deverá adicionar seu endereço IP AWS CloudShell à lista de permissões. Para encontrar seu endereço IP AWS CloudShell, execute o seguinte comando:

```
curl http://checkip.amazonaws.com
```

Quando tiver o endereço IP público, você precisará criar uma regra de lista de permissões:

```

aws eks update-cluster-config \
  --name my-test-cluster \
  --region us-east-1 \
  --resources-vpc-config
endpointPublicAccess=true,endpointPrivateAccess=true,publicAccessCidrs=["<Public
IP>/32"]

```

Em seguida, aplique a atualização no arquivo de configuração kubectl:

```
aws eks update-kubeconfig --name my-test-cluster --region us-east-1
```

5. Para testar se você tem acesso aos nós, execute o seguinte comando:

```
kubectl get nodes
```

A saída do comandos é:

NAME	STATUS	ROLES	AGE	VERSION

ip-192-168-107-235.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-165-40.ec2.internal	Ready	none	1h	v1.32.3-eks-473151a
ip-192-168-98-54.ec2.internal	Ready	none	1h	v1.32.1-eks-5d632ec

Etapa 2: preparando seu cluster EKS para o AWS Batch

Todas as etapas são obrigatórias e devem ser executadas no AWS CloudShell.

1. Crie um namespace dedicado para trabalhos do AWS Batch

Use o `kubectl` para criar um namespace.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
}
EOF
```

Resultado:

```
namespace/my-aws-batch-namespace created
```

2. Habilite o acesso por meio do controle de acesso com base em perfil (RBAC)

Use o `kubectl` para criar uma função do Kubernetes para o cluster do AWS Batch para permitir observar nós e pods e vincular a função. Você deve fazer isso uma vez para cada cluster do Amazon EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
```

```

metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: ["" ]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: ["" ]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["" ]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["" ]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: ["" ]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
  - kind: User
    name: aws-batch
    apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF

```

Resultado:

```

clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created

```

Crie uma função do com escopo de namespace do Kubernetes para o AWS Batch para gerenciar e fazer o ciclo de vida dos pods e vincule-os. Você deve fazer isso uma vez para cada namespace exclusivo.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
- apiGroups: [""]
  resources: ["pods"]
  verbs: ["create", "get", "list", "watch", "delete", "patch"]
- apiGroups: [""]
  resources: ["serviceaccounts"]
  verbs: ["get", "list"]
- apiGroups: ["rbac.authorization.k8s.io"]
  resources: ["roles", "rolebindings"]
  verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Resultado:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
```



```
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
created
```

Atualize o mapa de configuração do Kubernetes `aws-auth` para mapear as permissões RBAC anteriores para a função vinculada ao serviço do AWS Batch.

```
$ eksctl create iamidentitymapping \
  --cluster my-test-cluster \
  --arn "arn:aws:iam::<your-account-ID>:role/AWSServiceRoleForBatch" \
  --username aws-batch
```

Resultado:

```
2022-10-25 20:19:57 [#] adding identity "arn:aws:iam::<your-account-ID>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

Note

O caminho `aws-service-role/batch.amazonaws.com/` foi removido do ARN so perfil vinculado ao serviço. Isso ocorre devido a um problema com o mapa de configuração `aws-auth`. Para obter mais informações, consulte [Perfis com caminhos não funcionam quando o caminho é incluído em seu ARN no aws-authconfigmap](#).

Etapa 3: criar um ambiente computacional do Amazon EKS

Ambientes de computação do AWS Batch definem parâmetros de recursos de computação para atender às suas necessidades de workload em lote. Em um ambiente de computação gerenciado, o AWS Batch ajuda a gerenciar os tipos de capacidade e instância dos recursos de computação (nós do Kubernetes) dentro do seu cluster do Amazon EKS. Isso se baseia na especificação do recurso de computação que você define ao criar o ambiente de computação. Você pode usar instâncias sob demanda do EC2 ou instâncias spot do EC2.

Agora que a função `AWSServiceRoleForBatch` vinculada ao serviço tem acesso ao seu cluster do Amazon EKS, você pode criar recursos do AWS Batch. Primeiro, crie um ambiente computacional que aponte para seu cluster do Amazon EKS.

- Para o subnets executar `eksctl get cluster my-test-cluster` para obter as sub-redes usadas pelo cluster.
- Para o parâmetro `securityGroupIds` você pode usar o mesmo grupo de segurança do cluster do Amazon EKS. Esse comando recupera o ID do grupo de segurança do cluster.

```
$ aws eks describe-cluster \  
  --name my-test-cluster \  
  --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Use o ARN do `instanceRole` que você criou nos Pré-requisitos.

```
$ cat <<EOF > ./batch-eks-compute-environment.json  
{  
  "computeEnvironmentName": "My-Eks-CE1",  
  "type": "MANAGED",  
  "state": "ENABLED",  
  "eksConfiguration": {  
    "eksClusterArn": "arn:aws:eks:us-east-1:<your-account-ID>:cluster/my-test-cluster",  
    "kubernetesNamespace": "my-aws-batch-namespace"  
  },  
  "computeResources": {  
    "type": "EC2",  
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",  
    "minvCpus": 0,  
    "maxvCpus": 128,  
    "instanceTypes": [  
      "m5"  
    ],  
    "subnets": [  
      "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"  
    ],  
    "securityGroupIds": [  
      "<eks-cluster-sg>"  
    ],  
    "instanceRole": "<eks-instance-profile>"  
  }  
}  
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-compute-environment.json
```

Observações

- A manutenção de um ambiente de computação do Amazon EKS é uma responsabilidade compartilhada. Para obter mais informações, consulte [Segurança no Amazon EKS](#).

Etapa 4: Criar uma fila de trabalhos e anexar o ambiente de computação

Important

É importante confirmar se o ambiente de computação está íntegro antes de continuar. A operação da API [DescribeComputeEnvironments](#) pode ser usada para fazer isso.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Confirme se o parâmetro status não está INVALID. Se estiver, veja o parâmetro statusReason para saber a causa. Para obter mais informações, consulte [Solução de problemas AWS Batch](#).

Os trabalhos enviados para essa nova fila de trabalhos são executados como pods em nós gerenciados do AWS Batch que se juntaram ao cluster do Amazon EKS associado ao seu ambiente computacional.

```
$ cat <<EOF > ./batch-eks-job-queue.json
{
  "jobQueueName": "My-Eks-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-CE1"
    }
  ]
}
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-job-queue.json
```

Etapa 5: criar um Amazon ECR com cache pull through

Como o cluster não tem acesso público à Internet, você precisa criar um Amazon ECR para imagens de contêineres. As instruções a seguir criam um Amazon ECR com uma regra de cache pull-through para armazenar a imagem.

1. O comando a seguir cria a regra de cache pull-through. Você pode substituir *tutorial-prefix* por um prefixo diferente.

```
aws ecr create-pull-through-cache-rule \  
  --ecr-repository-prefix "my-prefix" \  
  --upstream-registry-url "public.ecr.aws" \  
  --region us-east-1
```

2. Autentique com o ECR público.

```
aws ecr get-login-password --region us-east-1 | docker login --username AWS --  
password-stdin <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com
```

Agora você pode puxar uma imagem.

```
docker pull <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/  
amazonlinux/amazonlinux:2
```

3. Você pode verificar o repositório e a imagem executando os seguintes comandos:

```
aws ecr describe-repositories
```

```
aws ecr describe-images --repository-name my-prefix/amazonlinux/amazonlinux
```

4. A string de imagem a ser usada para puxar o contêiner está no seguinte formato:

```
<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/amazonlinux/  
amazonlinux:2
```

Etapa 6: registrar uma definição de trabalho

A definição de trabalho a seguir instrui o pod a dormir por 60 segundos.

No campo de imagem da definição do trabalho, em vez de fornecer um link para a imagem em um repositório do ECR público, forneça o link para a imagem armazenada no nosso repositório do ECR privado. Veja o exemplo de definição de trabalho a seguir:

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-definition.json
```

Observações

- Há considerações sobre os parâmetros `cpu` e `memory`. Para obter mais informações, consulte [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#).

Etapa 7: enviar um trabalho para execução

Execute o comando AWS CLI a seguir no AWS CloudShell para enviar um novo trabalho e retornar o JobID exclusivo.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
  --job-definition MyJobOnEks_Sleep - --job-name My-Eks-Job1
```

Observações

- Para obter mais informações sobre como executar trabalhos em recursos do Amazon EKS, consulte [Trabalhos do Amazon EKS](#).

Etapa 8: ver a saída do trabalho

Para verificar o status de um trabalho:

```
$ aws batch describe-jobs --job <JobID-from-submit-response>
```

`startedAt` e `stoppedAt` devem ter um minuto de diferença.

Etapa 9: (Opcional) Enviar um trabalho com substituições

Esse trabalho substitui o comando passado para o contêiner.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
```

```
"jobDefinition": "MyJobOnEks_Sleep",
"eksPropertiesOverride": {
  "podProperties": {
    "containers": [
      {
        "command": [
          "/bin/sh"
        ],
        "args": [
          "-c",
          "echo hello world"
        ]
      }
    ]
  }
}
EOF
```

```
$ aws batch submit-job - -cli-input-json file:///./submit-job-override.json
```

Observações

- Para melhorar a visibilidade dos detalhes das operações, ative o log do ambiente de gerenciamento Amazon EKS. Para obter mais informações, consulte [Logs do ambiente de gerenciamento do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- A sobrecarga de Daemonsets e kubelets afeta os recursos de vCPU e memória disponíveis, especificamente a escalabilidade e o posicionamento do trabalho. Para obter mais informações, consulte [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#).

Etapa 10: limpar os recursos do tutorial

Você será cobrado pela instância do Amazon EC2 enquanto ela estiver ativada. É possível excluir a instância para não incorrer mais em cobranças.

Para excluir os recursos criados, faça o seguinte:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Filas de trabalhos.

3. Na tabela Fila de trabalhos, escolha a fila de trabalhos que você criou para o tutorial.
4. Em Ações, escolha Desativar. Depois que o Estado da fila de trabalhos estiver desativado, você poderá escolher Excluir.
5. Depois que a fila de trabalhos for excluída, no painel de navegação, escolha Ambientes computacionais.
6. Escolha o ambiente computacional que você criou para este tutorial e, em seguida, escolha Desativar em Ações. Pode levar de 1 a 2 minutos para que o ambiente computacional seja desativado.
7. Quando o Estado do ambiente computacional estiver desativado, escolha Excluir. Pode levar de 1 a 2 minutos para que o ambiente computacional seja excluído.

Recursos adicionais

Depois de concluir o tutorial, você pode querer explorar os seguintes tópicos:

- Saiba mais sobre as [Práticas recomendadas](#).
- Explore os componentes principais do AWS Batch. Para obter mais informações, consulte [Componentes do AWS Batch](#).
- Saiba mais sobre os diferentes [Ambientes computacionais](#) disponíveis no AWS Batch.
- Saiba mais sobre as [Filas de trabalhos](#) e suas diferentes opções de agendamento.
- Saiba mais sobre as [Definições de trabalho](#) e suas diferentes opções de configuração.
- Saiba mais sobre os diferentes tipos de [Trabalho](#).

Solução de problemas

Se os nós inicializados pelo AWS Batch não tiverem acesso ao repositório do Amazon ECR (ou a qualquer outro repositório) que armazena sua imagem, seus trabalhos poderão permanecer no estado STARTING. Isso ocorre porque o pod não conseguirá fazer download da imagem e executar seu trabalho do AWS Batch. Se você clicar no nome do pod iniciado pelo AWS Batch, poderá ver a mensagem de erro e confirmar o problema. A mensagem de erro deve ser semelhante à seguinte:

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =  
Unknown desc = failed to pull and unpack image  
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference  
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
```



```
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Para outros cenários comuns de solução de problemas, consulte [Solução de problemas com o AWS Batch](#). Para solução de problemas com base no status do pod, consulte [Como solucionar problemas do status do pod no Amazon EKS?](#).

Começando a usar AWS Batch a SageMaker IA

AWS Batch os trabalhos de serviço permitem que você envie trabalhos de SageMaker treinamento por meio de filas de AWS Batch trabalhos com recursos de agendamento, priorização e enfileiramento. Este tutorial demonstra como configurar e executar um trabalho de SageMaker treinamento simples usando trabalhos AWS Batch de serviço.

Sumário

- [Visão geral do](#)
- [Pré-requisitos](#)
- [Etapa 1: criar uma função de execução de SageMaker IA](#)
- [Etapa 2: criar seu ambiente de serviço](#)
- [Etapa 3: Crie sua fila SageMaker de trabalhos](#)
- [Etapa 4: criar e enviar um trabalho de treinamento](#)
- [Etapa 5: monitorar o status do trabalho](#)
- [Etapa 6: ver a saída do trabalho](#)
- [Etapa 7: limpar os recursos do tutorial](#)
- [Recursos adicionais do](#)

Visão geral do

Este tutorial demonstra como configurar trabalhos de AWS Batch serviço para trabalhos SageMaker de treinamento usando o. AWS CLI

Público-alvo

Este tutorial foi desenvolvido para cientistas de dados e desenvolvedores responsáveis por configurar e executar trabalhos de treinamento de machine learning em grande escala.

Características usadas

Este tutorial mostra como usar o AWS CLI para:

- Crie um ambiente de serviço para trabalhos SageMaker de treinamento
- Crie uma fila SageMaker de trabalhos de treinamento
- Enviar trabalhos de serviço usando a API `SubmitServiceJob`
- Monitorar o status do trabalho e visualizar os resultados
- CloudWatch Registros de acesso para trabalhos de treinamento

Tempo necessário

Ele levará aproximadamente 15 minutos para ser concluído.

Restrições regionais

Este tutorial pode ser concluído em qualquer AWS região em que tanto a IA AWS Batch quanto a SageMaker IA estejam disponíveis.

Custos de uso de recursos

Não há cobrança pela criação de uma AWS conta. No entanto, ao implementar essa solução, pode haver custos para os seguintes recursos:

Description	Custo (dólares dos EUA)
SageMaker Instâncias de treinamento de IA	Você paga por cada instância de treinamento de SageMaker IA usada. Para obter mais informações sobre preços, consulte Preços do SageMaker AI .
Armazenamento do Amazon S3	Custo mínimo para armazenar resultados do trabalho de treinamento. Para obter mais informações, consulte Preços do Amazon S3 .

Pré-requisitos

Antes de iniciar este tutorial, você deve instalar e configurar as seguintes ferramentas e recursos necessários para criar e gerenciar tanto os recursos de IA AWS Batch quanto os recursos de SageMaker IA.

- AWS CLI— Uma ferramenta de linha de comando para trabalhar com AWS serviços, incluindo AWS Batch SageMaker IA. Este guia exige que você use a versão 2.8.6 ou posterior. Para obter mais informações, consulte [Como instalar, atualizar e desinstalar a AWS CLI](#) no Guia do usuário da AWS Command Line Interface . Depois de instalar o AWS CLI, recomendamos que você também o configure. Para obter mais informações, consulte [Configuração rápida com o aws configure](#) no Manual do usuário do AWS Command Line Interface .

Etapa 1: criar uma função de execução de SageMaker IA

SageMaker A IA usa funções de execução para realizar operações em seu nome usando outros AWS serviços. Você deve criar uma função de execução e conceder permissões de SageMaker IA para usar os serviços e recursos necessários para trabalhos de treinamento. Use a política gerenciada AmazonSageMakerFullAccess, pois ela inclui permissões para o Amazon S3.

Note

Use as instruções a seguir para criar a função de execução de SageMaker IA para este tutorial.

Antes de criar uma função de execução para seu ambiente de produção, recomendamos que você analise [Como usar funções de execução de SageMaker IA](#) no [guia do desenvolvedor de SageMaker IA](#).

1. Crie o perfil do IAM.

Crie um arquivo JSON chamado `sagemaker-trust-policy.json` com a política de confiança a seguir:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "sagemaker.amazonaws.com"
      },
    },
  ],
}
```

```
        "Action": "sts:AssumeRole"
    }
  ]
}
```

Crie um perfil do IAM usando a política de confiança:

```
aws iam create-role \
  --role-name SageMakerExecutionRole \
  --assume-role-policy-document file:///sagemaker-trust-policy.json \
  --description "Execution role for SageMaker training jobs"
```

2. Anexar políticas gerenciadas

Anexe as políticas do IAM gerenciadas necessárias ao perfil:

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

```
aws iam attach-role-policy \
  --role-name SageMakerExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
```

3. Anotar o perfil de ARN

Obtenha o ARN do perfil, que você precisará em etapas posteriores:

```
aws iam get-role --role-name SageMakerExecutionRole --query 'Role.Arn' --output
text
```

Salve esse ARN da forma como ele será usado ao criar sua carga útil de trabalho de treinamento.

Etapa 2: criar seu ambiente de serviço

Um ambiente de serviço define as restrições de capacidade para trabalhos de SageMaker treinamento. O ambiente de serviço encapsula o número máximo de instâncias de treinamento que podem ser executadas simultaneamente.

⚠ Important

Quando você cria seu primeiro ambiente de serviço para SageMaker treinamento, cria AWS Batch automaticamente uma função vinculada ao serviço chamada `AWSServiceRoleForAWSBatchWithSagemaker` em sua conta. Essa função permite AWS Batch colocar em fila e gerenciar trabalhos SageMaker de treinamento em seu nome. Para obter mais informações sobre este perfil vinculado ao serviço e suas permissões, consulte [the section called “SageMaker função de integração”](#).

Crie um ambiente de serviço que possa lidar com até 5 instâncias:

```
aws batch create-service-environment \
  --service-environment-name TutorialServiceEnvironment \
  --service-environment-type SAGEMAKER_TRAINING \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=5
```

Saída:

```
{
  "serviceEnvironmentName": "TutorialServiceEnvironment",
  "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-id:service-
environment/TutorialServiceEnvironment"
}
```

Verifique se o ambiente de serviço foi criado com sucesso:

```
aws batch describe-service-environments --service-
environments TutorialServiceEnvironment
```

Saída:

```
{
  "serviceEnvironments": [
    {
      "serviceEnvironmentName": "TutorialServiceEnvironment",
      "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-
id:service-environment/TutorialServiceEnvironment",
      "serviceEnvironmentType": "SAGEMAKER_TRAINING",
```

```

        "state": "ENABLED",
        "status": "VALID",
        "capacityLimits": [
            {
                "maxCapacity": 5,
                "capacityUnit": "NUM_INSTANCES"
            }
        ],
        "tags": {}
    }
]
}

```

Para ter mais informações sobre ambientes de serviço, consulte [Ambientes de serviço](#).

Etapa 3: Crie sua fila SageMaker de trabalhos

Uma fila de SageMaker trabalhos gerencia o agendamento e a execução dos trabalhos de serviço. Os trabalhos enviados para essa fila serão enviados para seu ambiente de serviço com base na capacidade disponível.

Crie uma fila SageMaker de trabalhos de treinamento:

```

aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
  --priority 1 \
  --service-environment-order order=1,serviceEnvironment=TutorialServiceEnvironment

```

Saída:

```

{
  "jobQueueName": "my-sm-training-fifo-jq",
  "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq"
}

```

Verifique se a sua fila de trabalhos foi criada com sucesso:

```

aws batch describe-job-queues --job-queues my-sm-training-fifo-jq

```

Saída:

```
{
  "jobQueues": [
    {
      "jobQueueName": "my-sm-training-fifo-jq",
      "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-fifo-jq",
      "state": "ENABLED",
      "status": "VALID",
      "statusReason": "JobQueue Healthy",
      "priority": 1,
      "computeEnvironmentOrder": [],
      "serviceEnvironmentOrder": [
        {
          "order": 1,
          "serviceEnvironment": "arn:aws:batch:your-region:your-account-id:service-environment/TutorialServiceEnvironment"
        }
      ],
      "jobQueueType": "SAGEMAKER_TRAINING",
      "tags": {}
    }
  ]
}
```

Para obter mais informações sobre filas de SageMaker trabalhos, consulte [the section called “Criar uma fila de trabalhos do SageMaker”](#).

Etapa 4: criar e enviar um trabalho de treinamento

Agora você criará um trabalho de treinamento simples e o enviará para sua fila de trabalhos. Este exemplo usa um trabalho de treinamento básico do tipo “hello world” que demonstra a funcionalidade do trabalho de serviço.

Crie um arquivo chamado *my_training_job.json* com o conteúdo a seguir. *your-account-id* Substitua pelo ID AWS da sua conta:

Note

S3OutputPath é necessário para criar o trabalho de SageMaker treinamento, mas os resultados deste tutorial não são armazenados no bucket do Amazon S3 e você pode usar

o caminho no seguinte JSON. Em seu ambiente de produção, você precisará de um bucket Amazon S3 válido para armazenar a saída, se quiser.

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::your-account-id:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
      "hello world"
    ]
  },
  "ResourceConfig": {
    "InstanceType": "ml.c5.xlarge",
    "InstanceCount": 1,
    "VolumeSizeInGB": 1
  },
  "OutputDataConfig": {
    "S3OutputPath": "s3://your-s3-bucket/output"
  },
  "StoppingCondition": {
    "MaxRuntimeInSeconds": 30
  }
}
```

Envie o trabalho de treinamento usando a [SubmitServiceJobAPI](#):

```
aws batch submit-service-job \
  --job-queue my-sm-training-fifo-jq \
  --job-name my-batch-sm-job \
  --service-job-type SAGEMAKER_TRAINING \
  --retry-strategy attempts=1 \
  --timeout-config attemptDurationSeconds=60 \
  --service-request-payload file://my_training_job.json
```

Saída:

```
{
```



```

    "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
    "jobName": "my-batch-sm-job",
    "jobId": "your-job-id"
  }
}

```

Para obter mais informações sobre cargas úteis de trabalho de serviço, consulte [the section called “Cargas úteis de trabalho de serviço”](#). Para obter mais informações sobre como enviar trabalhos de serviço, consulte [the section called “Enviar um trabalho de serviço”](#).

Etapa 5: monitorar o status do trabalho

Você pode monitorar seus trabalhos de treinamento usando o seguinte AWS Batch APIs: [DescribeServiceJobListServiceJobs](#), [GetJobQueueSnapshot](#). Esta seção mostra maneiras diferentes de verificar o status do trabalho e as informações da fila.

Veja os trabalhos em execução na sua fila:

```

aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq --job-status RUNNING

```

Saída:

```

{
  "jobSummaryList": [
    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
      "jobId": "your-job-id",
      "jobName": "my-batch-sm-job",
      "serviceJobType": "SAGEMAKER_TRAINING",
      "status": "RUNNING",
      "startedAt": 1753718820
    }
  ]
}

```

```
}
```

Veja os trabalhos que estão no estado `RUNNABLE`:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq --job-status RUNNABLE
```

Tenha uma visão geral dos próximos trabalhos em sua fila:

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Saída:

```
{  
  "frontOfQueue": {  
    "jobs": [  
      {  
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",  
        "earliestTimeAtPosition": 1753718880  
      },  
      {  
        "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id-2",  
        "earliestTimeAtPosition": 1753718940  
      }  
    ],  
    "lastUpdatedAt": 1753718970  
  }  
}
```

Pesquise os trabalhos por nome:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --filters name=JOB_NAME,values="my-batch-sm-job"
```

Saída:

```
{  
  "jobSummaryList": [  

```

```

    {
      "latestAttempt": {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
        }
      },
      "createdAt": 1753718760,
      "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
      "jobId": "your-job-id",
      "jobName": "my-batch-sm-job",
      "serviceJobType": "SAGEMAKER_TRAINING",
      "status": "RUNNING"
    }
  ]
}

```

Para obter mais informações sobre o mapeamento de estado de trabalho, consulte [the section called “Status de serviço do trabalho”](#).

Etapa 6: ver a saída do trabalho

Depois que seu trabalho for concluído, você poderá visualizar a saída e os registros por meio da SageMaker IA APIs. AWS Batch

Obtenha informações detalhadas sobre seu trabalho em AWS Batch:

```

aws batch describe-service-job \
  --job-id your-job-id

```

Saída:

```

{
  "attempts": [
    {
      "serviceResourceId": {
        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/AWSBatch<my-simple-training-job><your-attempt-id>"
      },
      "startedAt": 1753718820,

```

```

        "stoppedAt": 1753718880,
        "statusReason": "Received status from SageMaker: Training job completed"
    },
    ],
    "createdAt": 1753718760,
    "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
    "jobId": "your-job-id",
    "jobName": "my-batch-sm-job",
    "jobQueue": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-
fifo-jq",
    "latestAttempt": {
        "serviceResourceId": {
            "name": "TrainingJobArn",
            "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
        }
    },
    "retryStrategy": {
        "attempts": 1,
        "evaluateOnExit": []
    },
    "serviceRequestPayload": "your-training-job-request-json",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "startedAt": 1753718820,
    "status": "SUCCEEDED",
    "statusReason": "Received status from SageMaker: Training job completed",
    "stoppedAt": 1753718880,
    "tags": {},
    "timeoutConfig": {
        "attemptDurationSeconds": 60
    }
}

```

Esse comando retorna informações abrangentes do trabalho, incluindo o ARN do trabalho de SageMaker treinamento, que você pode usar para acessar o trabalho diretamente por meio SageMaker da IA:

```

aws sagemaker describe-training-job \
  --training-job-name AWSBatch<my-simple-training-job><your-attempt-id>

```

Para ver os CloudWatch registros do seu trabalho de treinamento, primeiro obtenha o nome do fluxo de registros:

```
aws logs describe-log-streams \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name-prefix AWSBatchmy-simple-training-job
```

Saída:

```
{  
  "logStreams": [  
    {  
      "logStreamName": "your-log-stream-name",  
      "creationTime": 1753718830,  
      "firstEventTimestamp": 1753718840,  
      "lastEventTimestamp": 1753718850,  
      "lastIngestionTime": 1753718860,  
      "uploadSequenceToken": upload-sequence-token,  
      "arn": "arn:aws:logs:your-region:your-account-id:log-group:/aws/sagemaker/  
TrainingJobs:log-stream:AWSBatch<my-simple-training-job><your-attempt-id>/algo-1-algo-  
id",  
      "storedBytes": 0  
    }  
  ]  
}
```

Em seguida, recupere os logs usando o nome do fluxo de log da resposta anterior:

```
aws logs get-log-events \  
  --log-group-name /aws/sagemaker/TrainingJobs \  
  --log-stream-name your-log-stream-name
```

Saída:

```
{  
  "events": [  
    {  
      "timestamp": 1753718845,  
      "message": "hello world",  
      "ingestionTime": 1753718865  
    }  
  ],  
  "nextForwardToken": "next-forward-token",  
  "nextBackwardToken": "next-backward-token"  
}
```

```
}
```

A saída do log mostra a mensagem “hello world” do seu trabalho de treinamento, confirmando que o trabalho foi executado com sucesso.

Etapa 7: limpar os recursos do tutorial

Ao concluir este tutorial, limpe os recursos criados para evitar que causem cobranças.

Primeiro, desative e exclua a fila de trabalhos:

```
aws batch update-job-queue \  
  --job-queue my-sm-training-fifo-jq \  
  --state DISABLED
```

Aguarde até que a fila de trabalhos seja desativada e, em seguida, exclua-a:

```
aws batch delete-job-queue \  
  --job-queue my-sm-training-fifo-jq
```

Em seguida, desative e exclua o ambiente de serviço:

```
aws batch update-service-environment \  
  --service-environment TutorialServiceEnvironment \  
  --state DISABLED
```

Aguarde até que o ambiente de serviço seja desativado e, em seguida, exclua-o:

```
aws batch delete-service-environment \  
  --service-environment TutorialServiceEnvironment
```

Recursos adicionais do

Depois de concluir o tutorial, você pode querer explorar os seguintes tópicos:

- Recomendamos usar o PySDK para a criação de trabalhos de serviço e envio para sua fila de trabalhos porque o PySDK tem classes auxiliares e utilitários. Para ver um exemplo de uso do PySDK, consulte exemplos de [SageMaker IA](#) em. GitHub
- Saiba mais sobre [the section called “Trabalhos de serviço”](#).

- Explore [Cargas úteis de trabalho de serviço em AWS Batch](#) para configurações de trabalho de treinamento mais complexas.
- Saiba mais sobre o [Envie um trabalho de serviço em AWS Batch](#) e a API `SubmitServiceJob`.
- Revise [Mapeando AWS Batch o status do trabalho do serviço para o status de SageMaker IA](#) para entender as transições de estado do trabalho.
- Visite a [documentação do SageMaker AI Python SDK](#) para ver formas mais completas de criar e enviar trabalhos de treinamento SageMaker usando Python.
- Explore [SageMaker exemplos de notebooks](#) para fluxos de trabalho de aprendizado de máquina mais complexos.

O painel do widget AWS Batch

Por meio do painel do AWS Batch, você pode monitorar trabalhos recentes, as filas de trabalhos e os ambientes de computação. Por padrão, os seguintes widgets do painel são exibidos:

- Visão geral do trabalho: para obter mais informações sobre trabalhos do AWS Batch, consulte [Tarefas](#).
- Visão geral da fila de trabalhos: para obter mais informações fila de trabalhos do AWS Batch, consulte [Filas de tarefas](#).
- Visão geral do ambiente de computação: para obter mais informações sobre ambientes de computação do AWS Batch, consulte [Ambientes computacionais para AWS Batch](#).

Você pode personalizar os widgets que são exibidos na página do painel. As seções a seguir descrevem widgets adicionais que você pode adicionar.

Tópicos

- [Como adicionar o widget de fila de trabalhos única ao painel do AWS Batch](#).
- [Como adicionar o widget do CloudWatch Container Insights ao painel do AWS Batch](#)
- [Como adicionar o widget de logs de trabalhos ao painel do AWS Batch](#)

Como adicionar o widget de fila de trabalhos única ao painel do AWS Batch.

O widget de Fila de trabalho única exibe informações detalhadas sobre uma fila de trabalho única.

Para adicionar este widget, siga essas etapas.

1. Abra o [console de AWS Batch](#).
2. Na barra de navegação, escolha a Região da AWS desejada.
3. No painel de navegação, escolha Painel.
4. Escolha Adicionar widgets.
5. Em Fila de um único trabalho, escolha Adicionar widget.
6. Em Fila de trabalhos, selecione a fila de trabalhos desejada.
7. Em Status do trabalho, escolha os status do trabalho que você deseja exibir.

8. (Opcional) Desative Exibir ambientes de computação conectados se não desejar exibir as propriedades dos ambientes computacionais.
9. Em Propriedades do ambiente de computação, selecione as propriedades desejadas.
10. Escolha Adicionar.

Como adicionar o widget do CloudWatch Container Insights ao painel do AWS Batch

Esse widget exibe métricas agregadas para ambientes de computação e trabalhos do AWS Batch. Para obter mais informações sobre o Container Insights, consulte [the section called “CloudWatch Container Insights”](#).

Para adicionar este widget, siga essas etapas.

1. Abra o [console de AWS Batch](#).
2. Na barra de navegação, escolha a Região da AWS desejada.
3. No painel de navegação, escolha Painel.
4. Escolha Adicionar widgets.
5. Em Container insights, escolha Adicionar widget.
6. Em Ambiente de computação, escolha o ambiente de computação que você deseja.
7. Escolha Adicionar.

Como adicionar o widget de logs de trabalhos ao painel do AWS Batch

Esse widget exibe logs diferentes das trabalhos em um local conveniente. Para obter mais informações sobre os logs de trabalhos, consulte [the section called “Exibir registros de trabalhos em CloudWatch Registros”](#).

Para adicionar este widget, siga essas etapas.

1. Abra o [console de AWS Batch](#).
2. Na barra de navegação, escolha a Região da AWS desejada.
3. No painel de navegação, escolha Painel.

4. Escolha Adicionar widgets.
5. Em Logs de trabalhos, escolha Adicionar widget.
6. Em ID de trabalho, insira o ID do trabalho que você deseja.
7. Escolha Adicionar.

Ambientes computacionais para AWS Batch

As filas de trabalho são mapeadas para um ou mais ambientes de computação. Os ambientes de computação contêm as instâncias de contêiner do Amazon ECS que são usadas para executar trabalhos em lote em contêineres. Um ambiente de computação específico também podem ser mapeado para uma ou mais filas de trabalho. Em uma fila de trabalho, cada ambiente de computação associado tem um pedido que é usado pelo programador para determinar onde os trabalhos que estão prontos para serem executados serão executados. Se o primeiro ambiente de computação tiver um status de VALID e recursos disponíveis, o trabalho será agendado para uma instância de contêiner nesse ambiente de computação. Se o primeiro ambiente de computação tiver um status de INVALID ou não puder fornecer um recurso de computação adequado, o agendador tentará executar o trabalho no próximo ambiente de computação.

Tópicos

- [Ambientes de computação gerenciados](#)
- [Ambientes de computação não gerenciados](#)
- [Crie um ambiente de computação](#)
- [Atualizar um ambiente computacional no AWS Batch](#)
- [Recursos de computação de AMIs](#)
- [Use os modelos de lançamento do Amazon EC2 com AWS Batch](#)
- [Configuração do serviço de metadados de instância \(IMDS\)](#)
- [Configurações do EC2](#)
- [Estratégias de alocação de tipos de instâncias para o AWS Batch](#)
- [Gerenciamento de memória de recursos de computação](#)
- [Ambientes de computação Fargate](#)
- [Ambientes de computação Amazon EKS](#)

Ambientes de computação gerenciados

Você pode usar um ambiente computacional gerenciado para AWS Batch gerenciar a capacidade e os tipos de instância dos recursos computacionais dentro do ambiente. Isso se baseia nas especificações do recurso de computação que você define ao criar o ambiente de computação. Você pode optar por usar instâncias sob demanda do Amazon EC2 e instâncias spot do Amazon EC2.

Como alternativa, você pode usar a capacidade do Fargate e do Fargate Spot em seu ambiente de computação gerenciado. Ao usar instâncias spot, você pode, como opção, definir um preço máximo. Assim, as instâncias spot são iniciadas apenas quando o preço da instância spot estiver abaixo de uma porcentagem especificada do preço sob demanda.

 Important

As instâncias Fargate Spot não são suportadas no Windows containers on AWS Fargate. Uma fila de trabalhos será bloqueada se um FargateWindows trabalho for enviado a uma fila de trabalhos que usa somente ambientes computacionais Fargate Spot.

 Important

AWS Batch cria e gerencia vários AWS recursos em seu nome e dentro da sua conta, incluindo modelos de lançamento do Amazon EC2, grupos do Amazon EC2 Auto Scaling, frotas spot do Amazon EC2 e clusters do Amazon ECS. Esses recursos gerenciados são configurados especificamente para garantir a operação ideal do AWS Batch. A modificação manual desses recursos gerenciados em lote, a menos que seja explicitamente declarado na documentação AWS Batch, pode resultar em comportamento inesperado, resultando em ambiente computacional INVALID, comportamento de escalabilidade de instâncias abaixo do ideal, atraso no processamento do workload ou custos inesperados. Essas modificações manuais não podem ser sustentadas de forma determinística pelo AWS Batch serviço. Sempre use o Batch compatível APIs ou o console Batch para gerenciar seus ambientes de computação.

Ambientes de computação gerenciados iniciam instâncias do Amazon EC2 na VPC e nas sub-redes que você especifica e, em seguida, as registram em um cluster do Amazon ECS. As instâncias do Amazon EC2 precisam de acesso de rede externo para se comunicar com o endpoint de serviço do Amazon ECS. Algumas sub-redes não fornecem às instâncias do Amazon EC2 endereços IP públicos. Se suas instâncias do Amazon EC2 não tiverem um endereço IP público, elas deverão usar a conversão de endereço de rede (NAT) para obter esse acesso. Para obter mais informações, consulte [Gateways NAT](#) no Guia do usuário da Amazon VPC. Para obter mais informações sobre como criar uma VPC, consulte [Criar uma nuvem privada virtual](#).

Por padrão, os ambientes computacionais AWS Batch gerenciados usam uma versão recente e aprovada da AMI otimizada do Amazon ECS para recursos computacionais. No entanto, você pode

querer criar sua própria AMI a ser usada para seus ambientes de computação gerenciados por vários motivos. Para obter mais informações, consulte [Recursos de computação de AMIs](#).

Note

AWS Batch não atualiza automaticamente o AMIs em um ambiente computacional após sua criação. Por exemplo, ele não atualiza o AMIs em seu ambiente computacional quando uma versão mais recente da AMI otimizada do Amazon ECS é lançada. Você é responsável pelo gerenciamento do sistema operacional convidado. Isso inclui quaisquer atualizações e patches de segurança. Você também é responsável por quaisquer outros utilitários ou aplicativos de software que instalar nos recursos de computação. Há duas maneiras de usar uma nova AMI para seus AWS Batch trabalhos. O método original é concluir as seguintes etapas:

1. Crie um novo ambiente de computação com a nova AMI.
2. Adicione o ambiente de computação a uma fila de trabalhos existente.
3. Remova o antigo ambiente de computação da fila de trabalhos.
4. Exclua o ambiente de computação anterior.

Em abril de 2022, AWS Batch foi adicionado suporte aprimorado para atualização de ambientes computacionais. Para obter mais informações, consulte [Atualizar um ambiente computacional no AWS Batch](#). Para usar a atualização aprimorada de ambientes computacionais para atualizar AMIs, siga estas regras:

- Não defina o parâmetro da função de serviço ([serviceRole](#)) nem o defina como a função AWSServiceRoleForBatch vinculada ao serviço.
- Defina o parâmetro da estratégia de alocação ([allocationStrategy](#)) como BEST_FIT_PROGRESSIVE, SPOT_CAPACITY_OPTIMIZED ou SPOT_PRICE_CAPACITY_OPTIMIZED.
- Defina o parâmetro de atualização para a versão mais recente da imagem ([updateToLatestImageVersion](#)) como true.
- Não especifique uma ID de AMI em [imageId](#), [imageIdOverride](#) (em [ec2Configuration](#)) ou no modelo de execução ([launchTemplate](#)). Nesse caso, AWS Batch seleciona a AMI otimizada mais recente do Amazon ECS que é suportada AWS Batch no momento em que a atualização da infraestrutura é iniciada. Como alternativa, é possível especificar o ID da AMI os parâmetros `imageId` ou `imageIdOverride` ou o

modelo de inicialização identificado pelas propriedades `LaunchTemplate`. A alteração de qualquer uma dessas propriedades inicia uma atualização da infraestrutura. Se a ID da AMI for especificada no modelo de execução, ela não poderá ser substituída pela especificação de uma ID da AMI nos parâmetros `imageId` ou `imageIdOverride`. Ela só pode ser substituída especificando um modelo de lançamento diferente. Ou, se a versão do modelo de lançamento estiver definida como `$Default` ou `$Latest`, definindo uma nova versão padrão para o modelo de execução (se for `$Default`) ou adicionando uma nova versão ao modelo de execução (se for `$Latest`).

Se essas regras forem seguidas, qualquer atualização que inicie uma atualização de infraestrutura fará com que o ID da AMI seja novamente selecionado. Se a configuração da [version](#) no modelo de execução ([launchTemplate](#)) for definida como `$Latest` ou `$Default`, a versão mais recente ou padrão do modelo de inicialização será avaliada no momento da atualização da infraestrutura, mesmo que o [launchTemplate](#) não tenha sido atualizado.

Consideração ao criar trabalhos paralelos de vários nós

AWS Batch recomenda a criação de ambientes de computação dedicados para executar trabalhos paralelos de vários nós (MNP) e trabalhos não MNP. Isso se deve à forma como a capacidade de computação é criada em seu ambiente de computação gerenciado. Ao criar um novo ambiente de computação gerenciado, se você especificar um `minvCpu` valor maior que zero, AWS Batch criará um pool de instâncias para uso somente com trabalhos não MNP. Se um trabalho paralelo de vários nós for enviado, AWS Batch criará uma nova capacidade de instância para executar os trabalhos paralelos de vários nós. Nos casos em que há trabalhos paralelos de um e vários nós em execução no mesmo ambiente computacional em que um `maxvCpus` valor `minvCpus` ou é definido, se os recursos computacionais necessários não estiverem disponíveis, AWS Batch aguardará a conclusão dos trabalhos atuais antes de criar os recursos computacionais necessários para executar os novos trabalhos.

Ambientes de computação não gerenciados

Em um ambiente computacional não gerenciado, você gerencia seus próprios recursos computacionais. AWS Batch oferece suporte a ambientes computacionais não gerenciados para o

Amazon ECS e o Amazon EKS, permitindo que você mantenha o controle sobre sua infraestrutura enquanto aproveita os recursos de agendamento de tarefas do Batch.

 Note

AWS Os recursos do Fargate não são suportados em ambientes computacionais não gerenciados.

Ambientes computacionais não gerenciados do Amazon ECS

Para ambientes computacionais não gerenciados do Amazon ECS, você deve verificar se a AMI que você usa para seus recursos computacionais atende à especificação da AMI da instância de contêiner do Amazon ECS. Para obter mais informações, consulte [Especificação da AMI do recurso de computação](#) e [Tutorial: criar uma AMI de recurso de computação](#).

Depois de criar seu ambiente computacional não gerenciado, use a operação de [DescribeComputeEnvironments](#) API para ver os detalhes do ambiente computacional. Encontre o cluster do Amazon ECS que está associado ao ambiente, e inicie manualmente suas instâncias de contêiner nesse cluster do Amazon ECS.

O AWS CLI comando a seguir também fornece o ARN do cluster Amazon ECS.

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedCE \
  --query "computeEnvironments[].ecsClusterArn"
```

Para obter mais informações, consulte [Iniciando uma instância de contêiner do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service. Ao executar os recursos de computação, especifique o ARN do cluster do Amazon ECS que os recursos devem registrar com os seguintes dados de usuário do Amazon EC2. *ecsClusterArn* Substitua pelo ARN do cluster que você obteve com o comando anterior.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

Ambientes computacionais Amazon EKS não gerenciados

Em um ambiente computacional não gerenciado do Amazon EKS, você gerencia seus próprios nós do Kubernetes enquanto AWS Batch gerencia o agendamento e a colocação de trabalhos. Ele permite que você controle diretamente sua infraestrutura Kubernetes quanto aos requisitos operacionais, de segurança ou de conformidade. Você é responsável por provisionar e configurar seus nós do Amazon EKS, ao mesmo tempo em que AWS Batch se integra ao seu cluster Amazon EKS existente para agendar e executar trabalhos.

Para obter mais informações, consulte [Tutorial: Crie um ambiente computacional não gerenciado usando os recursos do Amazon EKS](#).

Crie um ambiente de computação

Antes de executar trabalhos no AWS Batch, você precisa criar um ambiente computacional. Você pode criar um ambiente computacional gerenciado onde AWS Batch gerencia as instâncias do Amazon EC2 ou os recursos do AWS Fargate dentro do ambiente com base em suas especificações. Ou, como alternativa, você pode criar um ambiente de computação não gerenciado onde você gerencia a configuração da instância do Amazon EC2 dentro do ambiente.

Important

As instâncias Fargate Spot não são suportadas nos seguintes cenários:

- Windows containers on AWS Fargate

Uma fila de trabalhos será bloqueada nesses cenários se um trabalho for enviado a uma fila de trabalhos que usa somente ambientes de computação Fargate Spot.

Tópicos

- [Tutorial: criar um ambiente de computação gerenciado usando recursos do Fargate](#)
- [Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EC2](#)
- [Tutorial: criar um ambiente de computação não gerenciado usando recursos do Amazon EC2](#)
- [Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EKS](#)
- [Tutorial: Crie um ambiente computacional não gerenciado usando os recursos do Amazon EKS](#)
- [Recurso: modelo de ambiente de computação](#)

- [Tabela de computação do tipo de instância](#)

Tutorial: criar um ambiente de computação gerenciado usando recursos do Fargate

Conclua as etapas a seguir para criar um ambiente computacional gerenciado usando recursos do AWS Fargate.

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Ambientes de computação.
4. Escolha Criar.
5. Configure o ambiente de computação.

Note

Os ambientes de computação para trabalhos Windows containers on AWS Fargate devem ter pelo menos uma vCPU.

- a. Para configuração do ambiente de computação, escolha Fargate.
 - b. Para Nome, especifique um nome exclusivo para seu ambiente de computação. O nome pode conter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
 - c. Em Perfil de serviço, escolha o perfil vinculado ao serviço que permite que o serviço AWS Batch faça chamadas para as operações de API da AWS necessárias em seu nome. Por exemplo, escolha AWSServiceRoleForBatch. Para obter mais informações, consulte [Usar perfis vinculados a serviço do AWS Batch](#).
 - d. (Opcional) Expanda as Tags. Para adicionar uma tag, escolha Adicionar tag. Insira uma chave e um Valor opcional. Escolha Adicionar Tag.
 - e. Escolha Próxima página.
6. Na seção Instance configuration:
 - a. (Opcional) Para usar a capacidade do Fargate Spot, ative o Fargate Spot. Para obter informações sobre o Fargate Spot, consulte [Usando o Amazon EC2 Spot](#) e o Fargate_Spot.

- b. Para Máximo de vCPUs, escolha o número máximo de vCPUs para o qual seu ambiente de computação deve aumentar, independentemente da demanda da fila de trabalhos.
 - c. Escolha Próxima página.
7. Configure redes.

 Important

Recursos de computação precisam de acesso para se comunicar com o endpoint de serviço do Amazon ECS. Isso pode ser feito por meio de uma interface do endpoint da VPC ou por meio das instâncias de contêiner que tenham endereços IP públicos. Para obter mais informações sobre endpoints da VPC de interface, consulte [Endpoints da VPC de interface do Amazon ECS \(AWS PrivateLink\)](#) no Manual do Desenvolvedor do Amazon Elastic Container Service.

Se você não tiver um endpoint da VPC de interface configurado e seus contêineres não tiverem endereços IP públicos, eles deverão usar a conversão de endereço de rede (NAT) para fornecer esse acesso. Para obter mais informações, consulte [Gateways NAT](#) no Guia do usuário da Amazon VPC. Para obter mais informações, consulte [the section called "Crie uma VPC"](#).

- a. Para ID de Nuvem Privada Virtual (VPC), escolha uma VPC na qual você deseja iniciar suas instâncias.
- b. Em Sub-redes, selecione a sub-rede que será usada. Por padrão, todas as sub-redes dentro da VPC selecionadas estão disponíveis.

 Note

AWS Batch no Fargate atualmente não é compatível com Local Zones. Para obter mais informações, consulte [Amazon ECS clusters em Local Zones, Wavelength Zones, e AWS Outposts](#) no Amazon Elastic Container Service Developer Guide.

- c. Para Security groups, escolha um security group a ser anexado às suas instâncias. Por padrão, o grupo de segurança padrão para sua VPC é escolhido.
 - d. Escolha Próxima página.
8. Para Revisar, reveja as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar ambiente de computação.

Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EC2

Conclua as etapas a seguir para criar um ambiente computacional gerenciado usando recursos do Amazon Elastic Compute Cloud (Amazon EC2).

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Ambientes.
4. Selecione Seus ambientes e, depois, selecione Criar ambiente.
5. Configure o ambiente.
 - a. Em configuração do ambiente de computação, escolha Amazon Elastic Compute Cloud (Amazon EC2).
 - b. Em Tipo de orquestração, escolha Gerenciado.
 - c. Para Nome, especifique um nome exclusivo para seu ambiente de computação. O nome pode conter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
 - d. Em Perfil de serviço, escolha o perfil vinculado ao serviço que permite que o serviço AWS Batch faça chamadas para as operações de API da AWS necessárias em seu nome. Por exemplo, escolha AWSServiceRoleForBatch. Para obter mais informações, consulte [Usar perfis vinculados a serviço do AWS Batch](#).
 - e. Para o Instance role, escolha criar um novo perfil de instância ou use um perfil de instância existente que tenha as permissões de IAM necessárias anexadas. Esse perfil de instância permite que as instâncias de contêiner do Amazon ECS que são criadas para seu ambiente de computação façam chamadas para as APIs da AWS necessárias em seu nome. Para obter mais informações, consulte [Perfil de instância do Amazon ECS](#). Se você optou por criar um novo perfil de instância, a função necessária (ecsInstanceRole) será criada para você.
 - f. (Opcional) Expanda as Tags.
 - i. (Opcional) Para tags EC2, escolha Adicionar tag para adicionar uma tag aos recursos que são lançados no ambiente computacional. Insira uma chave e um Valor opcional. Escolha Adicionar Tag.

- ii. (Opcional) Em Tags, escolha Adicionar tag. Insira uma chave e um Valor opcional. Escolha Adicionar Tag.

Para obter mais informações, consulte [Marcar com tag os recursos do AWS Batch](#).

- g. Escolha Próximo.
6. Na seção Instance configuration:
- a. (Opcional) Para Habilitar o uso de instâncias Spot, ative o Spot. Para obter mais informações, consulte [Spot Instances](#).
 - b. (Somente Spot) Em Maximum % on-demand price, escolha a porcentagem máxima que o preço que uma instância spot deve ter em comparação com o preço sob demanda para esse tipo de instância antes que as instâncias sejam executadas. Por exemplo, se o preço máximo for 20%, o preço spot deverá estar abaixo de 20% do preço atual sob demanda para essa instância do EC2. Você sempre paga o menor preço (mercado) e nunca mais do que sua porcentagem máxima. Se você deixar esse campo em branco, o valor padrão será 100% do preço sob demanda.
 - c. (Somente Spot) Para o perfil frota Spot, escolha uma função de IAM existente do Amazon EC2 Frota Spot para aplicar ao seu ambiente de computação Spot. Se você ainda não tiver um perfil do IAM da frota spot do Amazon EC2 existente, crie um primeiro. Para obter mais informações, consulte [Perfil de frota spot Amazon EC2](#).

 Important

Para marcar suas instâncias Spot na criação, sua função IAM do Amazon EC2 Frota Spot deve usar a política gerenciada mais recente do AmazonEC2SpotFleetTaggingRole. A política gerenciada AmazonEC2SpotFleetRole não tem as permissões necessárias para marcar as instâncias spot. Para obter mais informações, consulte [Instâncias spot sem tags na criação](#) e [the section called “Marcar com tag os recursos do”](#).

- d. Para Mínimo de vCPUs, escolha o número mínimo de vCPUs que seu ambiente de computação deve manter, independentemente da demanda da fila de trabalhos.
- e. Para vCPUs desejados, escolha o número de vCPUs com o qual seu ambiente de computação deve ser iniciado. À medida que a demanda da fila de trabalhos aumenta, o AWS Batch pode aumentar o número desejado de vCPUs no seu ambiente de computação e adicionar instâncias do EC2, até o máximo de vCPUs. À medida que a demanda diminui,

o AWS Batch pode diminuir o número desejado de vCPUs no seu ambiente de computação e remover instâncias, até o mínimo de vCPUs.

- f. Para Máximo de vCPUs, escolha o número máximo de vCPUs para o qual seu ambiente de computação deve aumentar, independentemente da demanda da fila de trabalhos.
- g. Para Tipos de instância permitidos, escolha os tipos de instância do Amazon EC2 que podem ser iniciados. Você pode especificar famílias de instâncias para iniciar qualquer tipo de instância dentro dessas famílias (por exemplo c5, c5n, ou p3). Ou você pode especificar tamanhos específicos dentro de uma família (como c5.xlarge). Os tipos de instância Metal não estão nas famílias de instâncias. Por exemplo, c5 não inclui c5.metal.

AWS Batch pode selecionar o tipo de instância para você se escolher uma das seguintes opções:

- `optimal` para selecionar tipos de instância (das famílias de instâncias c4, m4, r4, c5, m5 e r5) que correspondam à demanda de suas filas de trabalho.
- `default_x86_64` para selecionar tipos de instância baseados em x86 (das famílias de instâncias m6i, c6i, r6i e c7i) que correspondam à demanda de suas filas de trabalho.
- `default_arm64` para selecionar tipos de instância baseados em ARM (das famílias de instâncias m6g, c6g, r6g e c7g) que correspondam à demanda de suas filas de trabalho.

 Note

A partir de 01/11/2025, o comportamento do `optimal` será alterado para corresponder com `default_x86_64`. Durante a mudança, suas famílias de instâncias podem ser atualizadas para uma nova geração. Não há necessidade de realizar nenhuma ação para que a atualização ocorra. Para obter mais informações sobre a mudança, consulte

 Note

A partir de 01/11/2025, o comportamento do `optimal` será alterado para corresponder com `default_x86_64`.

Durante a mudança, suas famílias de instâncias podem ser

atualizadas para uma nova geração. Não há necessidade de realizar nenhuma ação para que a atualização ocorra.

AWS Batch suportou uma única opção em `instanceTypes` `optimal` para atender à demanda de suas filas de trabalho. Introduzimos duas novas opções de tipo de instância: `default_x86_64` e `default_arm64`. Usaremos `default_x86_64` se você não fizer nenhuma seleção do tipo de instância. Essas novas opções selecionarão automaticamente tipos de instância econômicos em diferentes famílias e gerações com base nos requisitos da fila de trabalhos, permitindo que você execute seus workloads rapidamente.

À medida que a capacidade suficiente de novos tipos de instância for disponibilizada em um Região da AWS, o pool padrão correspondente será atualizado automaticamente com o novo tipo de instância. A opção `optimal` existente continuará sendo compatível e não está sendo descontinuada, pois será compatível com os pools padrão subjacentes para fornecer instâncias atualizadas daqui para frente. Se você estiver usando `optimal`, não há necessidade de realizar nenhuma ação.

No entanto, esteja ciente de que somente `ENABLED` e `VALID` Compute Environments (CEs) serão atualizados automaticamente com novos tipos de instância. Se você tiver algum `DISABLED` ou `INVALID` CEs, eles receberão atualizações assim que forem reativados e definidos para um `VALID` estado.

 Note

- A disponibilidade da família de instâncias varia por Região da AWS. Por exemplo, algumas Região da AWS podem não ter nenhuma família de instâncias de quarta geração, mas ter famílias de instâncias de quinta e sexta geração.
- Ao usar os pacotes de instâncias `default_x86_64` ou `default_arm64`, AWS Batch seleciona famílias de instâncias com base em um equilíbrio entre custo-benefício e desempenho. Embora as instâncias de nova geração geralmente

ofereçam melhor relação preço-desempenho, o AWS Batch pode escolher uma família de instâncias de geração anterior se ela fornecer a combinação ideal de disponibilidade, custo e desempenho para seu workload. Por exemplo, em um Região da AWS em que ambas as instâncias c6i e c7i estejam disponíveis, AWS Batch pode selecionar instâncias c6i se elas oferecerem melhor custo-benefício para seus requisitos específicos de trabalho. Para obter mais informações sobre tipos de instâncias AWS Batch e disponibilidade Região da AWS, consulte a [Tabela de computação do tipo de instância](#).

- AWS Batch atualiza periodicamente suas instâncias em pacotes padrão para opções mais novas e econômicas. As atualizações acontecem automaticamente sem exigir nenhuma ação de sua parte. Seus workloads continuam em execução durante as atualizações sem interrupção.

 Note

Ao criar um ambiente de computação, os tipos de instância selecionados para ele devem compartilhar a mesma arquitetura. Por exemplo, você não pode misturar instâncias ARM e x86 no mesmo ambiente de computação.

 Note

O AWS Batch dimensionará GPUs com base na quantidade necessária nas filas de trabalho. Para usar o agendamento de GPU, o ambiente computacional deve incluir tipos de instância das famílias p6, p3, p4, p5, g3, g3s, g4, g5 ou g6.

- h. Para Estratégia de alocação, escolha a estratégia de alocação a ser usada ao selecionar tipos de instância na lista de tipos de instância permitidos. O BEST_FIT_PROGRESSIVE geralmente é a melhor opção para ambientes de computação sob demanda do EC2, SPOT_CAPACITY_OPTIMIZED e SPOT_PRICE_CAPACITY_OPTIMIZED para ambientes de computação EC2 Spot. Para obter mais informações, consulte [the section called “Estratégias de alocação de tipos de instâncias”](#).
- i. Expanda Additional configuration.

- i. (Opcional) Em Grupo de posicionamento, insira um nome de grupo de posicionamento para agrupar recursos no ambiente de computação.
- ii. (Opcional) Em Par de chaves do EC2, escolha um par de chaves pública e privada como credenciais de segurança ao se conectar à instância. Para obter mais informações sobre pares de chaves do Amazon EC2, consulte [Pares de chaves do Amazon EC2 e instâncias do Linux](#).
- iii. (Opcional) Para a configuração do EC2, escolha o tipo de imagem e os valores de Substituição do ID da imagem para fornecer informações para a AWS Batch selecionar as imagens de máquina da Amazon (AMIs) para instâncias no ambiente computacional. Se a substituição de ID da imagem não for especificada para cada tipo de imagem, o AWS Batch seleciona uma [AMI otimizada do Amazon ECS recente](#). Se nenhum tipo de imagem for especificado, o padrão será um Amazon Linux 2 para uma instância não GPU e não AWS Graviton.

 Important

Para usar uma AMI personalizada, escolha o tipo de imagem e insira a ID da AMI personalizada na caixa de substituição de ID da imagem.

[Amazon Linux 2](#)

Padrão para todas as famílias de instâncias baseadas no AWS Graviton (por exemplo, C6g, M6g, R6g e T4g) e pode ser usado para todos os tipos de instâncias não GPU.

[Amazon Linux 2 \(GPU\)](#)

Padrão para todas as famílias de instâncias GPU (por exemplo, P4 e G4) e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.

[Amazon Linux 2023](#)

AWS Batch é compatível com o Amazon Linux 2023.

 Note

O Amazon Linux 2023 não é compatível com instâncias A1.

Amazon Linux 2023 (GPU)

Padrão para todas as famílias de instâncias GPU (por exemplo, P4 e G4) e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.

Note

A AMI que você escolher para um ambiente de computação deve corresponder à arquitetura dos tipos de instância que você pretende usar para este ambiente. Por exemplo, se o ambiente de computação usar tipos de instância A1, a AMI de recursos de computação escolhida deverá oferecer suporte a instâncias ARM. O Amazon ECS vende as versões x86 e ARM da AMI do Amazon Linux 2 otimizada para Amazon ECS. Para obter mais informações, consulte [AMI do Amazon Linux 2 otimizada para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

j. (Opcional) Expandir Modelos de execução

- i. Em Modelo de execução padrão, selecione um modelo de execução existente do Amazon EC2 para configurar seus recursos de computação. A versão padrão do modelo é preenchida automaticamente. Para obter mais informações, consulte [Use os modelos de lançamento do Amazon EC2 com AWS Batch](#).

Note

Em um modelo de execução, é possível especificar uma AMI personalizada que você tenha criado.

- ii. (Opcional) Em Versão padrão, insira `$Default`, `$Latest` ou um número de versão específico para ser usado.

Note

Nota: se você usar uma variável de substituição (`$Default` ou `$Latest`), será aplicado o número da versão padrão atual ou o número da versão mais recente no momento em que essa configuração for salva. Se a versão padrão ou mais

recente mudar no futuro, você deverá atualizar as informações. Elas não serão atualizadas automaticamente.

 Important

Se o parâmetro de versão do modelo de execução for `$Default` ou `$Latest`, a versão padrão ou mais recente do modelo de execução especificado será avaliada durante uma atualização de infraestrutura. Se uma ID de AMI diferente for selecionada por padrão ou se a versão mais recente do modelo de execução for selecionada, essa ID de AMI será usada na atualização. Para obter mais informações, consulte [the section called “Seleção da AMI durante atualizações de infraestrutura”](#).

- iii. (Opcional) Em Modelo de execução de substituição, escolha Adicionar modelo de execução de substituição.
 - A. (Opcional) Em Modelo de execução, selecione um modelo de execução existente do Amazon EC2 para usar em tipos e famílias de instâncias específicos.
 - B. (Opcional) Em Versão padrão, insira um número de versão específico para ser usado, `$Default` ou `$Latest`.

 Note

Se você usar a variável `$Default` ou `$Latest`, o AWS Batch aplicará as informações atuais no momento em que o ambiente computacional for criado. Se a versão padrão ou mais recente for alterada no futuro, você deverá atualizar as informações por meio de [UpdateComputeEnvironment](#) ou por meio de Console de gerenciamento da AWS - AWS Batch.

- C. (Opcional) Para Tipos de instância de destino, selecione o tipo de instância ou a família à qual você deseja aplicar o modelo de execução de substituição.

 Note

Se você especificar um modelo de execução de substituição, os Tipos de instância de destino serão obrigatórios. Para obter mais informações, consulte [LaunchTemplateSpecificationOverride.targetInstanceTypes](#).

 Note

Se o tipo de instância ou a família que você deseja selecionar não aparecer nessa lista, revise as seleções feitas em `Allowed instance types`.

k. Escolha Próximo.

7. Na seção Configuração de rede:

 Important

Recursos de computação precisam de acesso para se comunicar com o endpoint de serviço do Amazon ECS. Isso pode ser feito por meio de uma interface do endpoint da VPC ou por meio das instâncias de contêiner que tenham endereços IP públicos. Para obter mais informações sobre endpoints da VPC de interface, consulte [Endpoints da VPC de interface do Amazon ECS \(AWS PrivateLink\)](#) no Manual do Desenvolvedor do Amazon Elastic Container Service.

Se você não tiver um endpoint da VPC de interface configurado e seus das instâncias de contêiner não tiverem endereços IP públicos, eles deverão usar a conversão de endereço de rede (NAT) para fornecer esse acesso. Para obter mais informações, consulte [Gateways NAT](#) no Guia do usuário da Amazon VPC. Para obter mais informações, consulte [the section called “Crie uma VPC”](#).

- a. Em Virtual Private Cloud (VPC) ID (Nuvem Privada Virtual Private Cloud), escolha uma VPC na qual executar suas instâncias.
- b. Em Sub-redes, selecione a sub-rede que será usada. Por padrão, todas as sub-redes dentro da VPC selecionadas estão disponíveis.

 Note

AWS Batch no Amazon EC2 oferece suporte a Zonas Locais. Para obter mais informações, consulte [Zonas Locais](#) no Guia do usuário do Amazon EC2 e [Clusters do Amazon ECS em zonas locais, zonas do Wavelength, e AWS Outposts](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

- c. (Opcional) Para Security groups, escolha um grupo de segurança a ser anexado às suas instâncias. Por padrão, o grupo de segurança padrão para sua VPC é escolhido.

 Note

Nota: se você usar uma variável de substituição (\$Default ou \$Latest), será aplicado o número da versão padrão atual ou o número da versão mais recente no momento em que essa configuração for salva. Se a versão padrão ou mais recente mudar no futuro, você deverá atualizar as informações. Elas não serão atualizadas automaticamente.

8. Escolha Próxima página.
9. Para Revisar, reveja as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar ambiente de computação.

Tutorial: criar um ambiente de computação não gerenciado usando recursos do Amazon EC2

Conclua as etapas a seguir para criar um ambiente computacional não gerenciado usando recursos do Amazon Elastic Compute Cloud (Amazon EC2).

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. Na página Ambientes computacionais selecione Criar.
4. Configure o ambiente.
 - a. Em configuração do ambiente de computação, escolha Amazon Elastic Compute Cloud (Amazon EC2).

- b. Em Tipo de orquestração, escolha Não gerenciado.
5. Para Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
6. Em Perfil de serviço, escolha um perfil que permite que o serviço AWS Batch faça chamadas para as operações de API da AWS necessárias em seu nome.

 Note

Não é possível usar `AWSServiceRoleForBatch` para ambientes computacionais não gerenciados.

7. Para Máximo de vCPUs, escolha o número máximo de vCPUs para o qual seu ambiente de computação deve aumentar, independentemente da demanda da fila de trabalhos.
8. (Opcional) Expanda as Tags. Para adicionar uma tag, escolha Adicionar tag. Insira uma chave e um Valor opcional. Escolha Adicionar Tag. Para obter mais informações, consulte [Marcar com tag os recursos do AWS Batch](#).
9. Escolha Próxima página.
10. Para Revisar, reveja as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar ambiente de computação.

Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EKS

Conclua as seguintes etapas para criar um ambiente computacional gerenciada usando recursos do Amazon Elastic Kubernetes Service (Amazon EKS).

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Ambientes de computação.
4. Escolha Criar.
5. Para a configuração do ambiente de computação, escolha Amazon Elastic Kubernetes Service (Amazon EKS).

6. Para Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
7. Em Perfil de instância, escolha um perfil de instância existente que tenha as permissões de IAM necessárias anexadas.

 Note

Para criar um ambiente de computação no AWS Batch console, escolha um perfil de instância que tenha as permissões `eks:ListClusters` e `eks:DescribeCluster`.

8. Para o cluster EKS, escolha um cluster existente do Amazon EKS.
9. Em Namespace, insira um Kubernetes namespace para agrupar seus AWS Batch processos no cluster.
10. (Opcional) Expanda as Tags. Escolha Adicionar tag e, em seguida, insira um par chave-valor.
11. Escolha Próxima página.
12. (Opcional) Para usar instâncias spot do EC2, ative Habilitar o uso de instâncias spot para usar instâncias spot do Amazon EC2.
13. (Somente Spot) Em Maximum % on-demand price, escolha a porcentagem máxima que o preço que uma instância spot deve ter em comparação com o preço sob demanda para esse tipo de instância antes que as instâncias sejam executadas. Por exemplo, se o preço máximo for 20%, o preço spot deverá estar abaixo de 20% do preço atual sob demanda para essa instância do EC2. Você sempre paga o menor preço (mercado) e nunca mais do que sua porcentagem máxima. Se você deixar esse campo em branco, o valor padrão será 100% do preço sob demanda.
14. (Somente spot) Para o perfil de frota spot, escolha o perfil IAM da frota spot do Amazon EC2 para o ambiente SPOT de computação.

 Important

Esse perfil é necessário se a estratégia de alocação definida para o BEST_FIT ou se a estratégia de alocação não for especificada.

15. (Opcional) Para Mínimo de vCPUs, escolha o número mínimo de vCPUs que seu ambiente de computação deve manter, independentemente da demanda da fila de trabalhos.

16. (Opcional) Para Máximo de vCPUs, escolha o número máximo de vCPUs para o qual seu ambiente de computação deve aumentar, independentemente da demanda da fila de trabalhos.
17. Para Tipos de instância permitidos, escolha os tipos de instância do Amazon EC2 que podem ser iniciados. Você pode especificar famílias de instâncias para iniciar qualquer tipo de instância dentro dessas famílias (por exemplo c5, c5n, ou p3). Ou você pode especificar tamanhos específicos dentro de uma família (como c5.8xlarge). Os tipos de instância Metal não estão nas famílias de instâncias. Por exemplo, c5 não inclui c5.metal.

AWS Batch pode selecionar o tipo de instância para você se escolher uma das seguintes opções:

- `optimal` para selecionar tipos de instância (das famílias de instâncias c4, m4, r4, c5, m5 e r5) que correspondam à demanda de suas filas de trabalho.
- `default_x86_64` para selecionar tipos de instância baseados em x86 (das famílias de instâncias m6i, c6i, r6i e c7i) que correspondam à demanda de suas filas de trabalho.
- `default_arm64` para selecionar tipos de instância baseados em ARM (das famílias de instâncias m6g, c6g, r6g e c7g) que correspondam à demanda de suas filas de trabalho.

 Note

A partir de 01/11/2025, o comportamento do `optimal` será alterado para corresponder com `default_x86_64`. Durante a mudança, suas famílias de instâncias podem ser atualizadas para uma nova geração. Não há necessidade de realizar nenhuma ação para que a atualização ocorra. Para obter mais informações sobre a mudança, consulte [Configuração ideal do tipo de instância para receber atualizações automáticas da família de instâncias](#).

 Note

- A disponibilidade da família de instâncias varia por Região da AWS. Por exemplo, algumas Região da AWS podem não ter nenhuma família de instâncias de quarta geração, mas ter famílias de instâncias de quinta e sexta geração.
- Ao usar os pacotes de instâncias `default_x86_64` ou `default_arm64`, AWS Batch seleciona famílias de instâncias com base em um equilíbrio entre custo-benefício e desempenho. Embora as instâncias de nova geração geralmente ofereçam melhor

relação preço-desempenho, o AWS Batch pode escolher uma família de instâncias de geração anterior se ela fornecer a combinação ideal de disponibilidade, custo e desempenho para seu workload. Por exemplo, em um Região da AWS em que ambas as instâncias c6i e c7i estejam disponíveis, AWS Batch pode selecionar instâncias c6i se elas oferecerem melhor custo-benefício para seus requisitos específicos de trabalho. Para obter mais informações sobre tipos de instâncias AWS Batch e disponibilidade Região da AWS, consulte a [Tabela de computação do tipo de instância](#).

- AWS Batch atualiza periodicamente suas instâncias em pacotes padrão para opções mais novas e econômicas. As atualizações acontecem automaticamente sem exigir nenhuma ação de sua parte. Seus workloads continuam em execução durante as atualizações sem interrupção

 Note

Ao criar um ambiente de computação, os tipos de instância selecionados para ele devem compartilhar a mesma arquitetura. Por exemplo, você não pode misturar instâncias ARM e x86 no mesmo ambiente de computação.

 Note

O AWS Batch dimensionará GPUs com base na quantidade necessária nas filas de trabalho. Para usar o agendamento de GPU, o ambiente computacional deve incluir tipos de instância das famílias p6, p3, p4, p5, g3, g3s, g4, g5 ou g6.

18. (Opcional) Expanda Configuração adicional.

- (Opcional) Em Grupo de posicionamento, insira um nome de grupo de posicionamento para agrupar recursos no ambiente de computação.
- Em Estratégia de alocação, escolha BEST_FIT_PROGRESSIVE.
- (Opcional) Para a Amazon Machine Images (AMIs) Configuration, escolha Adicionar configuração de imagens de máquina da Amazon (amis).

Você pode usar uma AMI Amazon Linux otimizada para Amazon EKS ou uma AMI personalizada.

- i. Para usar uma [AMI do Amazon Linux otimizada para Amazon EKS](#):
 - A. Em Tipo de imagem, selecione uma das opções a seguir.
 - [Amazon Linux 2](#): padrão para todas as famílias de instâncias baseadas em AWS Graviton (por exemplo, C6g, M6g, R6g e T4g) e pode ser usado para todos os tipos de instâncias não GPU.
 - [Amazon Linux 2 \(acelerado\)](#): padrão para todas as famílias de instâncias de GPU (por exemplo, P4 e G4) e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.
 - [Amazon Linux 2023](#): AWS Batch é compatível com o Amazon Linux 2023 (AL2023).
 - [Amazon Linux 2023 \(acelerado\)](#): famílias de instâncias de GPU e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.
 - B. Em Versão Kubernetes, insira um [número de versão Kubernetes](#).
- ii. Para usar uma AMI personalizada:
 - A. Em Tipo de imagem, escolha o tipo de AMI no qual a AMI personalizada se baseia:
 - [Amazon Linux 2](#): padrão para todas as famílias de instâncias baseadas em AWS Graviton (por exemplo, C6g, M6g, R6g e T4g) e pode ser usado para todos os tipos de instâncias não GPU.
 - [Amazon Linux 2 \(acelerado\)](#): padrão para todas as famílias de instâncias de GPU (por exemplo, P4 e G4) e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.
 - [Amazon Linux 2023](#): AWS Batch é compatível com AL2023.
 - [Amazon Linux 2023 \(acelerado\)](#): famílias de instâncias de GPU e pode ser usado para todos os tipos de instâncias não baseadas no AWS Graviton.
 - B. Para Substituição de ID da imagem, insira o ID da AMI personalizada.
 - C. Em Versão Kubernetes, insira um [número de versão Kubernetes](#).
- d. (Opcional) Em Modelo de execução, escolha um [modelo de execução](#) existente.
- e. (Opcional) Em Launch template version, insira **\$Default**, **\$Latest** ou um número de versão.

- f. (Opcional) Em Modelo de execução de substituição, para adicionar uma substituição, escolha Adicionar Modelo de execução de substituição:
 - i. (Opcional) Em Modelo de execução, escolha o modelo de execução ao qual adicionar a substituição.
 - ii. (Opcional) Em Versão do modelo de lançamento, escolha o número da versão do modelo de lançamento, `$Default` ou `$Latest`.
 - iii. (Opcional) Em tipos de instâncias de destino, escolha o tipo de instância ou a família à qual essa substituição deve ser aplicada. Isso pode ser direcionado somente aos tipos e famílias de instâncias que estão incluídos nos Tipos de instância permitidos.
 - iv. (Opcional) Em `userDataType`, escolha a inicialização do nó EKS. Use esse campo somente se você tiver uma AMI especificada no modelo de execução ou como substituição do modelo de execução. Escolha `EKS_NODEADM` para AMIs personalizadas com base em `EKS_AL2023` ou `EKS_AL2023_NVIDIA` ou `EKS_BOOSTRAP_SH` para `EKS_AL2` e `EKS_AL_NVIDIA`. O valor padrão é `EKS_BOOSTRAP_SH`.

Você usaria `userDataType` quando tiver um [ambiente misto](#) em que esteja usando AMIs personalizadas baseadas em AL2 e no AL2023 no mesmo ambiente computacional.

- 19. Escolha Próxima página.
- 20. Em Virtual Private Cloud (VPC) ID (Nuvem Privada Virtual Private Cloud), escolha uma VPC na qual executar as instâncias.
- 21. Em Sub-redes, selecione a sub-rede que será usada. Por padrão, todas as sub-redes dentro da VPC selecionadas estão disponíveis.

 Note

AWS Batch no Amazon EKS é compatível com Zonas Locais. Para obter mais informações, consulte [Amazon EKS eAWS, em Zonas Locais](#) no Guia do usuário EKS.

- 22. (Opcional) Para Security groups, escolha um grupo de segurança a ser anexado às suas instâncias. Por padrão, o grupo de segurança padrão para sua VPC é escolhido.
- 23. Escolha Próxima página.
- 24. Para Revisar, reveja as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar ambiente de computação.

Tutorial: Crie um ambiente computacional não gerenciado usando os recursos do Amazon EKS

Conclua as etapas a seguir para criar um ambiente computacional não gerenciado usando os recursos do Amazon Elastic Kubernetes Service (Amazon EKS).

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação na parte superior da página, selecione a opção Região da AWS a ser usada.
3. No painel de navegação, escolha Ambientes de computação.
4. Escolha Criar.
5. Configure o ambiente.
 - a. Para a configuração do ambiente de computação, escolha Amazon Elastic Kubernetes Service (Amazon EKS).
 - b. Em Tipo de orquestração, escolha Não gerenciado.
6. Para Nome, especifique um nome exclusivo para seu ambiente de computação. Os nomes podem ter até 128 caracteres. Podem conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
7. Para o cluster EKS, escolha um cluster existente do Amazon EKS. Para criar um novo cluster EKS, siga as etapas na [página Criar um cluster do Amazon EKS](#).
8. Em Namespace, insira um Kubernetes namespace para agrupar seus AWS Batch processos no cluster.
9. (Opcional) Para Máximo v CPUs, especifique o número máximo de v CPUs disponível para agendamento de trabalhos a partir de sua capacidade provisionada.
10. (Opcional) Expanda as Tags. Escolha Adicionar tag e, em seguida, insira um par chave-valor.
11. Escolha Próxima página.
12. Para Revisar, reveja as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar ambiente de computação.

 **Atribuição de nós de cluster do Amazon EKS a um ambiente computacional não gerenciado**
Depois de criar o ambiente computacional não gerenciado, você precisa rotular seus nós do Amazon EKS com o ambiente computacional UUID.

Primeiro, obtenha o UUID do ambiente computacional a partir do resultado da DescribeComputeEnvironments API:

```
$ aws batch describe-compute-environments \
  --compute-environments unmanagedEksCE \
  --query "computeEnvironments[].{name: computeEnvironmentName, uuid: uuid}"
```

Obtenha as informações do nó:

```
kubectl get nodes -o name
```

Identifique os nós com o UUID do ambiente AWS Batch computacional:

```
kubectl label <node-name> batch.amazonaws.com/compute-environment-uuid=uuid
```

Recurso: modelo de ambiente de computação

O exemplo a seguir mostra um Modelo do Ambiente de Computação vazio. É possível usar esse modelo para criar seu ambiente de computação que pode ser salvo em um arquivo e usado com a opção `--cli-input-json` AWS CLI. Para obter mais informações sobre esses parâmetros, consulte [CreateComputeEnvironment](#) no AWS Batch API Reference.

Note

Você pode gerar o modelo de ambiente de computação com o comando da AWS CLI a seguir.

```
$ aws batch create-compute-environment --generate-cli-skeleton
```

```
{
  "computeEnvironmentName": "",
  "type": "UNMANAGED",
  "state": "DISABLED",
  "unmanagedvCpus": 0,
  "computeResources": {
    "type": "EC2",
```

```
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 0,
    "desiredvCpus": 0,
    "instanceTypes": [
        ""
    ],
    "imageId": "",
    "subnets": [
        ""
    ],
    "securityGroupIds": [
        ""
    ],
    "ec2KeyPair": "",
    "instanceRole": "",
    "tags": {
        "KeyName": ""
    },
    "placementGroup": "",
    "bidPercentage": 0,
    "spotIamFleetRole": "",
    "launchTemplate": {
        "launchTemplateId": "",
        "launchTemplateName": "",
        "version": ""
    },
    "ec2Configuration": [
        {
            "imageType": "",
            "imageIdOverride": "",
            "imageKubernetesVersion": ""
        }
    ],
    "serviceRole": "",
    "tags": {
        "KeyName": ""
    },
    "eksConfiguration": {
        "eksClusterArn": "",
        "kubernetesNamespace": ""
    }
}
```

}

Tabela de computação do tipo de instância

A tabela a seguir lista Região da AWS, a palavra-chave da família de instâncias e as famílias de instâncias disponíveis. AWS Batch tentará alocar uma instância da família mais recente, mas como a disponibilidade da família de instâncias varia, Região da AWS pode obter uma geração anterior da família de instâncias.

default_x86_64

Região	Famílias de instâncias
Todas as Região da AWS compatíveis com AWS Batch	m6i, c6i, r6i c7i

default_arm64

Região	Famílias de instâncias
Todas as Região da AWS compatíveis com AWS Batch	m6g, c6g, r6g c7g

Ideal

Região	Famílias de instâncias
• ap-northeast-1 • ap-northeast-2 • ap-south-1 • ap-southeast-1 • ap-southeast-2 • ca-central-1 • cn-north-1 • cn-northwest-1	m4, c4, r4

Região	Famílias de instâncias
• eu-central-1 • eu-west-1 • eu-west-2 • sa-east-1 • us-east-1 • us-east-2 • us-gov-west-1 • us-west-1 • us-west-2	

Região	Famílias de instâncias
• af-south-1 • ap-east-1 • ap-northeast-3 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ca-west-1 • eu-central-2 • eu-north-1 • eu-south-1 • eu-south-2 • eu-west-3 • il-central-1 • me-central-1 • me-south-1 • us-gov-east-1 • us-isob-east-1 • us-iso-east-1 • us-isof-south-1 • us-isof-east-1 • eu-isoe-west-1 • us-northeast-1	m5, c5, r5
• ap-southeast-5 • ap-southeast-7 • ap-east-2 • mx-central-1	m6, c6, r6

Atualizar um ambiente computacional no AWS Batch

O AWS Batch fornece várias estratégias para atualizar ambientes computacionais, cada uma projetada para cenários e requisitos específicos de atualização. Essas abordagens usam a mesma API de atualização subjacente, mas representam métodos prescritivos diferentes para gerenciar atualizações de forma eficaz. Você pode gerenciar essas atualizações usando o console AWS Batch ou o AWS CLI. A compreensão dessas estratégias ajuda você a escolher o método mais adequado às suas necessidades e, ao mesmo tempo, minimizar a interrupção de seus workloads.

Este tópico dá uma visão geral das estratégias de atualização disponíveis e dá orientações sobre quando cada abordagem deve ser usada. Para obter procedimentos detalhados, consulte as seções individuais de cada estratégia de atualização.

Important

O AWS Batch cria e gerencia vários recursos AWS em seu nome e dentro da sua conta, incluindo Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Frotas Spot e Amazon ECS Clusters. Esses recursos gerenciados são configurados especificamente para garantir a operação ideal do AWS Batch. A modificação manual desses recursos gerenciados pela AWS Batch, a menos que seja explicitamente declarada na documentação AWS Batch, pode resultar em comportamento inesperado, incluindo ambientes computacionais `INVALID`, comportamento de escalabilidade de instâncias abaixo do ideal, atraso no processamento da workload ou custos inesperados. Essas modificações manuais não podem ser sustentadas de forma determinística pelo serviço AWS Batch. Sempre use as APIs AWS Batch compatíveis ou o console AWS Batch para gerenciar seus ambientes computacionais.

Tópicos

- [Estratégias de atualização do ambiente computacional](#)
- [Escolhendo a estratégia de atualização correta](#)
- [Executar atualizações de escalabilidade](#)
- [Realizar atualizações da infraestrutura](#)
- [Realizar atualizações em azul/verde para ambientes computacionais](#)

Estratégias de atualização do ambiente computacional

Quando você usa atualizações de escalabilidade ou infraestrutura, seu ambiente computacional é atualizado no local. Para a estratégia de atualização azul/verde, você está criando um novo ambiente computacional (verde) e, em seguida, migrando seu workload do ambiente computacional antigo (azul) para o novo ambiente computacional (verde).

AWS Batch fornece três estratégias diferentes para atualizações do ambiente computacional:

Atualizações de escalabilidade

As atualizações de escalabilidade ajustam a capacidade do seu ambiente computacional adicionando ou removendo instâncias sem substituir as existentes. Esse é o cenário de atualização mais rápido e não requer tempo de inatividade. Use atualizações de escalabilidade quando precisar alterar as configurações de capacidade (vCPUs). Essas atualizações geralmente são concluídas em minutos.

As atualizações Fargate são realizadas usando os mesmos procedimentos das atualizações de escalabilidade. Para obter mais informações, consulte [Executar atualizações de escalabilidade](#).

Atualizações da infraestrutura

As atualizações de infraestrutura substituem as instâncias em seu ambiente computacional por novas instâncias com configurações atualizadas. Essas atualizações exigem configurações específicas de perfil de serviço e estratégia de alocação, mas oferecem um tempo de inatividade mínimo, com a possibilidade de interrupção dos trabalhos em execução. Use atualizações de infraestrutura quando precisar modificar tipos de instância, configuração de AMI, configurações de rede, perfil de serviço, estado do ambiente ou outros componentes da infraestrutura. Essas atualizações geralmente são concluídas em 10 a 30 minutos, dependendo da conclusão do trabalho.

Para obter mais informações, consulte [Realizar atualizações da infraestrutura](#).

Atualizações azul/verdes

As atualizações azul/verde criam um novo ambiente computacional junto com o ambiente existente, permitindo uma transição gradual do workload sem tempo de inatividade. Essa abordagem fornece o caminho de atualização mais seguro, mas exige a execução temporária de dois ambientes. Use atualizações azul/verdes quando precisar de tempo de inatividade zero, quiser testar as alterações antes da implantação completa, precisar de capacidade de reversão

rápida ou estiver usando configurações não compatíveis com atualizações de infraestrutura. O tempo de conclusão é variável e controlado por você.

Para obter mais informações, consulte [Realizar atualizações em azul/verde para ambientes computacionais](#).

Escolhendo a estratégia de atualização correta

Use este guia de decisão para selecionar a estratégia de atualização mais adequada às suas necessidades:

Escolha atualizações de escalabilidade quando

Escolha a estratégia de atualização de escalabilidade quando precisar apenas ajustar a capacidade computacional (vCPUs). As atualizações de escalabilidade são ideais quando você precisa de atualizações rápidas, sem tempo de inatividade e sem necessidade de alterações na configuração da infraestrutura.

Para ver os procedimentos detalhados, consulte [Executar atualizações de escalabilidade](#).

Escolha atualizações de infraestrutura quando

Escolha a estratégia de atualização da infraestrutura quando precisar modificar tipos de instância, configurações de AMI, perfil de serviço, estado do ambiente ou configuração de rede. Seu ambiente deve usar o perfil vinculado ao serviço AWSServiceRoleForBatch e uma estratégia de alocação de BEST_FIT_PROGRESSIVE, SPOT_CAPACITY_OPTIMIZED ou SPOT_PRICE_CAPACITY_OPTIMIZED. As atualizações de infraestrutura funcionam bem quando alguma interrupção do trabalho é aceitável durante a atualização e você deseja atualizações automáticas para a AMI mais recente otimizada para Amazon ECS.

Para ver os procedimentos detalhados, consulte [Realizar atualizações da infraestrutura](#).

Escolha atualizações azuis/verdes quando

Escolha a estratégia de atualização azul/verde quando for necessário tempo de inatividade zero para seus workloads ou quando você precisar testar as alterações antes de fazer a transição dos workloads de produção. Essa abordagem é essencial quando a capacidade de reversão rápida é importante, seu ambiente usa a estratégia de alocação BEST_FIT ou seu ambiente não usa o perfil

vinculado ao serviço `AWSServiceRoleForBatch`. As atualizações azul/verde também são a melhor opção quando você usa AMIs personalizadas que exigem atualizações manuais ou precisam fazer grandes alterações na configuração.

Para ver os procedimentos detalhados, consulte [Realizar atualizações em azul/verde para ambientes computacionais](#).

Considerações sobre a atualização AMI

AWS Batch pode atualizar para a AMI otimizada para Amazon ECS mais recente durante as atualizações de [infraestrutura](#) quando todas essas condições forem atendidas:

Note

Após a conclusão da atualização da infraestrutura, `updateToLatestImageVersion` é definido como `false`. Para iniciar outra atualização, `updateToLatestImageVersion` deve ser definido como `true`.

- O ambiente computacional usa o perfil vinculado ao serviço `AWSServiceRoleForBatch`
- A estratégia de alocação está definida como `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` ou `SPOT_PRICE_CAPACITY_OPTIMIZED`
- Nenhum ID AMI está explicitamente especificado em `imageId`, `imageIdOverride` ou no modelo de lançamento
- `updateToLatestImageVersion` é definido como `true`

Atualizações AMI usando implantações azul/verde

Você deve usar a implantação azul/verde para atualizar as AMIs nesses cenários:

- Ao usar uma versão específica da AMI otimizada para Amazon ECS
- Quando o ID AMI é especificado em qualquer um dos seguintes:
 - Modelo de execução (é necessário atualizar o modelo ou removê-lo)
 - O parâmetro `imageId`
 - O parâmetro `imageIdOverride` na configuração EC2
- Ao usar a estratégia de alocação `BEST_FIT` (não suporta atualizações de infraestrutura)

- Quando não estiver usando o [perfil vinculado ao serviço](#) `AWSServiceRoleForBatch`.

Executar atualizações de escalabilidade

As atualizações de escalabilidade ajustam a capacidade do seu ambiente computacional adicionando ou removendo instâncias. Essa é a estratégia de atualização mais rápida e não exige a substituição das instâncias existentes. As atualizações de escalabilidade funcionam com qualquer tipo de perfil de serviço e estratégia de alocação, tornando-as a opção de atualização mais flexível.

Alterações que acionam uma atualização de escalabilidade

Quando você modifica somente as configurações a seguir, AWS Batch executa uma atualização de escalabilidade. Se você modificar qualquer uma dessas configurações junto com outras configurações do ambiente computacional, AWS Batch executará uma [atualização de infraestrutura](#).

As configurações a seguir acionam atualizações de escalabilidade quando modificadas exclusivamente:

- `desiredvCpus` — Define o número alvo de vCPUs para o ambiente.
- `maxvCpus` — Define o número máximo de vCPUs que é possível iniciar.
- `minvCpus` — Especifica o número mínimo de vCPUs a serem mantidas.

Para ambientes de computação Fargate, você também pode modificar essas configurações para atualizações de escalabilidade:

- `securityGroupIds` — IDs de grupos de segurança para o ambiente computacional.
- `subnets` — Sub-redes para o ambiente de computação.

Note

Recomendamos não usar `desiredvCpus` para iniciar uma atualização de escalabilidade, pois o AWS Batch se ajustará dinamicamente ao `desiredvCpus`. Em vez disso, você deve atualizar o `minvCpus`.

Ao atualizar o `desiredvCpus`, o valor deve estar entre `minvCpus` e `maxvCpus`. O novo valor deve ser maior ou igual ao valor `desiredvCpus` atual. Para obter mais informações, consulte [the section called “Mensagem de erro ao atualizar a configuração `desiredvCpus`”](#).

 Important

Se você modificar qualquer uma dessas configurações de escalabilidade junto com outras configurações do ambiente computacional (como tipos de instância, IDs de AMI ou modelos de execução), o AWS Batch executará uma atualização de infraestrutura em vez de uma atualização de escalabilidade. As atualizações da infraestrutura demoram mais e podem substituir as instâncias existentes.

Performing scaling updates using the Console de gerenciamento da AWS

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Ambientes e, então, a aba Ambientes computacionais.
3. Selecione o ambiente computacional a atualizar.
4. Escolha Ações e então Editar.
5. Modifique uma ou mais das [configurações que oferecem suporte a atualizações de escalabilidade](#). Por exemplo:
 - Em vCPUs mínimos, insira o número mínimo de vCPUs.
 - Em vCPUs desejados, insira o número desejado de vCPUs.
 - Em Número máximo de vCPUs, insira o número máximo de vCPUs.
6. Escolha Salvar alterações.
7. Monitore o status do ambiente de computação. A atualização deve ser concluída rapidamente, pois envolve apenas operações de escalonamento.

Performing scaling updates using the AWS CLI

Use o comando `update-compute-environment` para realizar atualizações de escalabilidade. Os dois exemplos a seguir demonstram operações comuns de escalabilidade. Você pode modificar uma ou mais das seguintes [configurações que oferecem suporte a atualizações de escalabilidade](#)

- Este exemplo atualiza vCPUs desejados, mínimos e máximos:

```
aws batch update-compute-environment \
  --compute-environment your-compute-environment-name \
  --compute-resources minvCpus=2,maxvCpus=8
```

Monitorar atualizações de escalabilidade

Monitore suas atualizações de escalabilidade usando o console AWS Batch para ver o status do ambiente computacional e verificar a contagem de instâncias e as métricas da vCPU. Você também pode usar o AWS CLI com o comando `describe-compute-environments` para verificar o status e monitorar contagens de instâncias e valores de vCPU.

Realizar atualizações da infraestrutura

As atualizações de infraestrutura substituem as instâncias em seu ambiente computacional por novas instâncias com configurações atualizadas. Essa estratégia de atualização leva mais tempo do que escalar as atualizações e exige configurações específicas do perfil de serviço e da estratégia de alocação. As atualizações de infraestrutura fornecem uma maneira de modificar as configurações fundamentais do ambiente computacional e, ao mesmo tempo, manter a disponibilidade do serviço.

Important

As atualizações de infraestrutura exigem o perfil vinculado ao serviço `AWSServiceRoleForBatch` e uma estratégia de alocação de `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` ou `SPOT_PRICE_CAPACITY_OPTIMIZED`. Se seu ambiente não atender a esses requisitos, use atualizações azul/verde.

Mudanças que acionam atualizações de infraestrutura

Quando você modifica qualquer uma das configurações a seguir, AWS Batch realiza uma atualização de infraestrutura. As atualizações de infraestrutura também ocorrem quando você modifica essas configurações junto com as configurações de atualização de escala.

As configurações a seguir acionam atualizações de infraestrutura:

Configuração de computação

- `allocationStrategy` — Determina como AWS Batch seleciona os tipos de instância.
- `instanceTypes` — Especifica quais tipos de instância do EC2 usar.
- `bidPercentage` — Porcentagem máxima do preço sob demanda para instâncias spot.
- `type` — Tipo de ambiente computacional (EC2 ou SPOT).

Configuração da AMI e lançamento

- `imageId` — AMI específica para usar em instâncias.
- `ec2Configuration` — Configuração do EC2, incluindo `imageIdOverride`.
- `launchTemplate` — Configurações do modelo de execução do EC2.
- `ec2KeyPair` — Par de chaves SSH para acesso a instância.
- `updateToLatestImageVersion` — Configuração de atualizações automáticas da AMI.

Redes e segurança

- `subnets` — sub-redes VPC em que as instâncias são lançadas (para ambientes computacionais EC2).
- `securityGroupIds` — Grupos de segurança para instâncias (para ambientes computacionais EC2).
- `placementGroup` — Configuração do grupo de posicionamento do EC2.

Outras configurações

- `instanceRole` — Perfil do IAM para instâncias do EC2.
- `tags` — Tags aplicadas a instâncias do EC2.

Important

Se você modificar qualquer configuração de atualização de infraestrutura junto com as configurações de atualização de escalabilidade (como `desiredvCpus`, `maxvCpus` ou `minvCpus`), AWS Batch executará uma atualização de infraestrutura. Atualizações de infraestrutura levam mais tempo do que atualizações de escalabilidade.

Seleção da AMI durante atualizações de infraestrutura

Durante uma atualização de infraestrutura, o ID da AMI do ambiente de computação pode mudar, dependendo se as AMIs estão especificadas em qualquer uma dessas três configurações. As AMIs são especificadas em `imageId` (em `computeResources`), `imageIdOverride` (em `ec2Configuration`) ou no modelo de lançamento especificado em `launchTemplate`. Suponha

que nenhuma ID de AMI seja especificada em nenhuma dessas configurações e a configuração `updateToLatestImageVersion` seja `true`. Em seguida, a AMI mais recente otimizada para o Amazon ECS e compatível com AWS Batch é usada para qualquer atualização da infraestrutura.

Se uma ID de AMI for especificada em pelo menos uma dessas configurações, a atualização dependerá de qual configuração forneceu a ID de AMI usada antes da atualização. Quando você cria um ambiente de computação, a prioridade para selecionar uma ID de AMI é primeiro o modelo de execução, depois a configuração `imageId` e, finalmente, a configuração `imageIdOverride`. No entanto, se a ID da AMI usado for do modelo de execução, a atualização das configurações `imageIdOverride` ou `imageId` não a atualizará. A única maneira de atualizar uma ID de AMI selecionada no modelo de execução é atualizando o modelo de execução. Se o parâmetro de versão do modelo de execução for `$Default` ou `$Latest`, a versão padrão ou mais recente do modelo de execução especificado será avaliada. Se uma ID de AMI diferente for selecionada por padrão ou se a versão mais recente do modelo de execução for selecionada, essa ID de AMI será usada na atualização.

Se o modelo de execução não foi usado para selecionar a ID da AMI, a ID da AMI especificada nos parâmetros `imageId` ou `imageIdOverride` será usada. Se ambos forem especificados, a ID da AMI especificada no parâmetro `imageIdOverride` será usada.

Suponha que o ambiente de computação use uma ID de AMI especificada pelos parâmetros `imageId`, `imageIdOverride`, ou `launchTemplate` e você queira usar a AMI otimizada mais recente do Amazon ECS compatível com AWS Batch. Em seguida, a atualização deve remover as configurações que forneceram os IDs de AMI. Para `imageId`, isso requer a especificação de uma string vazia para esse parâmetro. Para `imageIdOverride`, isso requer a especificação de uma string vazia para o parâmetro `ec2Configuration`.

Se a ID da AMI veio do modelo de lançamento, você pode mudar para a AMI otimizada mais recente do Amazon ECS que é suportada AWS Batch por uma das seguintes formas:

- Remova o modelo de execução especificando uma string vazia para o parâmetro `launchTemplateId` ou `launchTemplateName`. Isso remove todo o modelo de lançamento, em vez de apenas a ID da AMI.
- Se a versão atualizada do modelo de lançamento não especificar uma ID de AMI, o parâmetro `updateToLatestImageVersion` deverá ser definido como `true`.

Manuseio de trabalhos durante atualizações

Configure como os trabalhos em execução são tratados durante uma atualização de infraestrutura usando a política de atualização. Quando você define `terminateJobsOnUpdate=true`, os trabalhos em execução são encerrados imediatamente, a configuração `jobExecutionTimeoutMinutes` é ignorada e a atualização prossegue assim que as instâncias podem ser substituídas. Quando você define `terminateJobsOnUpdate=false`, os trabalhos em execução continuam pelo período de tempo limite especificado com um tempo limite padrão de 30 minutos, e os trabalhos são encerrados se excederem o tempo limite.

Note

Para repetir trabalhos que são encerrados durante uma atualização, configure uma estratégia de repetição do trabalho. Para obter mais informações, consulte [the section called “Repetições de trabalho automatizadas”](#).

Performing infrastructure updates using the Console de gerenciamento da AWS

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Ambientes e, em seguida, Ambientes computacionais.
3. Selecione o ambiente computacional a atualizar.
4. Escolha Ações e então Editar.
5. Na seção Atualizar comportamento, configure como os trabalhos em execução são tratados:
 - Escolha Atualizar AMI para a versão mais recente para atualizar a AMI para a versão mais recente.
 - Escolha Encerrar trabalhos imediatamente na atualização para encerrar trabalhos quando o processo de atualização for executado.
 - Em Tempo limite de execução do trabalho, insira o número de minutos de espera antes de iniciar o processo de atualização.
6. Modifique uma ou mais das [configurações que exigem atualizações de infraestrutura](#). Por exemplo:
 - Perfil da instância
 - Usar instâncias spot EC2

- Tipos de instância permitidos
 - Grupo de posicionamento
 - EC2 key pair
 - Configuração do EC
 - Modelos de inicialização
 - Sub-redes
 - Grupos de segurança
7. Escolha Salvar alterações.
 8. Monitore o status do ambiente de computação. O ambiente exibirá UPDATING durante o processo de atualização.

Performing infrastructure updates using the AWS CLI

Use o comando `update-compute-environment` com uma alteração em uma ou mais das [configurações que exigem atualizações de infraestrutura](#). Os três exemplos a seguir são operações comuns de infraestrutura.

- Este exemplo atualiza os tipos de instância e configura a política de atualização:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources instanceTypes=default_x86_64 \  
  --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=30
```

- Este exemplo atualiza as sub-redes e os grupos de segurança da VPC:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources subnets=subnet-abcd1234,subnet-efgh5678 \  
  securityGroupIds=sg-abcd1234 \  
  --update-policy terminateJobsOnUpdate=true
```

- Este exemplo permite atualizações automáticas da AMI mais recente, otimizada para o Amazon ECS:

```
aws batch update-compute-environment \  
  --compute-environment your-compute-environment-name \  
  --compute-resources updateToLatestImageVersion=true \  
  --update-policy terminateJobsOnUpdate=true
```

```
--update-policy terminateJobsOnUpdate=false, jobExecutionTimeoutMinutes=60
```

Monitorando as atualizações da infraestrutura

Monitore suas atualizações de infraestrutura usando o console AWS Batch para observar a alteração do status do ambiente computacional para UPDATING, monitorar o progresso da substituição da instância e verificar se há falhas nas atualizações. A atualização é bem-sucedida quando o estado do ambiente computacional é VAILED. Você também pode usar o CloudWatch para rastrear eventos de encerramento de instâncias e monitorar os estados do trabalho durante a atualização. Com o AWS CLI, use o comando `describe-compute-environments` para verificar o status e monitorar os eventos do ciclo de vida da instância.

Realizar atualizações em azul/verde para ambientes computacionais

Uma atualização azul/verde é uma estratégia de atualização que reduz o tempo de inatividade e o risco criando um novo ambiente computacional (verde) junto com seu ambiente computacional existente (azul). Essa abordagem permite que você faça a transição gradual dos workloads para o novo ambiente, mantendo o ambiente existente operacional. As atualizações azul/verde fornecem o caminho de atualização mais seguro e funcionam com qualquer tipo de perfil de serviço ou estratégia de alocação.

Visão geral

As atualizações azul/verde oferecem várias vantagens que as tornam ideais para ambientes de produção. Elas fornecem zero tempo de inatividade, mantendo seus workloads em execução contínua durante o processo de atualização. A abordagem permite recursos como reversão fácil, permitindo que você volte rapidamente para o ambiente original se surgirem problemas. Você pode implementar uma estratégia de transição gradual, verificando o desempenho do novo ambiente antes de mudar totalmente seus workloads de produção. Esse método também oferece excelente mitigação de riscos, pois o ambiente original permanece inalterado e operacional até que você decida removê-lo.

Quando atualizações azul/verde são necessárias

Você deve usar atualizações azul/verde nas seguintes situações:

- Quando seu ambiente computacional usa a estratégia de alocação BEST_FIT (não suporta atualizações de infraestrutura)

- Quando seu ambiente computacional não usa o perfil vinculado ao serviço `AWSServiceRoleForBatch`
- Quando você precisa fazer a transição entre diferentes tipos de perfil de serviço

Quando atualizações azul/verde são recomendadas

As atualizações em azul e verde são especialmente recomendadas para ambientes de produção em que o tempo de inatividade zero é essencial para seus workloads. Essa abordagem funciona bem quando você precisa testar novas configurações antes de fazer a transição dos workloads de produção, garantindo que as alterações atendam aos seus requisitos de desempenho e confiabilidade. Escolha atualizações azul/verde quando a capacidade de reversão rápida for importante para suas operações, especialmente se você estiver atualizando AMIs personalizadas com alterações significativas. Esse método também é ideal quando você deseja validar as características de desempenho e o comportamento antes de se comprometer totalmente com as mudanças, proporcionando confiança em seu processo de atualização.

Pré-requisitos

Antes de realizar uma atualização azul/verde, verifique se você tem:

- [Permissões do IAM](#) apropriadas para criar e gerenciar ambientes computacionais
- Acesso para visualizar e modificar as configurações da fila de trabalhos
- Estratégias de repetição de trabalhos configurados para suas definições de trabalhos para lidar com possíveis falhas durante a transição. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).
- O AMI ID para o novo ambiente computacional. Ele pode ser:
 - Uma versão recente e aprovada da AMI otimizada do Amazon ECS (usada por padrão)
 - Uma AMI personalizada que atende à especificação da AMI da instância de contêiner do Amazon ECS. Ao usar uma AMI personalizada, você pode especificá-la em uma das formas a seguir:
 - Usando o campo de Substituição de ID de imagem na configuração do EC2
 - Especificando em um modelo de inicialização

Para obter mais informações sobre como criar AMIs personalizadas, consulte [Tutorial: criar uma AMI de recurso de computação](#).

Antes de criar o novo ambiente, você precisa registrar a configuração do seu ambiente computacional existente. Você pode fazer isso usando o Console de gerenciamento da AWS ou a AWS CLI.

 Note

Os procedimentos a seguir detalham como realizar uma atualização azul/verde que altera somente a AMI. Você pode atualizar outras configurações para o novo ambiente.

 Important

Quando você remove o ambiente computacional antigo (azul), qualquer trabalho atualmente em execução nessas instâncias falhará porque as instâncias serão encerradas. Configure estratégias de repetição de trabalhos em suas definições de trabalhos para lidar com essas falhas automaticamente. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).

Quando você estiver confiante no novo ambiente:

1. Remova o antigo ambiente computacional da fila de trabalhos.
2. Aguarde a conclusão de qualquer trabalho em execução no ambiente antigo.
3. Exclua o antigo ambiente de computação.

Performing blue/green updates using the Console de gerenciamento da AWS

1. Clone seu ambiente computacional atual
 - a. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
 - b. Selecione seu ambiente computacional existente.
 - c. Escolha Ações e então Clonar.
 - d. Para Nome, insira um nome exclusivo para seu novo ambiente computacional.
 - e. Escolha Próximo.
 - f. Na seção Configuração da instância, defina estes valores:
 - i. Expanda Configuração Adicional.

- ii. Para a Configuração EC2, especifique o novo tipo de AMI em Tipo de imagem e ID de AMI no campo Substituição de ID de imagem.
 - g. Escolha Próximo.
 - h. Para Configuração de rede, escolha Avançar.
 - i. Revise as outras configurações que são copiadas automaticamente do seu ambiente existente.
 - j. Escolha Criar ambiente computacional.
 - k. Aguarde até que o novo status do ambiente computacional se torne VALID.
2. Alterar a ordem da fila de trabalhos
- a. No painel de navegação, escolha Filas de Tarefas.
 - b. Selecione a fila de trabalhos associada ao seu ambiente computacional existente.
 - c. Escolha Editar.
 - d. Em Ambiente computacional conectado, adicione o novo ambiente computacional:
 - Adicione o novo ambiente computacional com um número de pedido maior do que o ambiente existente para fazer a transição do workload.
 - Depois de verificar se o novo ambiente está funcionando corretamente, você pode torná-lo o ambiente principal fornecendo a ele um número de pedido menor.
 - e. Escolha Atualizar fila de trabalhos.
3. Limpeza
- a. Monitore a execução do trabalho no novo ambiente para garantir que tudo esteja funcionando conforme o esperado.
 - b. Quando você estiver confiante no novo ambiente:
 1. Remova o antigo ambiente computacional da fila de trabalhos.
 2. Aguarde a conclusão de qualquer trabalho em execução no ambiente antigo.
 3. Exclua o antigo ambiente de computação.

Performing blue/green updates using the AWS CLI

1. Para obter a configuração usando o AWS CLI, use o seguinte comando:

```
--compute-environments your-compute-environment-name
```

Salve a saída para referência ao criar o novo ambiente.

2. Crie um novo ambiente computacional usando a configuração do seu ambiente atual, mas com a nova AMI. Veja um exemplo de estrutura de comando:

Substitua os valores de exemplo pela configuração real da etapa anterior:

```
cat <<EOF > ./blue-green-compute-environment.json
{
  "computeEnvironmentName": "your-new-compute-environment-name",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "instanceRole": "arn:aws:iam::012345678901:instance-profile/
ecsInstanceRole",
    "type": "EC2",
    "minvCpus": 2,
    "desiredvCpus": 2,
    "maxvCpus": 256,
    "instanceTypes": [
      "optimal"
    ],
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "ec2Configuration": [
      {
        "imageType": "ECS_AL2023",
        "imageIdOverride": "ami-0abcdef1234567890"
      }
    ],
    "subnets": [
      "subnet-0abcdef1234567890"
    ],
    "securityGroupIds": [
      "sg-0abcdef1234567890"
    ]
  }
}
EOF
```



```
$ aws batch create-compute-environment --cli-input-json file:///./blue-green-compute-environment.json
```

3. Aguarde até que o novo ambiente fique disponível:

```
aws batch describe-compute-environments \
  --compute-environments your-new-compute-environment-name \
  --query 'computeEnvironments[].status'
```

4. Adicione o ambiente computacional a sua fila de trabalhos:

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-existing-environment \
  order=2,computeEnvironment=your-new-compute-environment-name
```

5. Depois de verificado, atualize novamente para tornar o novo ambiente primário:

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-new-compute-environment-name
```

Depois que todos os trabalhos forem concluídos no ambiente antigo, desative e exclua o ambiente antigo:

```
aws batch update-compute-environment \
  --compute-environment your-existing-environment \
  --state DISABLED
```

```
aws batch delete-compute-environment \
  --compute-environment your-existing-environment
```

Recursos de computação de AMIs

Por padrão, os ambientes de computação gerenciados do AWS Batch usam uma versão recente e aprovada da AMI otimizada do Amazon ECS para recursos de computação. No entanto, você pode

querer criar sua própria AMI a ser usada para seus ambientes de computação gerenciados. Se você precisar de alguma das opções a seguir, recomendamos que crie sua própria AMI:

- Aumentar o tamanho do armazenamento dos volumes de dados ou raiz da AMI
- Adicionar volumes de armazenamento de instância com suporte para tipos de instância do Amazon EC2
- Inspecionar o agente de contêiner do Amazon ECS
- Personalizando o Docker
- Configurar uma AMI de workload de GPU que permite que os contêineres acessem o hardware de GPU nos tipos de instância do Amazon EC2 compatíveis

Note

Depois que um ambiente de computação é criado, AWS Batch não faz upgrade das AMIs no ambiente de computação. AWS Batch também não atualiza as AMIs em seu ambiente de computação quando uma versão mais recente da AMI otimizada para Amazon ECS está disponível. Você é responsável pelo gerenciamento do sistema operacional convidado. Isso inclui quaisquer atualizações e patches de segurança. Você também é responsável por quaisquer outros utilitários ou aplicativos de software que instalar nos recursos de computação. Para usar uma nova AMI para seus trabalhos AWS Batch, faça o seguinte:

1. Crie um novo ambiente de computação com a nova AMI.
2. Adicione o ambiente de computação a uma fila de trabalhos existente.
3. Remova o antigo ambiente de computação da fila de trabalhos.
4. Exclua o ambiente de computação anterior.

Em abril de 2022, o AWS Batch adicionou suporte aprimorado para a atualização de ambientes de computação. Para obter mais informações, consulte [Atualizar um ambiente computacional no AWS Batch](#). Para utilizar a atualização aprimorada de ambientes de computação para atualizar AMIs, siga estas regras:

- Não defina o parâmetro de perfil de serviço ([serviceRole](#)) ou defina-o como a função vinculada ao serviço `AWSServiceRoleForBatch`.

- Defina o parâmetro da estratégia de alocação ([allocationStrategy](#)) como `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` ou `SPOT_PRICE_CAPACITY_OPTIMIZED`.
- Defina o parâmetro de atualização para a versão mais recente da imagem ([updateToLatestImageVersion](#)) como `true`.
- Não especifique uma ID de AMI em [imageId](#), [imageIdOverride](#) (em [ec2Configuration](#)) ou no modelo de execução ([launchTemplate](#)). Quando você não especifica uma ID da AMI, o AWS Batch seleciona a AMI mais recente otimizada para o Amazon ECS e AWS Batch compatível com o no momento em que a atualização da infraestrutura for iniciada. Como alternativa, você pode especificar a ID da AMI nos parâmetros `imageId` ou `imageIdOverride`. Ou pode especificar o modelo de execução que é identificado pelas propriedades `LaunchTemplate`. A alteração de qualquer uma dessas propriedades inicia uma atualização da infraestrutura. Se a ID da AMI for especificada no modelo de lançamento, a ID da AMI não poderá ser substituída pela especificação de uma ID da AMI nos parâmetros `imageId` ou `imageIdOverride`. A ID da AMI só pode ser substituída pela especificação de um modelo de lançamento diferente. Se a versão do modelo de lançamento estiver definida como `$Default` ou `$Latest`, a ID da AMI poderá ser substituída definindo uma nova versão padrão para o modelo de lançamento (se `$Default`) ou adicionando uma nova versão ao modelo de lançamento (se `$Latest`).

Se essas regras forem seguidas, qualquer atualização que acione uma atualização de infraestrutura fará com que a ID da AMI seja novamente selecionada. Se a configuração da [version](#) no modelo de execução ([launchTemplate](#)) for definida como `$Latest` ou `$Default`, a versão mais recente ou padrão do modelo de inicialização será avaliada no momento da atualização da infraestrutura, mesmo que o [launchTemplate](#) não tenha sido atualizado.

Tópicos

- [Especificação da AMI do recurso de computação](#)
- [Tutorial: criar uma AMI de recurso de computação](#)
- [Usar uma AMI de workload de GPU](#)
- [Depreciação do Amazon Linux](#)

- [Descontinuação da AMI Amazon EKS Amazon Linux 2](#)
- [Descontinuação da AMI Amazon ECS Amazon Linux 2](#)

Especificação da AMI do recurso de computação

A especificação da AMI dos recursos básicos AWS Batch de computação consiste no seguinte:

Obrigatória

- A distribuição Linux moderna executando pelo menos a versão 3.10 do Linux kernel em uma AMI de tipo de virtualização HVM. Não há suporte para contêineres do Windows.

Important

Tarefas paralelas de vários nós só podem ser executadas em recursos de computação que foram iniciados em uma instância do Amazon Linux com o pacote `ecs-init` instalado. Recomendamos que você use a AMI otimizada para Amazon ECS ao criar o ambiente de computação. Você pode fazer isso sem especificar uma AMI personalizada. Para obter mais informações, consulte [Trabalhos em paralelo de vários nós](#).

- O agente de contêiner do Amazon ECS. É recomendável usar a versão mais recente. Para obter mais informações, consulte [Instalar o agente de contêiner do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- O driver de log `awslogs` deve ser especificado como um driver de log disponível com a variável de ambiente `ECS_AVAILABLE_LOGGING_DRIVERS` quando o agente de contêiner do Amazon ECS é iniciado. Para obter mais informações, consulte [Configuração do Agente de Contêineres do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- Um daemon do Docker em execução, pelo menos, na versão 1.9 e quaisquer dependências de runtime do Docker. Para mais informações, consulte [Verificar dependências do runtime](#) na documentação do Docker.

Note

Recomendamos a versão do Docker que é fornecida e testada com a versão de agente de contêiner do Amazon ECS que você está usando. O Amazon ECS fornece um log de

alterações para a variante do Linux da AMI otimizada para Amazon ECS no GitHub. Para obter mais informações, consulte [Log de alterações](#).

Recomendado

- Um processo de inicialização e de nanny para executar e monitorar o agente do Amazon ECS. A AMI otimizada para Amazon ECS usa o processo de inicialização `ecs-init` e outros sistemas operacionais podem usar `systemd`. Para obter mais informações e exemplos, consulte [Example container instance User Data Configuration Scripts](#) no Guia do desenvolvedor do Amazon Elastic Container Service. Para obter mais informações sobre `ecs-init`, consulte o [projeto ecs-init](#) no GitHub. No mínimo, os ambientes de computação gerenciados exigem que o agente do Amazon ECS seja iniciado na inicialização. Se o agente do Amazon ECS não estiver sendo executado em seu recurso de computação, ele não poderá aceitar trabalhos do AWS Batch.

As AMIs otimizadas para Amazon ECS são pré-configuradas com esses requisitos e com essas recomendações. Recomendamos que você use a AMI otimizada para Amazon ECS ou uma AMI do Amazon Linux com o pacote `ecs-init` que está instalado para seus recursos de computação. Escolha outra AMI se o aplicativo precisar de um sistema operacional específico ou uma versão do Docker que ainda não está disponível nessas AMIs. Para obter mais informações, consulte [Amazon ECS-Optimized AMI](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Tutorial: criar uma AMI de recurso de computação

É possível criar uma AMI personalizada de recursos de computação para usar em seus ambientes de computação gerenciados e não gerenciados. Para obter instruções, consulte o [Especificação da AMI do recurso de computação](#). Então, depois de criar uma AMI personalizada, você pode criar um ambiente de computação que usa essa AMI, a que você pode associar uma fila de trabalhos. Por fim, comece a enviar trabalhos para essa fila.

Para criar uma AMI de recursos de computação personalizada

1. Escolha uma AMI base para começar. A AMI básica deve usar virtualização de HVM. A AMI básica não pode ser uma AMI do Windows.

 Note

A AMI que você escolher para um ambiente de computação deve corresponder à arquitetura dos tipos de instância que você pretende usar para este ambiente. Por exemplo, se o ambiente de computação usar tipos de instância A1, a AMI de recursos de computação escolhida deverá oferecer suporte a instâncias ARM. O Amazon ECS vende as versões x86 e ARM da AMI do Amazon Linux 2 otimizada para Amazon ECS. Para obter mais informações, consulte [AMI do Amazon Linux 2 otimizada para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

A AMI do Amazon Linux 2 otimizada para Amazon ECS é a AMI padrão para recursos de computação em ambientes de computação gerenciados. O Amazon Linux 2 AMI otimizado para Amazon ECS é pré-configurado e testado AWS Batch por AWS engenheiros. É uma AMI mínima com a qual você pode começar a usar e fazer com que seus recursos computacionais sejam executados AWS rapidamente. Para obter mais informações, consulte [AMI otimizada para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Como alternativa, você pode escolher outra variante do Amazon Linux 2 e instalar o pacote `ecs-init` com os seguintes comandos. Para obter mais informações, consulte [Instalação do agente de contêiner do Amazon ECS em uma EC2 instância do Amazon Linux 2](#) no Guia do desenvolvedor do Amazon Elastic Container Service:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Por exemplo, se você quiser executar cargas de trabalho de GPU em seus recursos AWS Batch computacionais, você pode começar com a [Amazon Linux Deep Learning](#) AMI. Em seguida, configure a AMI para executar AWS Batch trabalhos. Para obter mais informações, consulte [Usar uma AMI de workload de GPU](#).

 Important

Você pode escolher uma AMI básica que não seja compatível com o pacote `ecs-init`. No entanto, se fizer isso, você deverá configurar uma forma de iniciar o atendente do Amazon ECS na inicialização e mantê-lo em execução. Você também pode ver vários exemplos de scripts de configuração de dados do usuário que usam `systemd` para

iniciar e monitorar o agente de contêiner do Amazon ECS. Para obter mais informações, consulte [Scripts de configuração de dados de usuário de instância de contêiner de exemplo](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

2. Execute uma instância de sua AMI base selecionada com as opções de armazenamento adequadas para sua AMI. Você pode configurar o tamanho e o número de volumes do Amazon EBS conectados ou volumes de armazenamento de instância se o tipo de instância selecionado for compatível com eles. Para obter mais informações, consulte [Lançamento de uma instância](#) e [Amazon EC2 Instance Store](#) no Guia EC2 do usuário da Amazon.
3. Conecte-se à sua instância com o SSH e execute todas as tarefas de configuração necessárias. Isso pode incluir qualquer uma das ou todas as seguintes etapas:
 - Como instalar o agente de contêiner do Amazon ECS. Para obter mais informações, consulte [Instalar o agente de contêiner do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
 - Configuração de um script para formatar volumes de armazenamento de instâncias.
 - Adição de volume de armazenamento de instância ou sistemas de arquivos Amazon EFS para o arquivo `/etc/fstab` para que eles sejam montados na inicialização.
 - Configuração de opções do Docker, como habilitar a depuração ou ajustar o tamanho da imagem base.
 - Instalação de pacotes ou cópia de arquivos.

Para obter mais informações, consulte [Conectando-se à sua instância Linux usando SSH](#) no Guia do EC2 usuário da Amazon.

4. Se você iniciou o agente de contêiner do Amazon ECS em sua instância, deve interrompê-lo e remover todos os arquivos persistentes do ponto de verificação de dados antes de criar sua AMI. Caso contrário, se você não fizer isso, o atendente não iniciará nas instâncias que são executadas a partir da sua AMI.
 - a. Interrompa o agente de contêiner do Amazon ECS.
 - AMI do Amazon Linux 2 otimizada para Amazon ECS:

```
sudo systemctl stop ecs
```

- AMI do Amazon Linux otimizada para Amazon ECS:

```
sudo stop ecs
```

- b. Remova os arquivos de ponto de verificação de dados persistentes. Por padrão, esses arquivos estão localizados no diretório `/var/lib/ecs/data/`. Use o comando a seguir para remover esses arquivos, se houver algum.

```
sudo rm -rf /var/lib/ecs/data/*
```

5. Crie uma nova AMI da sua instância em execução. Para obter mais informações, consulte [Criação de uma AMI Linux suportada pelo Amazon EBS](#) no guia EC2 do usuário da Amazon.

Para usar sua nova AMI com AWS Batch

1. Após a criação da nova AMI, crie um ambiente de computação com a nova AMI. Para fazer isso, escolha o tipo de imagem e insira a ID de AMI personalizada na caixa de substituição da ID de imagem ao criar o ambiente AWS Batch computacional. Para obter mais informações, consulte [the section called “Tutorial: criar um ambiente de computação gerenciado usando recursos do Amazon EC2”](#).

 Note

A AMI que você escolher para um ambiente de computação deve corresponder à arquitetura dos tipos de instância que você pretende usar para este ambiente. Por exemplo, se o ambiente de computação usar tipos de instância A1, a AMI de recursos de computação escolhida deverá oferecer suporte a instâncias ARM. O Amazon ECS vende as versões x86 e ARM da AMI do Amazon Linux 2 otimizada para Amazon ECS. Para obter mais informações, consulte [AMI do Amazon Linux 2 otimizada para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

2. Crie uma fila de trabalhos e associe seu novo ambiente de computação. Para obter mais informações, consulte [Crie uma fila de trabalhos](#).

 Note

Todos os ambientes de computação associados a uma fila de trabalhos devem compartilhar a mesma arquitetura. AWS Batch não oferece suporte à mistura de tipos de arquitetura de ambiente de computação em uma única fila de tarefas.

3. (Opcional) Envie um trabalho de amostra para sua nova fila de trabalhos. Para obter mais informações, consulte [Exemplo de definição de trabalhos](#), [Criar uma definição de tarefa de nó único](#) e [Tutorial: enviar um trabalho](#).

Usar uma AMI de workload de GPU

Para executar cargas de trabalho de GPU em seus recursos de computação do AWS Batch, você deve usar uma AMI com suporte para GPU. Para obter mais informações, consulte [Como trabalhar com GPUs no Amazon ECS](#) e [AMIs otimizadas para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Em ambientes computacionais gerenciados, se o ambiente computacional especificar qualquer família de instância ou tipo de instância p6, p3, p4, p5, g3, g3s, g4, g5 ou g6, o AWS Batch usa uma AMI otimizada para GPU do Amazon ECS.

Em ambientes de computação não gerenciados, é recomendada uma AMI otimizada para GPU do Amazon ECS. Você pode usar as operações AWS Command Line Interface ou AWS Systems Manager Parameter Store [GetParameter](#), [GetParameters](#), e [GetParametersByPath](#) para recuperar os metadados para as AMIs otimizadas para GPU do Amazon ECS recomendadas.

Note

A família de p5 instância só é compatível com versões iguais ou posteriores à AMI otimizada para GPU 20230912 do Amazon ECS, e elas são incompatíveis com os tipos de instância p2 e g2. Se você precisar usar instâncias p5, certifique-se de que seu ambiente computacional não contenha instâncias p2 ou g2 e use o Batch AMI padrão mais recente. A criação de um novo ambiente de computação usará a AMI mais recente, mas se você estiver atualizando seu ambiente de computação para incluir p5, você pode garantir que está usando a AMI mais recente configurando o [updateToLatestImageVersion](#) como true nas propriedades ComputeResource. Para obter mais informações sobre a compatibilidade do AMI com instâncias de GPU, consulte [Como trabalhar com GPUs no Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Os exemplos a seguir mostram como usar o comando [GetParameter](#).

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended \
                        --region us-east-2 --output json
```

A saída inclui as informações de AMI no parâmetro Value.

```
{
  "Parameter": {
    "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
    "LastModifiedDate": 1555434128.664,
    "Value": "{\"schema_version\":1,\"image_name\": \"amzn2-ami-ecs-gpu-
hvm-2.0.20190402-x86_64-ebs\", \"image_id\": \"ami-083c800fe4211192f\", \"os\": \"Amazon
Linux 2\", \"ecs_runtime_version\": \"Docker version 18.06.1-ce\", \"ecs_agent_version
\": \"1.27.0\"}",
    "Version": 9,
    "Type": "String",
    "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended"
  }
}
```

Python

```
from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id   = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

A saída inclui apenas o ID da AMI e nome da AMI:

```
image_id   = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Os exemplos a seguir demonstram o uso de [GetParameters](#).

AWS CLI

```
$ aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                           /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                           --region us-east-2 --output json
```

A saída inclui os metadados completos para cada um dos parâmetros:

```
{
  "InvalidParameters": [],
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
      "LastModifiedDate": 1555434128.749,
      "Value": "ami-083c800fe4211192f",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
      "LastModifiedDate": 1555434128.712,
      "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
      "Version": 9,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    }
  ]
}
```

Python

```
from __future__ import print_function

import boto3
```

```

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
    Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
          '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])

```

A saída inclui o ID da AMI e nome da AMI, usando o caminho completo para os nomes.

```

/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs

```

Os exemplos a seguir mostram como usar o comando [GetParametersByPath](#).

AWS CLI

```

$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \
                                --region us-east-2 --output json

```

A saída inclui todos os metadados completos para todos os parâmetros no caminho especificado.

```

{
  "Parameters": [
    {
      "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
      "LastModifiedDate": 1555434128.801,
      "Value": "1.27.0",
      "Version": 8,
      "Type": "String",
      "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
    },
    {

```

```

        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
        "LastModifiedDate": 1548368308.213,
        "Value": "Docker version 18.06.1-ce",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
        "LastModifiedDate": 1555434128.749,
        "Value": "ami-083c800fe4211192f",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
        "LastModifiedDate": 1555434128.712,
        "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
        "LastModifiedDate": 1548368308.143,
        "Value": "Amazon Linux 2",
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
    },
    {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
        "LastModifiedDate": 1548368307.914,
        "Value": "1",

```

```
        "Version": 1,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
    }
]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

A saída inclui os valores de todos os nomes de parâmetro no caminho especificado, usando o caminho completo para os nomes.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Para obter mais informações, consulte [Recuperar os metadados da AMI otimizada para Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Depreciação do Amazon Linux

A AMI Amazon Linux (também chamada de Amazon Linux 1) chegou ao fim de sua vida útil em 31 de dezembro de 2023. O AWS Batch encerrou o suporte à AMI do Amazon Linux, pois ele não receberá atualizações de segurança ou correções de bugs a partir de 1º de janeiro de 2024. Para obter mais informações sobre o fim da vida útil do Amazon Linux, consulte [Perguntas frequentes sobre o AL](#).

Recomendamos que você atualize os ambientes de computação existentes baseados no Amazon Linux para o Amazon Linux 2023 para evitar interrupções imprevistas da workload e continuar a receber atualizações de segurança e outras.

Seus ambientes de computação que usam a AMI Amazon Linux podem continuar funcionando após a data de fim de vida útil de 31 de dezembro de 2023. Porém, esses ambientes de computação não receberão mais novas atualizações de software, patches de segurança ou correções de bugs do AWS. É sua responsabilidade manter esses ambientes de computação na AMI Amazon Linux após o fim da vida útil. Recomendamos a migração de ambientes de computação do AWS Batch para o Amazon Linux 2023 ou Amazon Linux 2 para manter o desempenho e a segurança ideais.

Para obter ajuda para migrar o AWS Batch da AMI Amazon Linux para o Amazon Linux 2023 ou Amazon Linux 2, consulte [Atualizar ambientes de computação: AWS Batch](#).

Descontinuação da AMI Amazon EKS Amazon Linux 2

A AWS encerrará o suporte às AMIs Amazon Linux 2 otimizadas para Amazon EKS a partir de 26/11/25. Recomendamos a migração dos ambientes computacionais Amazon EKS AWS Batch para o Amazon Linux 2023 antes de 26/11/2025 para manter o desempenho e a segurança ideais.

Embora você possa continuar usando as AMIs do Amazon Linux 2 otimizadas pelo Amazon EKS fornecidas pelo Batch em seus ambientes computacionais do Amazon EKS após a data de fim do suporte em 26/11/25, esses ambientes computacionais não receberão mais novas atualizações de software, patches de segurança ou correções de bugs da AWS. É sua responsabilidade manter esses ambientes computacionais no AMI Amazon Linux 2 atualizados para Amazon EKS após o fim da vida útil.

Para obter mais informações sobre o fim da vida útil do Amazon EKS AL2, consulte [Perguntas frequentes sobre a descontinuação da AMI Amazon EKS](#) no Guia do usuário Amazon EKS.

Para ajuda para migrar os ambientes computacionais AWS Batch Amazon EKS do Amazon Linux 2 para o Amazon Linux 2023, consulte [Como fazer o upgrade do EKS AL2 para o EKS AL2023](#).

Descontinuação da AMI Amazon ECS Amazon Linux 2

AWS encerrará o suporte para o Amazon Linux 2. A partir de janeiro de 2026, AWS Batch mudará a AMI padrão para novos ambientes computacionais do Amazon ECS do Amazon Linux 2 para o Amazon Linux 2023. Recomendamos a migração dos ambientes de computação AWS Batch Amazon ECS para o Amazon Linux 2023 para manter o desempenho e a segurança ideais.

Para obter mais informações sobre o fim da vida útil do Amazon Linux 2, consulte [Perguntas frequentes sobre o Amazon Linux 2](#).

Para saber mais sobre as diferenças entre o Amazon Linux 2 e o Amazon Linux 2023, consulte [Comparação entre o Amazon Linux 2023 e o Amazon Linux 2](#) no Guia do usuário do Amazon Linux 2023.

Para obter informações sobre mudanças do Amazon Linux 2023 para o AMI Amazon otimizado para ECS, consulte [Migrar de um Amazon Linux 2 para um Amazon Linux 2023 AMI otimizado para ECS](#) no Guia do usuário Amazon ECS.

Para ajuda para migrar os ambientes computacionais AWS Batch Amazon ECS do Amazon Linux 2 para o Amazon Linux 2023, consulte [Como migrar do ECS AL2 para o ECS AL2023](#).

Use os modelos de lançamento do Amazon EC2 com AWS Batch

AWS Batch suporta o uso de modelos de lançamento do Amazon EC2 com seus ambientes computacionais do EC2. Com os modelos de lançamento, você pode modificar a configuração padrão dos seus recursos AWS Batch computacionais sem precisar criar modelos personalizados. AMIs

Note

Os modelos de lançamento não são compatíveis com os recursos do AWS Fargate.

Você deve criar um modelo de execução antes de associá-lo a um ambiente de computação. É possível criar um modelo de execução usando o console Amazon EC2. Ou você pode usar o AWS CLI ou um AWS SDK. Por exemplo, o arquivo JSON a seguir representa um modelo de execução que redimensiona o volume de dados do Docker para o recurso de AWS Batch computação padrão AMI e também o define para ser criptografado.


```
{
  "LaunchTemplateName": "increase-container-volume-encrypt",
  "LaunchTemplateData": {
    "BlockDeviceMappings": [
      {
        "DeviceName": "/dev/xvda",
        "Ebs": {
          "Encrypted": true,
          "VolumeSize": 100,
          "VolumeType": "gp2"
        }
      }
    ]
  }
}
```

Você pode criar o modelo de execução anterior salvando o JSON em um arquivo chamado `lt-data.json` e executando o AWS CLI comando a seguir.

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Para obter mais informações sobre modelos de execução, consulte [Executar uma instância de um modelo de execução](#), no Guia do usuário do Amazon EC2.

Se você usar um modelo de execução para criar seu ambiente de computação, poderá mover os seguintes parâmetros de ambiente de computação existentes para o seu modelo de execução:

Note

Suponha que qualquer um desses parâmetros (exceto as tags do Amazon EC2) seja especificado no modelo de execução e na configuração do ambiente de computação. Então, os parâmetros do ambiente de computação têm precedência. As tags do Amazon EC2 são mescladas entre o modelo de lançamento e a configuração do ambiente de computação. Se houver uma colisão na chave de tag, o valor na configuração do ambiente de computação tem precedência.

- Pares de chaves do Amazon EC2
- ID da AMI do Amazon EC2

- Grupo de segurança IDs
- Etiquetas do Amazon EC2

Os seguintes parâmetros do modelo de lançamento são ignorados por AWS Batch:

- Tipo de instância (especifique os tipos de instâncias desejados ao criar o ambiente de computação)
- Função de instância (especifique a função de instância desejada ao criar o ambiente de computação)
- Sub-redes de interface de rede (especifique as sub-redes desejadas ao criar seu ambiente de computação)
- Opções de mercado de instâncias (AWS Batch deve controlar a configuração da instância spot)
- Desativar o encerramento da API (AWS Batch deve controlar o ciclo de vida da instância)

AWS Batch somente atualiza o modelo de lançamento com uma nova versão do modelo de lançamento durante as atualizações da infraestrutura. Para obter mais informações, consulte [Atualizar um ambiente computacional no AWS Batch](#).

Modelos de execução padrão e substituídos

Você pode definir um modelo de execução padrão para o ambiente computacional e um modelo de execução substituto para tipos e famílias de instâncias específicos. Isso pode ser útil para que o modelo padrão seja usado para a maioria dos tipos de instância nos ambientes de computação.

As variáveis de substituição `$Default` e `$Latest` podem ser usadas em vez de nomear uma versão específica. Se você não fornecer um modelo de execução substituto, o modelo de execução padrão será aplicado automaticamente.

Se você usar a `$Latest` variável `$Default` ou, AWS Batch aplicará as informações atuais no momento em que o ambiente computacional for criado. Se a versão padrão ou mais recente mudar no futuro, você deverá atualizar as informações por meio do - [UpdateComputeEnvironment](#) ou por meio do Console de gerenciamento da AWS - AWS Batch.

Para fornecer flexibilidade adicional, você pode definir modelos de execução substitutivos aplicados a tipos ou famílias de instâncias computacionais específicos.

 Note

Você pode especificar até dez (10) modelos de substituições de execução para cada ambiente computacional.

Use o parâmetro `targetInstanceTypes` para selecionar o tipo de instância ou a família que deve usar esse modelo de execução substituto. O tipo de instância ou família deve ser identificado primeiro pelo parâmetro [instanceTypes](#).

Se você definir substituições do modelo de execução e decidir removê-las posteriormente, poderá passar uma matriz vazia para cancelar a definição do parâmetro [overrides](#) na operação da API [UpdateComputeEnvironment](#). Você também pode optar por não incluir o parâmetro `overrides` ao enviar a operação da API [UpdateComputeEnvironment](#). Para obter mais informações, consulte [LaunchTemplateSpecification.overrides](#)

Para obter mais informações, consulte

[LaunchTemplateSpecificationOverride.targetInstanceTypes](#) no Guia de referência AWS Batch da API.

Dados de usuário do Amazon EC2 em modelos de execução

Você pode fornecer os dados de usuário do Amazon EC2 em seu modelo de execução que é executado por [cloud-init](#) quando suas instâncias são iniciadas. Seus dados de usuário podem executar cenários de configuração comuns, incluindo, dentre outros:

- [Incluindo usuários ou grupos](#)
- [Instalar pacotes](#)
- [Criar partições e sistemas de arquivos](#)

Os dados do usuário do Amazon EC2 em modelos de execução devem estar no formato [MIME multi-part archive](#). Isso ocorre porque seus dados de usuário são mesclados com outros dados de AWS Batch usuário necessários para configurar seus recursos computacionais. Você pode combinar vários blocos de dados de usuário em um único arquivo MIME de várias partes. Por exemplo, convém combinar um boothook de nuvem que configure o daemon do Docker com um script do shell de dados do usuário que grave informações do atendente de contêiner do Amazon ECS.

Se você estiver usando AWS CloudFormation, o [AWS::CloudFormation::Init](#) tipo pode ser usado com o script auxiliar [cfn-init](#) para realizar cenários de configuração comuns.

Um arquivo em várias partes MIME consiste nos seguintes componentes:

- O tipo de conteúdo e a declaração de limite da parte: `Content-Type: multipart/mixed; boundary=="==BOUNDARY=="`
- A declaração da versão MIME: `MIME-Version: 1.0`
- Um ou mais blocos de dados do usuário que contêm os seguintes componentes:
 - O limite de abertura, que sinaliza o início de um bloco de dados do usuário: `--==BOUNDARY==`. Você deve manter a linha antes desse limite em branco.
 - A declaração de tipo de conteúdo para o bloco: `Content-Type: text/cloud-config; charset="us-ascii"`. Para mais informações sobre tipos de conteúdo, consulte a [documentação do Cloud-Init](#). Você deve manter a linha após o branco da declaração do tipo de conteúdo.
 - O conteúdo de dados do usuário, por exemplo, uma lista de comandos de shell ou diretivas do `cloud-init`.
- O limite de fechamento que sinaliza o fim do arquivo MIME de várias partes: `--==BOUNDARY==--`. Você deve manter a linha antes do branco do limite de fechamento.

 Note

Se você adicionar dados de usuário a um modelo de execução no console do Amazon EC2, você pode colá-los como texto simples ou fazer upload de um arquivo. Ou você pode fazer o upload de um arquivo. Se você usa o AWS CLI ou um AWS SDK, deve primeiro base64 codificar os dados do usuário e enviar essa string como o valor do UserData parâmetro ao chamar [CreateLaunchTemplate](#), conforme mostrado neste arquivo JSON.

```
{
  "LaunchTemplateName": "base64-user-data",
  "LaunchTemplateData": {
    "UserData":
      "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWZzZS1jb250YWluZXItbm9sdW..."
  }
}
```

Tópicos

- [Referência: exemplos de modelos de inicialização do Amazon EC2](#)

Referência: exemplos de modelos de inicialização do Amazon EC2

Veja a seguir um exemplo de arquivo MIME com várias partes que você pode usar para criar seus próprios modelos.

Exemplos

- [Exemplo: montar um sistema de arquivos existente do Amazon EFS](#)
- [Exemplo: substituir a configuração padrão do atendente de contêiner do Amazon ECS](#)
- [Exemplo: montar um sistema de arquivos Amazon FSx for Lustre existente](#)

Exemplo: montar um sistema de arquivos existente do Amazon EFS

Example

Este exemplo de arquivo MIME de várias partes configura o recurso de computação para instalar o pacote `amazon-efs-utils` e montar um sistema de arquivos existente do Amazon EFS em `/mnt/efs`.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY--
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--MYBOUNDARY--
```

Exemplo: substituir a configuração padrão do atendente de contêiner do Amazon ECS

Example

Este exemplo de arquivo MIME de várias partes substitui as configurações padrão de limpeza de imagem de docker para um recurso de computação.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

--MYBOUNDARY==
```

Exemplo: montar um sistema de arquivos Amazon FSx for Lustre existente

Example

Este exemplo de arquivo MIME de várias partes configura o recurso computacional para instalar o `lustre2.10` pacote da Biblioteca Extras e montar um sistema de arquivos existente FSx para Lustre em `/scratch` e um nome de montagem de `fsx`. Este exemplo é para o Amazon Linux 2. Para obter instruções de instalação para outras distribuições Linux, consulte [Instalando o Lustre Client no Guia do usuário do Amazon FSx for Lustre](#). Para obter mais informações, consulte [Montar seu sistema de FSx arquivos Amazon automaticamente](#) no Guia do usuário do Amazon FSx for Lustre.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
```

```
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
  ${fsx_directory}

--==MYBOUNDARY==--
```

Nos membros [volumes](#) e [mountPoints](#) das propriedades do contêiner, os pontos de montagem devem ser mapeados para o contêiner.

```
{
  "volumes": [
    {
      "host": {
        "sourcePath": "/scratch"
      },
      "name": "Scratch"
    }
  ],
  "mountPoints": [
    {
      "containerPath": "/scratch",
      "sourceVolume": "Scratch"
    }
  ],
}
```

Configuração do serviço de metadados de instância (IMDS)

O Serviço de metadados de instância (IMDS) fornece metadados sobre suas instâncias do EC2 aos aplicativos em execução nessas instâncias. Use o IMDSv2 para todos os novos workloads e migre os workloads existentes do IMDSv1 para o IMDSv2 para melhorar a segurança. Para obter mais informações sobre o IMDS e a configuração do IMDS, consulte [Usar metadados de instância para gerenciar sua instância do EC2](#) e [Configurar opções de metadados de instância para novas instâncias](#) no Guia do usuário do Amazon EC2.

Cenários de configuração

Escolha o método de configuração apropriado com base na configuração do seu ambiente computacional:

AMI padrão sem modelo de execução

Ao usar a AMI AWS Batch padrão e não especificar um modelo de execução, escolha uma das seguintes opções:

1. Usar a AMI padrão do Amazon Linux 2023 — O Amazon Linux 2023 exige o IMDSv2 por padrão. Ao criar seu ambiente computacional, selecione Amazon Linux 2023 como o tipo de imagem.
2. Definir a configuração do IMDSv2 em nível de conta — Configure sua conta AWS para exigir o IMDSv2 para todas as novas instâncias. Essa configuração afeta todas as novas instâncias que você executa na conta. Para obter instruções, consulte [Definir IMDSv2 como padrão para a conta](#) no Guia do usuário do Amazon EC2.

Note

A configuração do IMDS no nível da conta pode ser substituída pelo modelo de execução ou pela configuração da AMI. As configurações do modelo de execução têm precedência sobre as configurações em nível de conta.

AMI personalizada sem modelo de execução

Ao usar uma AMI personalizada sem um modelo de execução, escolha uma das seguintes opções:

1. Usar o Amazon Linux 2023 como base — Crie sua AMI personalizada usando o Amazon Linux 2023 como imagem base. Para obter informações sobre como criar AMIs personalizadas para o Batch, consulte [Tutorial: criar uma AMI de recurso de computação](#).
2. Configurar IMDSv2 em sua AMI personalizada — Ao criar sua AMI personalizada, configure-a para exigir o IMDSv2. Para as instruções, consulte [Configurar opções de metadados da instância para AMI personalizada](#) no Guia do usuário do Amazon EC2.
3. Definir a configuração do IMDSv2 em nível de conta — Configure sua conta AWS para exigir o IMDSv2 para todas as novas instâncias. Essa configuração afeta todas as novas instâncias que você executa na conta. Para obter instruções, consulte [Definir IMDSv2 como padrão para a conta](#) no Guia do usuário do Amazon EC2.

 Note

A configuração do IMDS no nível da conta pode ser substituída pelo modelo de execução ou pela configuração da AMI. As configurações do modelo de execução têm precedência sobre as configurações em nível de conta.

Usando modelos de execução

Ao usar modelos de execução em seu ambiente computacional, adicione opções de metadados ao seu modelo de execução para exigir o IMDSv2. Para obter mais informações sobre como usar modelos de execução com o Batch, consulte [Use os modelos de lançamento do Amazon EC2 com AWS Batch](#).

```
{
  "LaunchTemplateName": "batch-imdsv2-template",
  "VersionDescription": "IMDSv2 only template for Batch",
  "LaunchTemplateData": {
    "MetadataOptions": {
      "HttpTokens": "required"
    }
  }
}
```

Como criar o modelo de execução usando o AWS CLI:

```
aws ec2 create-launch-template --cli-input-json file://imds-template.json
```

Configurações do EC2

AWS Batch usa AMIs otimizadas do Amazon ECS para ambientes de computação EC2 e EC2 Spot. O padrão é o [Amazon Linux 2](#) (ECS_AL2). A partir de janeiro de 2026, o padrão mudará para [AL2023](#) (ECS_AL2023).

AWS encerrará o suporte para o Amazon Linux 2. Recomendamos a migração dos ambientes de computação AWS Batch Amazon ECS para o Amazon Linux 2023 para manter o desempenho e a segurança ideais. Para obter mais informações, consulte [Descontinuação da AMI Amazon ECS Amazon Linux 2](#).

Recomendamos que você atualize os ambientes de computação existentes baseados no Amazon Linux para o Amazon Linux 2023 para evitar interrupções imprevistas da workload e continuar a receber atualizações de segurança e outras.

Para obter ajuda para migrar o AWS Batch do Amazon Linux AMI para o Amazon Linux 2023, consulte [Como migrar do ECS AL2 para o ECS AL2023](#)

Tópicos

- [Como migrar do ECS AL2 para o ECS AL2023](#)

Como migrar do ECS AL2 para o ECS AL2023

O AL2023 é um sistema operacional baseado no Linux que foi projetado para fornecer um ambiente seguro, estável e de alta performance para as aplicações em nuvem. Para saber mais sobre as diferenças entre o AL2 e o AL2023, consulte [Comparação entre o Amazon Linux 2023 e o Amazon Linux 2](#) no Guia do usuário do Amazon Linux 2023.

A partir de janeiro de 2026, AWS Batch mudará a AMI padrão para novos ambientes computacionais do Amazon ECS do Amazon Linux 2 para o Amazon Linux 2023, pois o AWS [encerrará o suporte para o Amazon Linux 2](#). A AMI padrão é usada quando você não especifica um valor para o campo [ImageType.ec2Configuration](#) ao criar um novo ambiente computacional. Recomendamos a migração dos ambientes de computação AWS Batch Amazon ECS para o Amazon Linux 2023 para manter o desempenho e a segurança ideais.

Dependendo de como seu ambiente computacional está configurado, você pode usar um dos seguintes caminhos de atualização do AL2 para o AL2023.

Atualizar usando EC2Configuration.ImageType

- Se você não estiver usando um modelo de execução ou substituições de modelo de execução, altere [EC2Configuration.ImageType](#) para ECS_AL2023 (ou ECS_AL2023_NVIDIA ao usar instâncias de GPU) e execute [UpdateComputeEnvironment](#).
- [Se você especificar um EC2Configuration.ImageIdOverride, EC2Configuration.ImageType deverá corresponder ao tipo de AMI especificado em EC2Configuration.ImageIdOverride.](#)

Se você não fizer a correspondência entre ImageIdOverride e ImageType, o ambiente computacional poderá não funcionar corretamente.

Atualizar usando modelos de execução

- Se você usar um modelo de lançamento que especifica uma AMI com base em ECS_AL2023, certifique-se de que seu modelo de lançamento seja compatível com o Amazon Linux 2023. Para obter informações sobre mudanças do Amazon Linux 2023 para o AMI Amazon otimizado para ECS, consulte [Migrar de um Amazon Linux 2 para um Amazon Linux 2023 AMI otimizado para ECS](#) no Guia do usuário Amazon ECS.
- Para AMIs AL2023, verifique se todos os dados personalizados do usuário ou scripts de inicialização são compatíveis com o ambiente do AL2023 e com o sistema de gerenciamento de pacotes.

Atualize usando o CloudFormation

- Se você usa o CloudFormation para gerenciar seus ambientes de computação, atualize seu modelo para alterar a propriedade ImageType em Ec2Configuration de ECS_AL2 para ECS_AL2023 (ou ECS_AL2023_NVIDIA ao usar instâncias de GPU):

```
ComputeEnvironment:
  Type: AWS::Batch::ComputeEnvironment
  Properties:
    ComputeResources:
      Ec2Configuration:
        - ImageType: ECS_AL2023
```

Em seguida, atualize sua pilha CloudFormation para aplicar as alterações.

- Se seu modelo CloudFormation especificar uma AMI personalizada usando ImageIdOverride, certifique-se de que o ID AMI corresponda a uma AMI baseada em AL2023 e corresponda à configuração ImageType.

Considerações sobre a migração

Ao migrar do Amazon Linux 2 para o Amazon Linux 2023, considere o seguinte:

- Gerenciamento de pacotes — O Amazon Linux 2023 usa dnf em vez de yum para gerenciamento de pacotes.
- Serviços do sistema — Alguns serviços do sistema e suas configurações podem diferir entre o AL2 e o AL2023.

- Runtime do contêiner — Tanto o AL2 quanto o AL2023 oferecem suporte ao Docker, mas o AL2023 pode ter configurações padrão diferentes.
- Segurança — O AL2023 inclui recursos de segurança aprimorados e pode exigir atualizações nas configurações relacionadas à segurança.
- Instance Metadata Service Version 2 (IMDSv2) — O IMDSv2 é um serviço orientado a sessões que exige autenticação baseada em token para acessar os metadados da instância EC2, fornecendo segurança aprimorada. Para obter mais informações sobre o IMDS, consulte [Como o serviço de metadados de instância versão 2 funciona](#) no Guia do usuário do Amazon EC2.

Para obter uma lista abrangente de alterações e considerações de migração, consulte [Migrar de uma AMI do Amazon Linux 2 para uma AMI do Amazon Linux 2023 otimizada para Amazon ECS](#) no Guia do usuário Amazon ECS.

Estratégias de alocação de tipos de instâncias para o AWS Batch

Quando um ambiente de computação gerenciado é criado, o AWS Batch seleciona os tipos de instância no [instanceTypes](#) especificado que sejam mais adequados às necessidades dos trabalhos. A estratégia de alocação define o comportamento quando o AWS Batch precisar de capacidade adicional. Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate. Não especifique este parâmetro.

BEST_FIT (padrão)

O AWS Batch seleciona um tipo de instância que seja mais adequado às necessidades dos trabalhos com preferência para o tipo de instância de menor custo. Se as instâncias adicionais do tipo de instância selecionado não estiverem disponíveis, o AWS Batch aguardará até que as instâncias adicionais estejam disponíveis. Se não houver instâncias suficientes disponíveis ou se o usuário estiver atingindo os [limites de cotas de serviço do Amazon EC2](#), não serão executados trabalhos adicionais até que os trabalhos em execução no momento estejam concluídos. Essa estratégia de alocação mantém os custos mais baixos, mas pode limitar a escalabilidade. Se você estiver usando frota spot com o BEST_FIT, o perfil do IAM de frota spot deve ser especificada. O BEST_FIT não é compatível com a atualização de ambientes de computação. Para obter mais informações, consulte [Atualizar um ambiente computacional no AWS Batch](#).

 Note

O AWS Batch gerencia recursos da AWS na sua conta. Ambientes de computação com a estratégia de alocação BEST_FIT utilizavam originalmente configurações de lançamento por padrão. Porém, o uso de configurações de lançamento com novas contas da AWS será restrito ao longo do tempo. Portanto, a contar do final de abril de 2024, os ambientes de computação BEST_FIT recém-criados terão como padrão os modelos de lançamento. Se o perfil de serviço não tiver permissões para gerenciar modelos de inicialização, o AWS Batch poderá continuar a utilizar as configurações de inicialização. Os ambientes de computação existentes continuarão a usar as configurações de inicialização.

BEST_FIT_PROGRESSIVE

O AWS Batch seleciona os tipos de instância adicionais que sejam grandes o suficiente para atender aos requisitos dos trabalhos na fila. Os tipos de instância com um custo menor para cada unidade vCPU são preferidos. Se as instâncias adicionais dos tipos de instância selecionados anteriormente não estiverem disponíveis, o AWS Batch selecionará novos tipos de instância.

 Note

Para [trabalhos paralelos de vários nós](#), o AWS Batch escolhe o tipo de instância ideal disponível. Se o tipo de instância ficar indisponível devido à capacidade insuficiente, outros tipos de instância da família não serão executados.

SPOT_CAPACITY_OPTIMIZED

O AWS Batch seleciona um ou mais tipos de instância adicionais que sejam grandes o suficiente para atender aos requisitos dos trabalhos na fila. Os tipos de instância com menor probabilidade de serem interrompidos são preferidos. Essa estratégia de alocação só está disponível para recursos de computação de instâncias spot.

SPOT_PRICE_CAPACITY_OPTIMIZED

A estratégia de alocação otimizada para preço e capacidade analisa o preço e a capacidade para selecionar os grupos de instâncias spot com menor probabilidade de interrupção e com o preço mais baixo possível. Essa estratégia de alocação só está disponível para recursos de computação de instâncias spot.

 Note

Em vez disso, recomendamos utilizar `SPOT_PRICE_CAPACITY_OPTIMIZED` em vez de `SPOT_CAPACITY_OPTIMIZED` na maioria das instâncias.

As estratégias `BEST_FIT_PROGRESSIVE` e `BEST_FIT` usam instâncias spot ou sob demanda, e as estratégias `SPOT_CAPACITY_OPTIMIZED` e `SPOT_PRICE_CAPACITY_OPTIMIZED` usam instâncias spot. No entanto, o AWS Batch pode precisar exceder o `maxvCpus` para atender aos requisitos de capacidade. Nesse caso, o AWS Batch nunca excederá `maxvCpus` em mais de uma única instância.

Gerenciamento de memória de recursos de computação

Quando o agente de contêiner do Amazon ECS registra um recurso de computação da instância de contêiner em um ambiente de computação do cluster, o atendente deve determinar a quantidade de memória que o recurso de computação da instância de contêiner tem disponível para reservá-la para os seus trabalhos da tarefa. Devido à sobrecarga de memória da plataforma e a memória ocupada pelo kernel do sistema, esse número é diferente da quantidade de memória instalada para instâncias do Amazon EC2. Por exemplo, uma instância `m4.large` tem 8 GiB de memória instalada. No entanto, isso nem sempre traduz para 8192 MiB de memória disponível para trabalhos nos quais o recurso de computação é registrado.

Suponha que você especifique 8192 MiB para o trabalho e que nenhum dos seus recursos de computação tenha 8192 MiB de memória disponível ou maior para satisfazer a esse requisito. O trabalho não poderá ser colocado em seu ambiente de computação. Se você estiver usando um ambiente de computação gerenciado, o AWS Batch deverá executar um tipo de instância maior para acomodar a solicitação.

A AMI padrão de recursos de computação AWS Batch também reserva 32 MiB de memória para o agente de contêiner do Amazon ECS e outros processos críticos do sistema. Essa memória não está disponível para alocação de trabalhos. Para obter mais informações, consulte [Reservar memória do sistema](#).

O agente de contêiner do Amazon ECS usa a função `ReadMemInfo()` do Docker para consultar a memória disponível total para o sistema operacional. O Linux oferece utilitários de linha de comando para determinar a memória total.

Example- Determinar a memória total do Linux

O comando `free` retorna a memória total reconhecida pelo sistema operacional.

```
$ free -b
```

O exemplo a seguir é um exemplo de saída para uma instância `m4.large` executando a AMI do Amazon Linux otimizada para Amazon ECS.

```
              total        used        free      shared    buffers     cached
Mem:      8373026816  348180480  8024846336          90112    25534464    205418496
-/+ buffers/cache:  117227520  8255799296
```

Essa instância tem 8373026816 bytes de memória total. Isso significa que há 7985 MiB disponíveis para tarefas.

Tópicos

- [Reservar memória do sistema](#)
- [Tutorial: visualizar memória de recursos de computação](#)
- [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#)

Reservar memória do sistema

Se você ocupar toda a memória de um recurso de computação com seus trabalhos, é possível que os trabalhos disputem a memória com processos críticos do sistema e causem uma falha do sistema. O agente de contêiner do Amazon ECS fornece uma variável de configuração chamada `ECS_RESERVED_MEMORY`. Você pode usar essa variável de configuração para remover um número específico de MiB de memória do grupo alocado aos seus trabalhos. Isso reserva de forma efetiva a memória para processos críticos do sistema.

A AMI AWS Batch padrão de recursos de computação reserva 32 MiB de memória para o agente de contêiner do Amazon ECS e outros processos críticos do sistema. Recomendamos reservar um buffer de memória de 5% para o atendente de contêiner do Amazon ECS e outros processos críticos do sistema.

Tutorial: visualizar memória de recursos de computação

É possível visualizar a quantidade de memória que um recurso de computação registra no campo do console do Amazon ECS ou por meio da operação da API [DescribeContainerInstances](#). Se você estiver tentando maximizar a sua utilização de recursos fornecendo aos trabalhos o máximo de memória possível para um tipo de instância específico, você poderá observar a memória disponível para esse recurso de computação e atribuir essa quantidade de memória aos seus trabalhos.

Para visualizar a memória dos recursos de computação

1. Abra o console em <https://console.aws.amazon.com/ecs/v2>.
2. Escolha os Clusters e, em seguida, escolha o cluster que hospeda os recursos de computação a serem visualizados.

O nome do cluster do seu ambiente de computação começa com o seu nome do ambiente computacional.

3. Escolha Infraestrutura.
4. Em Instâncias de Contêiner, escolha a opção de instância de contêiner com a qual deseja estabelecer conexão.
5. A seção Recursos e Rede exibe a memória disponível registrada para o recurso de computação.

O valor da memória de Capacidade total é o mesmo registrado pelo recurso de computação com o Amazon ECS quando a instância foi executada primeiro, e o valor da memória Disponível é o valor ainda não atribuído aos trabalhos.

Considerações sobre memória e vCPU para AWS Batch no Amazon EKS

No AWS Batch do Amazon EKS, você pode especificar os recursos que forem disponibilizados para um contêiner. Por exemplo, você pode especificar valores `requests` ou `limits` para recursos de vCPU e memória.

Os seguintes são as restrições para especificar os recursos de vCPU:

- Pelo menos um valor de `requests` ou `limits` de vCPU deve ser especificado.
- Uma unidade de vCPU é equivalente a um núcleo físico ou virtual.
- O valor da vCPU deve ser inserido em números inteiros ou em acréscimos de 0,25.
- O menor valor válido de vCPU é 0,25.

- Se ambos forem especificados, o valor de `requests` deverá ser menor que ou igual ao valor de `limits`. Dessa forma, você pode configurar as configurações de vCPU flexíveis e rígidas.
- Os valores de vCPU não podem ser especificados no formato milliCPU. Por exemplo, 100m não é um valor válido.
- AWS Batch utiliza o valor `requests` para decisões em escala. Se um valor `requests` não for especificado, o valor `limits` será copiado para o valor `requests`.

Os seguintes são as restrições para especificar os recursos de memória:

- Pelo menos um valor de `requests` ou `limits` de memória deve ser especificado.
- Os valores de memória devem estar em mebibytes (MiBs).
- Se ambos forem especificados, o valor de `requests` deverá ser igual ao valor de `limits`.
- AWS Batch utiliza o valor `requests` para decisões em escala. Se um valor de `requests` não for especificado, o valor de `limits` será copiado para o valor de `requests`.

Os seguintes são as restrições para especificar os recursos de GPU:

- Se ambos forem especificados, o valor de `requests` deverá ser igual ao valor de `limits`.
- AWS Batch utiliza o valor `requests` para decisões em escala. Se um valor `requests` não for especificado, o valor `limits` será copiado para o valor `requests`.

Exemplos: definições de trabalho

O seguinte AWS Batch sobre a configuração de definição de trabalho do Amazon EKS configura compartilhamentos flexíveis de vCPU. Isso permite que o AWS Batch no Amazon EKS use toda a capacidade da vCPU para o tipo de instância. No entanto, caso existam outros trabalhos em execução, ao trabalho será alocado um máximo de 2 vCPUs. A memória é limitada a 2 GB.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
```

```

        "command": ["sleep", "60"],
        "resources": {
            "requests": {
                "cpu": "2",
                "memory": "2048Mi"
            }
        }
    }
]
}
}
}

```

O seguinte AWS Batch na definição de trabalho do Amazon EKS tem um valor request de 1 e aloca um máximo de 4 vCPUs para o trabalho.

```

{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "requests": {
              "cpu": "1"
            },
            "limits": {
              "cpu": "4",
              "memory": "2048Mi"
            }
          }
        }
      ]
    }
  }
}

```

O seguinte AWS Batch na configuração de definição de trabalho do Amazon EKS define um valor limits de vCPU de 1 e um valor de limits de memória de 1 GB.

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["sleep", "60"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ]
    }
  }
}
```

Quando o AWS Batch traduz um AWS Batch no trabalho do Amazon EKS em um pod do Amazon EKS, o AWS Batch copia o valor de `limits` para o valor de `requests`. Isso ocorre quando um valor de `requests` não é especificado. Quando você envia o exemplo de definição de trabalho de precedência, o pod spec é o seguinte.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
      - sleep
      - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
```

```
memory: 1024Mi
```

```
...
```

Reserva de Memória e Reserva de Nó de CPU

AWS Batch depende da lógica padrão do arquivo `bootstrap.sh` para reservas de vCPU e reservas de memória. Para mais informações sobre o arquivo `bootstrap.sh`, consulte [bootstrap.sh](#). Ao medir o tamanho dos seus recursos de vCPU e memória, considere os exemplos a seguir.

Note

Se nenhuma instância estiver em execução, as reservas de vCPU e reservas de memória podem, de início, afetar a lógica em escala do AWS Batch e a tomada de decisões. Depois que as instâncias estiverem em execução, o AWS Batch ajusta as alocações iniciais.

Exemplo: reserva de nó de CPU

O valor da reserva de CPU é calculado em milicores usando o número de vCPUs total disponível para a instância.

Número da vCPU	Porcentagem reservada
1	6%
2	1%
3-4	0,5%
4 e acima	0.25%

Usando os valores que precedem, o seguinte é verdadeiro:

- O valor da reserva de CPU para uma instância `c5.large` com 2 vCPUs é 70 m. Isso é calculado da seguinte maneira: $(1*60) + (1*10) = 70$ m.
- O valor da reserva de CPU para uma instância `c5.24xlarge` com 96 vCPUs é 310 m. Isso é calculado da seguinte maneira: $(1*60) + (1*10) + (2*5) + (92*2.5) = 310$ m.

Neste exemplo, há 1930 (calculadas de 2000-70) unidades de vCPU de milicore disponíveis para executar trabalhos em uma instância `c5.large`. Suponha que seu trabalho requeira 2 unidades de vCPU (2*1000 m). O trabalho não caberá em uma instância única `c5.large`. No entanto, um trabalho que exige 1.75 unidades de vCPU é adequado.

Exemplo: reserva de memória do nó

O valor da reserva de memória é calculado em mebibytes usando o seguinte:

- a capacidade da instância em mebibytes. Por exemplo, uma instância de 8 GB tem 7.748 MiB.
- O valor de `kubeReserved`. O valor de `kubeReserved` é a quantidade de memória a ser reservada para os daemons do sistema. O valor de `kubeReserved` é calculado no campo da seguinte forma: $((11 * \text{número de pods máximo compatível pelo tipo de instância}) + 255)$. Para informações sobre o número máximo de pods compatível com cada tipo de instância, consulte [eni-max-pods.txt](#)
- O valor `HardEvictionLimit`. Quando a memória disponível fica abaixo do valor `HardEvictionLimit`, a instância tenta remover os pods.

A fórmula para calcular a memória disponível para alocação é a seguinte:

$(\text{instance\_capacity\_in\_MiB}) - (11 * (\text{maximum\_number\_of\_pods})) - 255 - (\text{HardEvictionLimitvalor})$.

Uma instância `c5.large` suporta até 29 pods. Para uma instância `c5.large` de 8 GB com um valor `HardEvictionLimit` de 100 MiB, a memória disponível para alocação é 7074 MiB. Isso é calculado da seguinte maneira: $(7748 - (11 * 29) - 255 - 100) = 7074$ MiB. Neste exemplo, um trabalho MiB de 8.192 não cabe nessa instância, embora seja uma instância de 8 gibibyte (GiB).

DaemonSets

Ao usar `DaemonSets`, considere o seguinte:

- Se nenhum `AWS Batch` nas instâncias do Amazon EKS estiver em execução, `DaemonSets` pode afetar inicialmente a lógica em escala do `AWS Batch` e a tomada de decisões. Inicialmente, `AWS Batch` aloca 0,5 unidades de vCPU e 500 MiB para o `DaemonSets` esperado. Depois que as instâncias estiverem em execução, o `AWS Batch` ajusta as alocações iniciais.
- Se um `DaemonSet` definir limites de vCPU ou limite de memória, os trabalhos `AWS Batch` no Amazon EKS terão menos recursos. Recomendamos que você mantenha o número de `DaemonSets` atribuídos a trabalhos `AWS Batch` o mais baixo possível.

Ambientes de computação Fargate

O Fargate é uma tecnologia que você pode usar AWS Batch para executar [contêineres](#) sem precisar gerenciar servidores ou clusters de instâncias do Amazon EC2. Com o Fargate, não é mais necessário provisionar, configurar nem escalar os clusters de máquinas virtuais para executar contêineres. Isso elimina a necessidade de escolher tipos de servidor, decidir quando dimensionar clusters ou otimizar o agrupamento de clusters.

Ao executar seus trabalhos com recursos do Fargate, você empacota sua aplicação em contêineres, especifica os requisitos de CPU e de memória, define as políticas do IAM e de rede e inicia a aplicação. Cada trabalho do Fargate tem seu próprio limite de isolamento e não compartilha o kernel subjacente, os recursos de CPU, os recursos de memória nem a interface de rede elástica com outro trabalho.

Tópicos

- [Quando usar o Fargate](#)
- [Definições de trabalho no Fargate](#)
- [Filas de trabalhos no Fargate](#)
- [Ambientes de computação no Fargate](#)

Quando usar o Fargate

Recomendamos usar o Fargate na maioria dos cenários. O Fargate inicia e escala a computação de acordo com os requisitos de recursos que você especifica para o contêiner. Com o Fargate, você não precisa provisionar em excesso nem pagar por servidores adicionais. Você também não precisa se preocupar com as especificidades dos parâmetros relacionados à infraestrutura, como o tipo de instância. Quando o ambiente de computação precisa ter a escala aumentada verticalmente, os trabalhos executados nos recursos do Fargate podem ser iniciados mais rapidamente. Normalmente, são necessários alguns minutos para ativar uma nova instância do Amazon EC2. No entanto, as tarefas executadas no Fargate podem ser provisionadas em cerca de 30 segundos. O tempo exato necessário depende de vários fatores, incluindo o tamanho da imagem do contêiner e o número de trabalhos.

No entanto, recomendamos que você use o Amazon EC2 se seus trabalhos exigirem qualquer um dos seguintes:

- Mais de 16 v CPUs

- Mais de 120 gibibytes (GiB) de memória
- UMA GPU
- Imagem de máquina da Amazon (AMI) personalizada
- Qualquer um dos parâmetros do [linuxParameters](#)

Se você tem um grande número de trabalhos, recomendamos usar a infraestrutura do Amazon EC2. Por exemplo, se o número de trabalhos em execução simultânea exceder os limites de controle de utilização do Fargate. Isso ocorre porque, com o EC2, os trabalhos podem ser enviados em uma taxa maior para os recursos do EC2 do que para os recursos do Fargate. Além disso, mais trabalhos podem ser executados simultaneamente quando você usa o EC2. Para obter mais informações, consulte [Service quotas Fargate](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Definições de trabalho no Fargate

AWS Batch as tarefas AWS Fargate ativadas não oferecem suporte a todos os parâmetros de definição de tarefas disponíveis. Alguns parâmetros são totalmente incompatíveis e outros se comportam de maneira diferente nos trabalhos do Fargate.

A lista a seguir descreve os parâmetros de definição de tarefa que não são válidos ou são restritos nos trabalhos do Fargate.

platformCapabilities

Deve ser especificado como FARGATE.

```
"platformCapabilities": [ "FARGATE" ]
```

type

Deve ser especificado como container.

```
"type": "container"
```

Parâmetros em containerProperties

executionRoleArn

Deve ser especificado para trabalhos em execução nos recursos do Fargate. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

fargatePlatformConfiguration

(Opcional, somente para definições de trabalho do Fargate). Especifica a versão da plataforma do Fargate a LATEST ou de uma versão da plataforma recente. Valores possíveis para `platformVersion` são 1.3.0, 1.4.0 e LATEST (padrão).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

instanceType, ulimits

Não é aplicável a trabalhos executados nos recursos do Fargate.

memory, vcpus

Essas configurações devem ser especificadas em `resourceRequirements`

privileged

Não especifique esse parâmetro ou especifique `false`.

```
"privileged": false
```

resourceRequirements

Os requisitos de memória e vCPU devem ser especificados usando [valores compatíveis](#). Os recursos de GPU não são compatíveis com trabalhos executados nos recursos do Fargate.

Se você usa o GuardDuty Runtime Monitoring, há uma pequena sobrecarga de memória para o agente GuardDuty de segurança. Portanto, o limite de memória deve incluir o tamanho do agente GuardDuty de segurança. Para obter informações sobre os limites de memória do GuardDuty Security Agent, consulte [Limites de CPU e memória](#) no Guia GuardDuty do usuário. Para obter informações sobre as práticas recomendadas, consulte [Como corrigir erros de falta de memória nas minhas tarefas do Fargate após habilitar o Monitoramento de runtime](#), no Guia do desenvolvedor do Amazon ECS.

```
"resourceRequirements": [  
  {"type": "MEMORY", "value": "512"},  
  {"type": "VCPU", "value": "0.25"}  
]
```


Parâmetros em linuxParameters

devices, maxSwap, sharedMemorySize, swappiness, tmpfs

Não é aplicável a trabalhos que sejam executados nos recursos do Fargate.

Parâmetros em logConfiguration

logDriver

Somente awslogs e splunk são compatíveis. Para obter mais informações, consulte [Usar o driver de log awslogs.](#)

Membros em networkConfiguration

assignPublicIp

Se a sub-rede privada não tiver um gateway NAT conectado para enviar tráfego para a Internet, [assignPublicIp](#) deverá ser “ENABLED”. Para obter mais informações, consulte [AWS Batch Função de execução do IAM.](#)

Filas de trabalhos no Fargate

AWS Batch as filas de trabalho AWS Fargate ativadas permanecem essencialmente inalteradas. A única restrição é que os ambientes de computação listados em `computeEnvironmentOrder` devem ser todos ambientes de computação Fargate (FARGATE ou FARGATE_SPOT). Os ambientes de computação EC2 e Fargate não podem ser misturados.

Ambientes de computação no Fargate

AWS Batch os ambientes computacionais ativados AWS Fargate não oferecem suporte a todos os parâmetros do ambiente computacional disponíveis. Alguns parâmetros são totalmente incompatíveis. Outros têm requisitos específicos para o Fargate.

A lista a seguir descreve os parâmetros do ambiente de computação que não são válidos ou são restritos nos trabalhos do Fargate.

type

Esse parâmetro precisa ser MANAGED.

```
"type": "MANAGED"
```

Parâmetros no objeto computeResources

allocationStrategy, bidPercentage, desiredvCpus, imageId, instanceTypes, ec2Configuration, ec2KeyPair, instanceRole, launchTemplate, minvCpus, placementGroup, spotIamFleetRole

Eles não são aplicáveis aos ambientes de computação Fargate e não podem ser fornecidos.

subnets

Se as sub-redes listadas nesse parâmetro não tiverem gateways NAT conectados, o parâmetro assignPublicIp na definição do trabalho deverá ser definido como ENABLED.

tags

Isso não é aplicável aos ambientes de computação Fargate e não pode ser fornecido. Para especificar tags para ambientes de computação Fargate, use o parâmetro tags que não está no objeto computeResources.

type

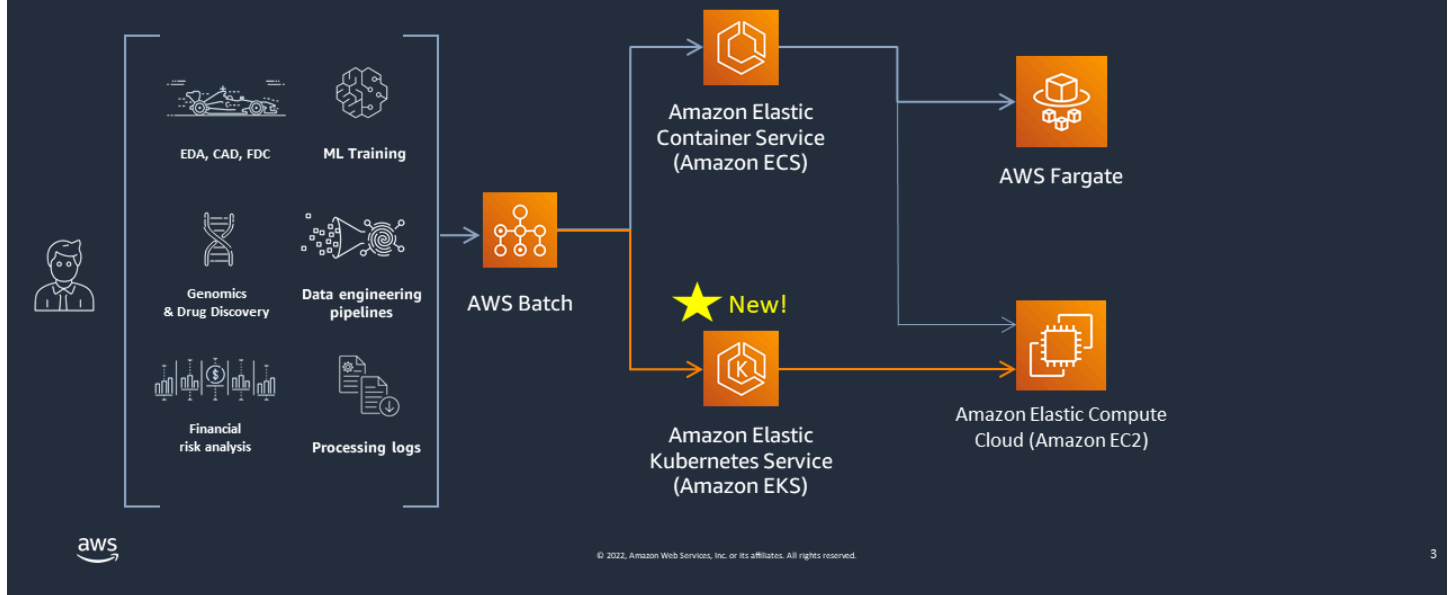
Isso deve ser FARGATE ou FARGATE_SPOT.

```
"type": "FARGATE_SPOT"
```

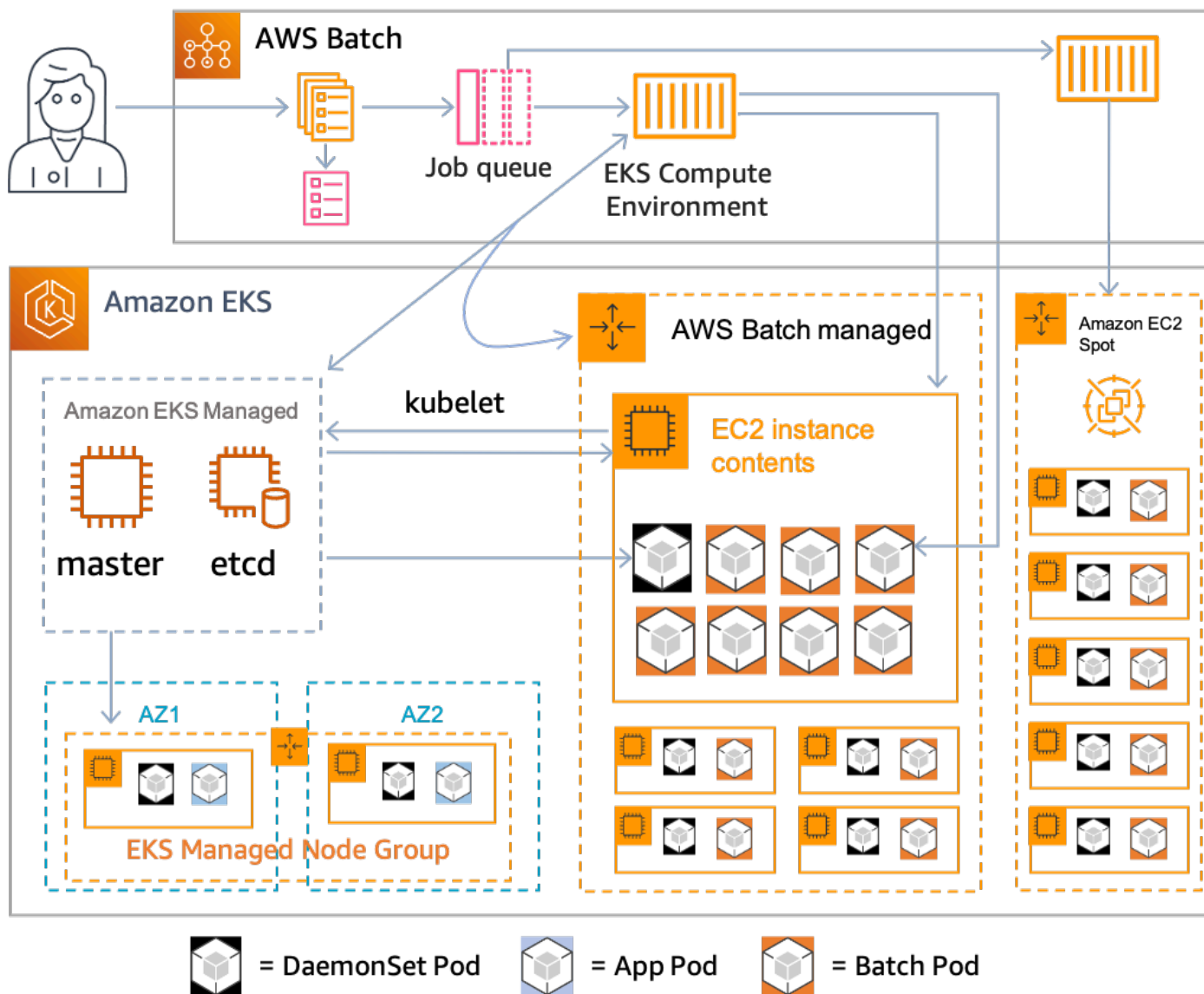
Ambientes de computação Amazon EKS

[Conceitos básicos do AWS Batch no Amazon EKS](#) fornece um pequeno guia para criar ambientes de computação EKS. Esta seção fornece mais detalhes sobre os ambientes de computação do Amazon EKS.

AWS Batch for Amazon Elastic Kubernetes Service (Amazon EKS)



AWS Batch simplifica suas cargas de trabalho em lotes nos clusters do Amazon EKS fornecendo recursos gerenciados em lotes. Isso inclui filas, rastreamento de dependências, tentativas e prioridades gerenciadas de tarefas, gerenciamento de pods e escalabilidade de nós. AWS Batch pode lidar com várias zonas de disponibilidade e vários tipos e tamanhos de instâncias do Amazon EC2. AWS Batch integra várias das melhores práticas do Amazon EC2 Spot para executar suas cargas de trabalho de forma tolerante a falhas, permitindo menos interrupções. Você pode usar o AWS Batch para executar um punhado de trabalhos noturnos ou milhões de trabalhos essenciais à missão com confiança.



AWS Batch é um serviço gerenciado que orquestra cargas de trabalho em lotes em seus Kubernetes clusters que são gerenciados pelo Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch conduz essa orquestração externamente aos seus clusters usando um modelo de “sobreposição”. Como AWS Batch é um serviço gerenciado, não há Kubernetes componentes (por exemplo, operadores ou recursos personalizados) para instalar ou gerenciar em seu cluster. AWS Batch só precisa que seu cluster seja configurado com controles de acesso baseados em funções (RBAC) que permitem AWS Batch a comunicação com o servidor da API. Kubernetes AWS Batch chamadas Kubernetes APIs para criar, monitorar e excluir Kubernetes pods e nós.

AWS Batch tem lógica de escalabilidade integrada para escalar Kubernetes nós com base na carga da fila de trabalhos com otimizações em termos de alocações de capacidade de trabalho. Quando a fila de trabalhos está vazia AWS Batch , reduz os nós até a capacidade mínima que você define,

que por padrão é zero. AWS Batch gerencia todo o ciclo de vida desses nós e decora os nós com rótulos e manchas. Dessa forma, outras Kubernetes cargas de trabalho não são colocadas nos nós gerenciados pelo AWS Batch. A exceção a isso é `DaemonSets`, que pode direcionar AWS Batch os nós para fornecer monitoramento e outras funcionalidades necessárias para a execução adequada dos trabalhos. Além disso, AWS Batch não executa trabalhos, especificamente pods, em nós do seu cluster que ele não gerencia. Assim, você pode usar lógica e serviços de escalabilidade separadamente para outros aplicativos no cluster.

Para enviar trabalhos para AWS Batch, você interage diretamente com a AWS Batch API. AWS Batch traduz trabalhos em podspecs e, em seguida, cria as solicitações para colocar pods em nós gerenciados pelo seu AWS Batch cluster Amazon EKS. Você pode usar ferramentas como `kubectl` para visualizar pods e nós em execução. Quando um pod conclui sua execução, AWS Batch exclui o pod criado para manter uma carga menor no Kubernetes sistema.

Você pode começar conectando um cluster válido do Amazon EKS com AWS Batch o. Em seguida, anexe uma fila de AWS Batch trabalhos a ela e registre uma definição de trabalho do Amazon EKS usando atributos podspec equivalentes. Por fim, envie trabalhos usando a operação [SubmitJob](#) da API que faz referência à definição do trabalho. Para obter mais informações, consulte [Conceitos básicos do AWS Batch no Amazon EKS](#).

Amazon EKS

Tópicos

- [AMI padrão do Amazon EKS](#)
- [Ambientes AMI mistos](#)
- [Versões do Kubernetes com suporte](#)
- [Atualizar a versão Kubernetes do ambiente de computação](#)
- [Responsabilidade compartilhada dos nós Kubernetes](#)
- [Execute um DaemonSet em nós AWS Batch gerenciados](#)
- [Personalize os modelos de lançamento do Amazon EKS](#)
- [Como fazer o upgrade do EKS AL2 para o EKS AL2023](#)

AMI padrão do Amazon EKS

Ao criar um ambiente computacional Amazon EKS, você não precisa especificar uma Amazon Machine Image (AMI). AWS Batch seleciona uma AMI otimizada para Amazon

EKS com base na Kubernetes versão e nos tipos de instância especificados na sua [CreateComputeEnvironment](#) solicitação. Em geral, recomendamos usar a seleção AMI do padrão. Para obter mais informações sobre o Amazon EKS otimizado AMIs, consulte [Amazon Linux otimizado para Amazon EKS AMIs](#) no Guia do usuário do Amazon EKS.

 Important

O Amazon Linux 2023 AMIs é o padrão AWS Batch para o Amazon EKS. AWS encerrará o suporte para o Amazon EKS AL2 otimizado e AL2 acelerado a partir de 26/11/25 AMIs. Você pode continuar usando o Amazon Linux 2 otimizado para Amazon EKS AWS Batch fornecido AMIs em seus ambientes computacionais do Amazon EKS após a end-of-support data de 26/11/25. No entanto, esses ambientes computacionais não receberão mais novas atualizações de software, patches de segurança ou correções de bugs. AWS Para obter mais informações sobre a atualização de AL2 para AL2023, consulte [Como fazer o upgrade do EKS AL2 para o EKS AL2023](#) o Guia do AWS Batch usuário.

Execute o comando a seguir para ver qual tipo de AMI AWS Batch foi selecionado para seu ambiente computacional Amazon EKS. O exemplo a seguir é um tipo de instância sem GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.32 AMI,
# depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.32"
  }
]
```

O exemplo a seguir é um tipo de instância de GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.32 AMI
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
  | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.32"
  }
]
```

```
}  
]
```

Ambientes AMI mistos

Você pode usar substituições de modelos de lançamento para criar ambientes computacionais com o Amazon Linux 2 () AL2 e o Amazon Linux 2023 (). AL2023 AMIs Isso é útil para usar diferentes AMIs arquiteturas ou durante períodos de migração durante a transição de para. AL2 AL2023

Note

AWS encerrará o suporte para o Amazon EKS AL2 otimizado e AL2 acelerado a partir de 26/11/25 AMIs. Embora você possa continuar usando o Amazon Linux 2 otimizado para Amazon EKS AWS Batch fornecido AMIs em seus ambientes computacionais do Amazon EKS após a end-of-support data de 26/11/25, esses ambientes computacionais não receberão mais novas atualizações de software, patches de segurança ou correções de bugs. AWS Ambientes mistos de AMI podem ser úteis durante o período de transição, permitindo que você migre gradualmente as cargas de trabalho e AL2023 mantenha a compatibilidade com as cargas de trabalho AL2 baseadas existentes.

Exemplo de configuração usando os dois tipos de AMI:

```
{  
  "computeResources": {  
    "launchTemplate": {  
      "launchTemplateId": "TemplateId",  
      "version": "1",  
      "userDataType": "EKS_BOOTSTRAP_SH",  
      "overrides": [  
        {  
          "instanceType": "c5.large",  
          "imageId": "ami-al2-custom",  
          "userDataType": "EKS_BOOTSTRAP_SH"  
        },  
        {  
          "instanceType": "c6a.large",  
          "imageId": "ami-al2023-custom",  
          "userDataType": "EKS_NODEADM"  
        }  
      ]  
    }  
  }  
}
```

```
    ]  
  },  
  "instanceTypes": ["c5.large", "c6a.large"]  
}  
}
```

Versões do Kubernetes com suporte

AWS Batch no Amazon EKS atualmente oferece suporte às seguintes Kubernetes versões:

- 1.34
- 1.33
- 1.32
- 1.31
- 1.30
- 1.29

Talvez você veja uma mensagem de erro semelhante à seguinte ao usar a operação de `CreateComputeEnvironment` API ou a operação de `UpdateComputeEnvironment` API para criar ou atualizar um ambiente de computação. Esse problema ocorre se você especificar uma versão Kubernetes não suportada em `EC2Configuration`.

At least one imageKubernetesVersion in EC2Configuration is not supported.

Para resolver esse problema, exclua o ambiente de computação e recrie-o com uma versão Kubernetes compatível.

Você pode realizar uma pequena atualização de versão no seu cluster Amazon EKS. Por exemplo, você pode atualizar o cluster de 1.xx para 1.yy mesmo que a versão secundária não seja compatível.

No entanto, o status do ambiente de computação pode mudar para `INVALID` depois de uma atualização de versão principal. Por exemplo, se você realizar uma atualização de versão principal de 1.xx para 2.yy. Se a versão principal não for suportada pelo AWS Batch, você verá uma mensagem de erro semelhante à seguinte.

reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported

Atualizar a versão Kubernetes do ambiente de computação

Com AWS Batch, você pode atualizar a Kubernetes versão de um ambiente computacional para oferecer suporte às atualizações de cluster do Amazon EKS. A Kubernetes versão de um ambiente computacional é a versão do Amazon EKS AMI para os Kubernetes nós que são AWS Batch iniciados para executar trabalhos. Você pode realizar uma atualização de versão Kubernetes em seus nós do Amazon EKS antes ou depois de atualizar a versão do ambiente de gerenciamento do cluster do Amazon EKS. Recomendamos que você atualize os nós após atualizar o plano de controle. Para obter mais informações, consulte [Atualizando uma versão Kubernetes do cluster Amazon EKS](#) no Guia do usuário do Amazon EKS.

Para atualizar a versão Kubernetes de um ambiente de computação, use a operação API [UpdateComputeEnvironment](#).

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources \
  'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Responsabilidade compartilhada dos nós Kubernetes

A manutenção dos ambientes de computação é uma responsabilidade compartilhada.

- Não altere nem remova AWS Batch nós, rótulos, manchas, namespaces, modelos de lançamento ou grupos de escalonamento automático. Não adicione manchas aos nós AWS Batch gerenciados. Se você fizer alguma dessas alterações, seu ambiente de computação não poderá ser compatível e ocorrerão falhas, incluindo instâncias ociosas.
- Não direcione seus pods para nós AWS Batch gerenciados. Se você direcionar seus pods para os nós gerenciados, ocorrerão falhas no escalonamento e filas de trabalhos paralisadas. Execute cargas de trabalho que não são usadas AWS Batch em nós autogerenciados ou grupos de nós gerenciados. Para obter mais informações, consulte [Grupos de nós gerenciados](#) no Guia do usuário do Amazon EKS.
- Você pode DaemonSet direcionar um para ser executado em nós AWS Batch gerenciados. Para obter mais informações, consulte [Execute um DaemonSet em nós AWS Batch gerenciados](#).

AWS Batch não atualiza automaticamente o ambiente AMIs computacional. É sua responsabilidade atualizá-las. Execute o comando a seguir para atualizá-lo AMIs para a versão mais recente da AMI.

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch não atualiza automaticamente a Kubernetes versão. Execute o comando a seguir para atualizar a Kubernetes versão do ambiente do seu computador para **1.32** o.

```
$ aws batch update-compute-environment \
  --compute-environment <compute-environment-name> \
  --compute-resources \
    'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Ao atualizar para uma AMI ou versão Kubernetes mais recente, você pode especificar se os trabalhos serão encerrados quando eles forem atualizados (`terminateJobsOnUpdate`) e quanto tempo esperar até que uma instância seja substituída se os trabalhos em execução não forem concluídos (`jobExecutionTimeoutMinutes`.) Para obter mais informações, consulte [Atualizar um ambiente computacional no AWS Batch](#) e a política de atualização de infraestrutura ([UpdatePolicy](#)) definida na operação da API [UpdateComputeEnvironment](#).

Execute um DaemonSet em nós AWS Batch gerenciados

AWS Batch define manchas nos Kubernetes nós AWS Batch gerenciados. Você pode DaemonSet direcionar um para ser executado em nós AWS Batch gerenciados com o seguinte `tolerations`.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
```

Outra maneira de fazer isso é com o `tolerations` a seguir.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoSchedule"
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoExecute"
```

Personalize os modelos de lançamento do Amazon EKS

AWS Batch no Amazon EKS oferece suporte a modelos de lançamento. Há restrições sobre o que seu modelo de lançamento pode fazer.

Important

- Para EKS AL2 AMIs, AWS Batch funciona `/etc/eks/bootstrap.sh`. Não execute `/etc/eks/bootstrap.sh` em seu modelo de lançamento ou scripts cloud-init user-data. Você pode adicionar outros parâmetros além do parâmetro `--kubenet-extra-args` ao [bootstrap.sh](#). Para fazer isso, defina a variável `AWS_BATCH_KUBELET_EXTRA_ARGS` no arquivo `/etc/aws-batch/batch.config`. Consulte o código a seguir para ver um exemplo.
- Para o EKS AL2023, AWS Batch utiliza o [NodeConfigSpec](#) do EKS para fazer com que as instâncias se juntem ao cluster EKS. AWS Batch é [ClusterDetails](#) preenchido [NodeConfigSpec](#) para o cluster EKS e você não precisa especificá-los.

Note

Recomendamos que você não defina nenhuma das [NodeConfigSpec](#) configurações a seguir no modelo de lançamento, pois AWS Batch substituirá seus valores. Para obter mais informações, consulte [Responsabilidade compartilhada dos nós Kubernetes](#).

- Taints
- Cluster Name
- apiServerEndpoint
- certificatAuthority
- CIDR
- Não crie rótulos com o prefixo `batch.amazonaws.com/`

Note

Se o modelo de lançamento for alterado após [CreateComputeEnvironment](#) ser chamado, [UpdateComputeEnvironment](#) deverá ser chamado para avaliar a versão do modelo de lançamento para substituição.

Tópicos

- [Adicionar argumentos extras de kubelet](#)
- [Configurar o runtime do contêiner](#)
- [Montar um volume do Amazon EFS](#)
- [IPv6 apoio](#)

Adicionar argumentos extras de **kubelet**

AWS Batch suporta a adição de argumentos extras ao kubelet comando. Para obter a lista completa de parâmetros compatíveis, consulte [kubelet](#) na documentação Kubernetes. No exemplo a seguir para EKS AL2 AMIs, *--node-labels mylabel=helloworld* é adicionado à linha de kubelet comando.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=="MYBOUNDARY=="

--MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch

echo AWS_BATCH_KUBELET_EXTRA_ARGS="--node-labels mylabel=helloworld" >> /etc/
aws-batch/batch.config

--MYBOUNDARY==
```

Para EKS AL2023 AMIs, o formato do arquivo é YAML. Para obter a lista completa de parâmetros compatíveis, consulte [NodeConfigSpec](#) na documentação Kubernetes. No exemplo a seguir para EKS AL2023 AMIs, *--node-labels mylabel=helloworld* é adicionado à linha de kubelet comando.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: application/node.eks.aws

apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  kubelet:
    flags:
      - --node-labels=mylabel=helloworld

--==MYBOUNDARY==--
```

Configurar o runtime do contêiner

Você pode usar a variável de ambiente AWS Batch `CONTAINER_RUNTIME` ambiente para configurar o tempo de execução do contêiner em um nó gerenciado. O exemplo a seguir define o runtime do contêiner para `containerd` quando `bootstrap.sh` for executado. Para obter mais informações, consulte [containerd](#) na documentação Kubernetes.

Se você estiver usando uma AMI otimizada `EKS_AL2023` ou `EKS_AL2023_NVIDIA`, não precisará especificar o runtime do contêiner, pois somente `containerd` é compatível.

Note

A variável de ambiente `CONTAINER_RUNTIME` é equivalente à opção `--container-runtime` de `bootstrap.sh`. Para obter mais informações, consulte [Options](#) na documentação Kubernetes.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
mkdir -p /etc/aws-batch
```

```
echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

---MYBOUNDARY---
```

Montar um volume do Amazon EFS

Você pode usar modelos de execução para montar volumes no nó. No exemplo a seguir, as configurações cloud-config packages e runcmd são usadas. Para obter mais informações, consulte [Exemplos de configuração de Nuvem](#) na documentação cloud-init.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary==="MYBOUNDARY==="

---MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
  >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

---MYBOUNDARY---
```

Para usar esse volume na tarefa, ele deve ser adicionado ao parâmetro [EksProperties](#) a. [RegisterJobDefinition](#) O exemplo a seguir é uma grande parte da definição do trabalho.

```
{
  "jobDefinitionName": "MyJobOnEks_EFS",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": ["ls", "-la", "/efs"],
```

```
        "resources": {
            "limits": {
                "cpu": "1",
                "memory": "1024Mi"
            }
        },
        "volumeMounts": [
            {
                "name": "efs-volume",
                "mountPath": "/efs"
            }
        ]
    },
    "volumes": [
        {
            "name": "efs-volume",
            "hostPath": {
                "path": "/mnt/efs"
            }
        }
    ]
}
```

No nó, o volume do Amazon EFS é montado no diretório `/mnt/efs`. No contêiner do trabalho do Amazon EKS, o volume é montado no diretório `/efs`.

IPv6 apoio

AWS Batch é compatível com clusters do Amazon EKS que têm IPv6 endereços. Nenhuma personalização é necessária para AWS Batch suporte. No entanto, antes de começar, recomendamos que você analise as considerações e condições descritas em [Atribuição de IPv6 endereços a pods e serviços no](#) Guia do usuário do Amazon EKS.

Como fazer o upgrade do EKS AL2 para o EKS AL2023

Os Amazon EKS otimizados AMLs estão disponíveis em duas famílias com base no Amazon Linux 2 (AL2) e no Amazon Linux 2023 (AL2023). AL2023 é um sistema operacional baseado em Linux projetado para fornecer um ambiente seguro, estável e de alto desempenho para seus aplicativos em

nuvem. Para obter mais informações sobre as diferenças entre AL2 e AL2023 consulte [Atualização do Amazon Linux 2 para o Amazon Linux 2023](#) no Guia do usuário do Amazon EKS.

 Important

AWS encerrará o suporte para o Amazon EKS AL2 otimizado e AL2 acelerado a partir de 26/11/25 AMIs. Recomendamos migrar os ambientes computacionais do AWS Batch Amazon EKS para o Amazon Linux 2023 antes de 26/11/25 para manter o desempenho e a segurança ideais. Embora você possa continuar usando o Amazon Linux 2 otimizado para Amazon EKS AWS Batch fornecido AMIs em seus ambientes computacionais do Amazon EKS após a end-of-support data de 26/11/25, esses ambientes computacionais não receberão mais novas atualizações de software, patches de segurança ou correções de bugs. AWS Posteriormente end-of-life, é sua [responsabilidade manter](#) esses ambientes computacionais na AMI Amazon Linux 2 otimizada para Amazon EKS.

Dependendo de como seu ambiente computacional está configurado, você pode usar um dos seguintes caminhos de atualização de AL2 para AL2023.

Atualize usando o Ec2Configuration. ImageType

- [Se você não estiver usando um modelo de execução ou substituições de modelo de execução, altere Ec2Configuration. ImageType](#) para EKS_AL2023 ou EKS_AL2023_NVIDIA e depois execute [UpdateComputeEnvironment](#).
- Se você especificar uma configuração [Ec2. ImageIdOverride](#) depois [Ec2Configuration. ImageType](#) deve corresponder ao tipo de AMI especificado em [Ec2Configuration. ImageIdOverride](#).

Se você não corresponder ImageIdOverride e ImageType, o nó não se juntará ao cluster.

Atualizar usando modelos de execução

- Se você tiver argumentos kubelet extras definidos em um modelo de execução ou substituição de modelo de execução, eles precisarão ser atualizados para o novo [formato de argumentos kubelet extras](#).

Se você não corresponder ao formato dos argumentos extras kubelet, os argumentos extras não serão aplicados.

- Pois AL2023 AMIs, containerd é o único ambiente de execução de contêiner compatível. Você não precisa especificar um runtime de contêiner para EKS_AL2023 no modelo de execução.

Você não pode especificar um tempo de execução de contêiner personalizado com EKS_AL2023.

- Se você usar um modelo de execução ou uma substituição de modelo de execução que especifique uma AMI com base em EKS_AL2023, você precisará definir [userDataType](#) como EKS_NODEADM.

Se você não corresponder `userDataType` e AMI, o nó não se juntará ao cluster EKS.

Ambientes de serviço para AWS Batch

Os ambientes de serviço permitem AWS Batch a integração com a SageMaker IA. Um ambiente de serviço contém os parâmetros de configuração específicos de SageMaker IA necessários AWS Batch para enviar e gerenciar trabalhos de SageMaker treinamento, ao mesmo tempo em que fornece recursos AWS Batch de filas, agendamento e gerenciamento de prioridades.

Com ambientes de serviço, cientistas de dados e engenheiros de ML podem enviar trabalhos de SageMaker treinamento com prioridades para atender às filas de trabalhos. Essa integração elimina a necessidade de coordenação manual dos workloads de ML, evita gastos excessivos acidentais e melhora a utilização de recursos nos fluxos de trabalho de machine learning da sua organização.

Tópicos

- [O que são ambientes de serviço no AWS Batch](#)
- [Estados do ambiente de serviço e ciclo de vida em AWS Batch](#)
- [Criar um ambiente de serviço no AWS Batch](#)
- [Atualizar um ambiente de serviço no AWS Batch](#)
- [Excluir um ambiente de serviço no AWS Batch](#)

O que são ambientes de serviço no AWS Batch

Um ambiente de serviço é um AWS Batch recurso que contém os parâmetros de configuração necessários para a integração AWS Batch com a SageMaker IA. Os ambientes de serviço permitem AWS Batch enviar e gerenciar trabalhos de SageMaker treinamento, ao mesmo tempo em AWS Batch que fornecem recursos de filas, agendamento e gerenciamento de prioridades.

Os ambientes de serviço abordam desafios comuns que as equipes de ciência de dados enfrentam ao gerenciar workloads de machine learning. As organizações geralmente limitam o número de instâncias disponíveis para modelos de treinamento para evitar gastos excessivos acidentais, atender às restrições orçamentárias, economizar custos com instâncias reservadas ou usar tipos de instância específicos para workloads. No entanto, os cientistas de dados podem querer executar mais workloads simultaneamente do que é possível com suas instâncias alocadas, exigindo coordenação manual para decidir quais workloads serão executadas e quando.

Esse desafio de coordenação afeta organizações de todos os tamanhos, desde equipes com apenas alguns cientistas de dados até operações em grande escala. À medida que as organizações

crescem, a complexidade aumenta, exigindo mais tempo para gerenciar a coordenação do workload e, muitas vezes, exigindo o envolvimento do administrador da infraestrutura. Esses esforços manuais desperdiçam tempo e reduzem a eficiência das instâncias, resultando em custos reais para os clientes.

Com ambientes de serviço, cientistas de dados e engenheiros de ML podem enviar trabalhos de SageMaker treinamento com prioridades para filas configuráveis, garantindo que as cargas de trabalho sejam executadas automaticamente sem intervenção assim que os recursos estiverem disponíveis. Essa integração aproveita os amplos recursos AWS Batch de enfileiramento e agendamento, permitindo que os clientes personalizem suas políticas de enfileiramento e agendamento de acordo com as metas da organização.

Como os ambientes de serviço funcionam com outros AWS Batch componentes

Os ambientes de serviço se integram a outros AWS Batch componentes para permitir o enfileiramento de tarefas de SageMaker treinamento:

- Filas de trabalhos - Os ambientes de serviço são associados às filas de trabalhos para permitir que a fila processe trabalhos de serviço para trabalhos de treinamento SageMaker
- Trabalhos de serviço - Quando você envia um trabalho de serviço para uma fila associada a um ambiente de serviço, AWS Batch usa a configuração do ambiente para enviar o trabalho de SageMaker treinamento correspondente
- Políticas de agendamento - Os ambientes de serviço trabalham com políticas AWS Batch de agendamento para priorizar e gerenciar a ordem de execução dos trabalhos de treinamento SageMaker

Essa integração permite que você aproveite os recursos maduros AWS Batch de enfileiramento e agendamento da empresa, mantendo a funcionalidade e a flexibilidade completas das tarefas de SageMaker treinamento.

Práticas recomendadas para ambientes de serviço

Os ambientes de serviço fornecem recursos para gerenciar trabalhos SageMaker de treinamento em grande escala. Seguir essas práticas recomendadas ajuda a otimizar o custo, o desempenho e a eficiência operacional, evitando problemas comuns de configuração que podem afetar seus fluxos de trabalho de machine learning.

Ao planejar a capacidade do ambiente de serviço, considere as cotas e os limites específicos que se aplicam à fila de trabalhos de SageMaker treinamento. Cada ambiente de serviço tem um limite máximo de capacidade expresso em número de instâncias, o que controla diretamente quantas tarefas de SageMaker treinamento podem ser executadas simultaneamente. Compreender esses limites ajuda a evitar a contenção de recursos e garante tempos previsíveis de execução do trabalho.

O desempenho ideal do ambiente de serviço depende da compreensão das características exclusivas do agendamento de tarefas de SageMaker treinamento. Diferentemente dos trabalhos em contêineres tradicionais, os trabalhos de serviços passam por um SCHEDULED estado enquanto a SageMaker IA adquire e provisiona as instâncias de treinamento necessárias. Isso significa que os horários de início do trabalho podem variar significativamente com base na disponibilidade da instância e na capacidade regional.

Important

Os ambientes de serviço têm cotas específicas que podem afetar sua capacidade de escalar as cargas de trabalho de SageMaker treinamento. Você pode criar até 50 ambientes de serviço por conta, com cada fila de trabalhos suportando somente um ambiente de serviço associado. Além disso, a carga útil da solicitação de serviço para trabalhos individuais é limitada a 10 KiB, e a API `SubmitServiceJob` está limitada a 5 transações por segundo por conta. Compreender esses limites durante o planejamento da capacidade evita restrições de escalabilidade inesperadas.

O monitoramento eficaz dos ambientes de serviço requer atenção tanto às métricas de serviço AWS Batch quanto às métricas de serviços de SageMaker IA. As [transições de estado de trabalho](#) fornecem informações valiosas sobre o desempenho do sistema, especialmente o tempo gasto no estado SCHEDULED, o que indica padrões de disponibilidade de capacidade. Os ambientes de serviço mantêm seus próprios estados de ciclo de vida semelhantes aos ambientes de computação, passando pelos estados CREATING, VALID, INVALID e DELETING que devem ser monitorados quanto à integridade operacional. Organizações com práticas de monitoramento maduras geralmente monitoram a profundidade da fila, as taxas de conclusão de trabalhos e os padrões de utilização de instâncias para otimizar suas configurações de ambiente de serviço ao longo do tempo.

Estados do ambiente de serviço e ciclo de vida em AWS Batch

Os ambientes de serviço mantêm estados de ciclo de vida que indicam seu status operacional atual e sua prontidão para processar SageMaker trabalhos de treinamento. A compreensão desses

estados ajuda você a monitorar a integridade do ambiente de serviço, solucionar problemas de configuração e garantir o processamento confiável do trabalho. O sistema de gerenciamento estadual segue padrões estabelecidos de ambientes computacionais e, ao mesmo tempo, acomoda os requisitos exclusivos da integração de tarefas de SageMaker treinamento.

Os estados do ambiente de serviço são gerenciados automaticamente AWS Batch com base na validação da configuração, disponibilidade de recursos e verificações de integridade operacional. Diferentemente dos ambientes computacionais que gerenciam a infraestrutura física, os ambientes de serviço se concentram na validação da configuração e na prontidão da integração com os serviços de SageMaker IA. As transições de estado fornecem visibilidade sobre se seu ambiente de serviço pode enviar e gerenciar trabalhos de SageMaker treinamento com sucesso.

Definições do estado do ambiente de serviço

Os ambientes de serviço podem estar em um dos quatro estados possíveis que indicam seu status operacional atual e sua prontidão para processar trabalhos de SageMaker treinamento. Cada estado representa uma fase específica no ciclo de vida do ambiente de serviço, desde a criação inicial, passando pela prontidão operacional até a eventual exclusão. A tabela a seguir descreve cada estado e seu significado:

Estado	Description
CREATING	O estado inicial quando você cria um ambiente de serviço. Durante esse estado, AWS Batch valida os parâmetros de configuração e estabelece a integração com os serviços de SageMaker IA. O ambiente de serviço não consegue processar trabalhos, e qualquer fila de trabalhos associada a ele não aceitará envios de trabalhos de serviço. O processo de criação geralmente é concluído em alguns segundos para ambientes de serviço configurados corretamente.
VALID	O estado operacional que indica que o ambiente de serviço passou por todas as verificações de validação da configuração e está pronto para processar os trabalhos de

Estado	Description
	SageMaker treinamento. Esse estado indica que a configuração do ambiente de serviço está correta, todas as permissões necessárias estão em vigor e AWS Batch pode enviar trabalhos com sucesso para a SageMaker IA em seu nome. Os ambientes de serviço passam a maior parte de seu ciclo de vida operacional nesse estado.
INVALID	Um estado que indica que o ambiente de serviço encontrou um problema de configuração ou permissão que o impede de processar trabalhos de SageMaker treinamento. As filas de trabalhos associadas a ambientes de serviço inválidos não conseguem processar novos envios de trabalhos de serviço até que os problemas subjacentes sejam resolvidos.
DELETING	O estado que ocorre quando você solicita a exclusão de um ambiente de serviço. Durante esse estado, AWS Batch garante que nenhuma tarefa ativa de SageMaker treinamento esteja associada ao ambiente e realiza as operações de limpeza necessárias. Os ambientes de serviço nesse estado não conseguem processar novos envios de trabalhos, e o processo de exclusão é concluído quando todos os recursos associados são devidamente limpos.

Transições de estado do ambiente de serviço

As transições de estado do ambiente de serviço ocorrem automaticamente com base em alterações de configuração, resultados de validação e monitoramento da integridade operacional. O AWS Batch serviço monitora continuamente a integridade do ambiente de serviço e atualiza os estados

adequadamente. Compreender essas transições ajuda você a prever quando as alterações de configuração entrarão em vigor e como resolver problemas que causam estados inválidos.

Após a criação e validação bem-sucedidas, os ambientes de serviço fazem a transição de `CREATING` para `VALID`. Essa transição confirma que todos os parâmetros de configuração estão corretos, as permissões necessárias do IAM estão configuradas corretamente e que o ambiente de serviço pode se integrar com êxito aos serviços de SageMaker IA. Uma vez no estado `VALID`, as filas de trabalhos associadas podem começar a processar os envios de trabalhos de serviço.

Os ambientes de serviço fazem a transição de `VALID` para `INVALID` quando a validação da configuração falha ou quando as dependências ficam indisponíveis. Isso pode ocorrer devido a modificações no perfil do IAM, alterações no limite de capacidade que violam cotas ou modificações de recursos externos que afetam a capacidade de funcionamento do ambiente de serviço. O campo de motivo do status fornece detalhes específicos sobre o que causou o estado inválido.

Os ambientes de serviço podem fazer a transição de volta para `VALID` de `INVALID` quando os problemas subjacentes forem resolvidos. Isso pode envolver a atualização das permissões do IAM, a correção das configurações de capacidade ou a restauração do acesso aos recursos necessários. A transição normalmente ocorre automaticamente quando AWS Batch detecta que os problemas de configuração foram resolvidos.

Criar um ambiente de serviço no AWS Batch

Antes de executar trabalhos SageMaker de treinamento em AWS Batch, você precisa criar um ambiente de serviço. Você pode criar um ambiente de serviço que contenha os parâmetros de configuração necessários AWS Batch para se integrar aos serviços de SageMaker IA e enviar trabalhos de SageMaker treinamento em seu nome.

Pré-requisitos

Antes de criar um ambiente de serviço, verifique se você tem:

- Permissões do IAM: permissões para criar e gerenciar ambientes de serviço. Para obter mais informações, consulte [AWS Batch Políticas, funções e permissões do IAM](#).

Create a service environment (AWS Console)

Use o AWS Batch console para criar um ambiente de serviço por meio da interface da web.

Para criar um ambiente de serviço

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Ambientes.
3. Escolha Criar ambiente, selecione Ambiente de serviço.
4. Para configuração do ambiente de serviço, escolha SageMaker AI.
5. Em Nome, insira um nome exclusivo para seu ambiente de serviço. Os caracteres válidos são a-z, A-Z, 0-9, - (hífen) e _ (sublinhado).
6. Em Número máximo de instâncias, insira o número máximo de instâncias de treinamento simultâneas
7. (Opcional) Adicione tags escolhendo Adicionar tag e inserindo pares de chave-valor.
8. Escolha Próximo.
9. Analise os detalhes do novo ambiente de serviço e escolha Criar ambiente de serviço.

Create a service environment (AWS CLI)

Use o `create-service-environment` comando para criar um ambiente de serviço com a AWS CLI.

Para criar um ambiente de serviço

1. Criar um ambiente de serviço com os parâmetros básicos necessários:

```
aws batch create-service-environment \  
  --service-environment-name my-sagemaker-service-env \  
  --service-environment-type SAGEMAKER_TRAINING \  
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10
```

2. (Opcional) Criar um ambiente de serviço com tags:

```
aws batch create-service-environment \  
  --service-environment-name my-sagemaker-service-env \  
  --service-environment-type SAGEMAKER_TRAINING \  
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10 \  
  --tags team=data-science,project=ml-training
```

3. Verifique se o ambiente do serviço foi criado com sucesso:

```
aws batch describe-service-environments \  

```



```
--service-environment my-sagemaker-service-env
```

O ambiente de serviço aparece na lista Ambientes com um estado CREATING. Quando a criação é concluída com sucesso, o estado muda para VALID e o ambiente de serviço está pronto para ter uma fila de trabalhos de serviço adicionada a ele para que o ambiente de serviço possa começar a processar os trabalhos.

Atualizar um ambiente de serviço no AWS Batch

Você pode atualizar um ambiente de serviço para modificar seus limites de capacidade, alterar seu estado operacional ou atualizar tags de recursos. As atualizações do ambiente de serviço permitem que você ajuste a capacidade à medida que seus requisitos de carga de trabalho de SageMaker treinamento mudam ou modifiquem as configurações operacionais sem recriar o ambiente. Antes de atualizar um ambiente de serviço, entenda quais parâmetros podem ser modificados e o impacto das mudanças nos trabalhos em execução.

Você pode alterar os limites de capacidade, o estado ou as tags de um ambiente de serviço.

Update a service environment (AWS Console)

Use o AWS Batch console para atualizar um ambiente de serviço por meio da interface da web.

Para atualizar um ambiente de serviço

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Ambientes.
3. Escolha a guia Ambiente de serviço.
4. Escolha o ambiente de serviço a ser atualizado.
5. Escolha Ações e, em seguida, escolha uma das opções abaixo:
 - Estado - Escolha Ativar ou Desativar para alterar o estado.
 - Limite de capacidade - modifique o Número máximo de instâncias
6. Escolha Salvar alterações para aplicar as alterações.

O ambiente de serviço é atualizado imediatamente. Verifique os detalhes do ambiente para confirmar se as alterações foram aplicadas com sucesso. Se você desativou o ambiente de

serviço, as filas de trabalhos associadas deixarão de processar novos envios de trabalhos de serviço até que o ambiente seja reativado.

Update a service environment (AWS CLI)

Use o `update-service-environment` comando para modificar um ambiente de serviço com a AWS CLI.

Para atualizar os limites de capacidade do ambiente de serviço

1. Atualizar o limite de capacidade para um ambiente de serviço:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=20
```

2. Verificar se a atualização foi aplicada com sucesso:

```
aws batch describe-service-environments \
  --service-environments my-sagemaker-service-env
```

Para atualizar o estado do ambiente de serviço

1. Desativar um ambiente de serviço para interromper o processamento de novos trabalhos:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --state DISABLED
```

2. Reativar um ambiente de serviço para retomar o processamento:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
  --state ENABLED
```

As atualizações do ambiente de serviço entram em vigor imediatamente. Monitore o estado do ambiente de serviço para garantir que as atualizações sejam concluídas com sucesso antes de enviar novos trabalhos.

Excluir um ambiente de serviço no AWS Batch

Você pode excluir um ambiente de serviço quando ele não for mais necessário para seus trabalhos SageMaker de treinamento. A exclusão de um ambiente de serviço remove a configuração e impede novos envios de trabalhos. Antes de excluir um ambiente de serviço, certifique-se de que nenhum trabalho de SageMaker treinamento ativo dependa dele e que nenhuma fila de trabalhos esteja associada ao ambiente de serviço.

Important

A exclusão do ambiente de serviço é irreversível. Depois da exclusão não será possível recuperar o ambiente de serviço ou sua configuração. Se você precisar de uma funcionalidade semelhante no futuro, deverá criar um novo ambiente de serviço com as configurações necessárias. Considere desativar o ambiente de serviço em vez de excluí-lo se precisar reativá-lo posteriormente.

Note

A exclusão de todos os ambientes de serviço em sua conta não remove automaticamente a função vinculada ao serviço criada para AWS Batch a integração de IA. SageMaker O perfil vinculado ao serviço permanece disponível para a criação futura do ambiente de serviço. Se quiser remover o perfil vinculado ao serviço, você deve excluí-lo separadamente usando a IAM depois de garantir que não existam ambientes de serviço em sua conta.

Pré-requisitos de exclusão

Antes de excluir um ambiente de serviço, você deve desassociar qualquer fila de trabalhos de serviço e, em seguida, desabilitar o ambiente de serviço.

Antes de excluir um ambiente de serviço:

- Verifique as tarefas ativas - Certifique-se de que nenhuma SageMaker tarefa de treinamento esteja sendo executada atualmente no ambiente de serviço.
- Revise as filas de trabalhos - identifique as filas de trabalhos associadas ao ambiente de serviço e associe a fila de trabalhos a um ambiente de serviço diferente ou desative e exclua a fila de trabalhos.

Gerenciamento de filas de trabalhos: as filas de trabalhos associadas a um ambiente de serviço excluído ainda podem existir, mas não podem processar trabalhos de serviço. Você deve excluir filas de trabalhos não utilizadas ou associá-las a um ambiente de serviço diferente antes de excluir o ambiente de serviço original.

Delete a service environment (AWS Console)

Use o AWS Batch console para excluir um ambiente de serviço por meio da interface da web.

Para excluir um ambiente de serviço

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Ambientes.
3. Escolha a guia Ambiente de serviço e escolha um ambiente de serviço.
4. Se o ambiente de serviço estiver ativado, escolha Ações e, em seguida, Desativar.
5. Depois que o ambiente de serviço estiver desativado, escolha Ações e, em seguida, Excluir.
6. Na caixa de diálogo de confirmação, escolha Confirmar.

O ambiente do serviço mostra um estado DELETING enquanto a exclusão ocorre. Quando a exclusão for concluída, o ambiente de serviço desaparecerá da lista de Ambientes.

Delete a service environment (AWS CLI)

Use o `delete-service-environment` comando para remover um ambiente de serviço com a AWS CLI.

Para excluir um ambiente de serviço

1. Verifique as filas de trabalhos associadas ao ambiente de serviço:

```
aws batch describe-job-queues
```

Se houver filas de trabalhos associadas ao ambiente de serviço, você poderá [desassociar a fila de trabalhos](#) do ambiente de serviço e associá-la a um ambiente de serviço diferente ou excluir a fila de trabalhos.

2. Desabilitar o ambiente de serviço:

```
aws batch update-service-environment \
  --service-environment my-sagemaker-service-env \
```

```
--state DISABLED
```

3. Excluir o ambiente de serviço:

```
aws batch delete-service-environment \  
  --service-environment my-sagemaker-service-env
```

4. Monitorar o processo de exclusão:

```
aws batch describe-service-environments \  
  --service-environment my-sagemaker-service-env
```

O ambiente do serviço muda para um estado DELETING durante o processo de exclusão. Depois que a exclusão for concluída, o ambiente de serviço não será mais listado nas operações de descrição. As filas de trabalhos associadas permanecem, mas não podem processar trabalhos de serviço até serem associadas a um ambiente de serviço diferente.

Filas de tarefas

Tarefas são enviadas para uma fila, onde permanecem até que sejam programadas para execução em um ambiente de computação. Uma AWS conta pode ter várias filas de trabalho. Por exemplo, você pode criar uma fila com instâncias sob demanda Amazon EC2 para tarefas de alta prioridade e outra com instâncias spot Amazon EC2 para tarefas de menor prioridade. As filas de tarefas têm uma prioridade usada pelo programador para determinar quais tarefas em quais filas devem ser avaliadas para execução em primeiro lugar.

Tópicos

- [Crie uma fila de trabalhos](#)
- [Exibir uma fila de trabalhos em AWS Batch](#)
- [Excluir uma fila de trabalhos no AWS Batch](#)
- [Políticas de agendamento de compartilhamento justo](#)
- [Agendamento com reconhecimento de recurso](#)

Crie uma fila de trabalhos

Antes de enviar trabalhos em AWS Batch, você precisa criar uma fila. Ao criar uma fila de tarefas, você associa um ou mais ambientes de computação à fila e lhes atribui uma ordem de preferência.

Você também pode definir uma prioridade para a fila de tarefas para determinar a ordem na qual o programador AWS Batch irá posicionar as tarefas. Isso quer dizer que, se um ambiente de computação for associado a mais de uma fila de tarefas, aquela com prioridade mais alta terá preferência.

Tópicos

- [Criar uma fila de trabalhos Amazon EC2](#)
- [Criar uma fila de trabalhos Fargate](#)
- [Criar uma fila de trabalhos Amazon EKS](#)
- [Criar uma fila de trabalhos do SageMaker Training no AWS Batch](#)
- [Modelo de fila de trabalhos](#)

Criar uma fila de trabalhos Amazon EC2

Conclua as etapas a seguir para criar uma fila de trabalhos para o Amazon Elastic Compute Cloud (Amazon EC2).

Para criar uma fila de tarefas Amazon EC2

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Filas de Tarefas.
4. Escolha Criar.
5. Para o Tipo de orquestração, escolha Amazon Elastic Compute Cloud (Amazon EC2).
6. Para Nome, insira um nome exclusivo para a sua fila de tarefas. Ele pode ter até 128 caracteres e conter letras minúsculas, maiúsculas, números e sublinhados (_).
7. Em Prioridade, insira um valor inteiro para a prioridade da fila de tarefas. As filas com prioridade mais alta serão executadas antes das filas com prioridade mais baixa associadas ao mesmo ambiente de computação. A prioridade é determinada em ordem decrescente. Por exemplo, uma fila de tarefas com valor de prioridade 10 tem preferência de programação em relação a uma fila com valor de prioridade 1.
8. (Opcional) Para Política de Agendamento do Nome do Recurso da Amazon (ARN), escolha uma política de agendamento existente.
9. Em Ambientes de Computação Conectados, selecione um ou mais ambientes de computação da lista para associar à lista de tarefas. Selecione ambientes de computação na ordem que quiser que a fila tente posicionar a fila de tarefas. O programador de trabalhos usará a ordem do ambiente de computação para determinar o ambiente de computação responsável por iniciar uma tarefa específica. Antes de poder associá-los a uma fila de tarefas, os ambientes de computação devem estar no estado VALID. Você pode associar até três ambientes de computação a uma fila de tarefas. Caso não possua um ambiente de computação existente, escolha Criar Ambiente Computacional

Note

Todos os ambientes de computação associados a uma fila devem compartilhar o mesmo modelo de provisionamento. AWS Batch não é compatível com a mistura de modelos de provisionamento em uma única fila de tarefas.

10. Em Computar Ordem de Provisionamento, escolha as setas para cima e para baixo para configurar a ordem desejada.
11. Escolha Criar Fila de Tarefas para concluir e criar sua fila.

Criar uma fila de trabalhos Fargate

Conclua as etapas a seguir para criar uma fila de trabalho para o AWS Fargate.

Para criar uma fila de tarefas

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Filas de Tarefas.
4. Escolha Criar.
5. Para Tipo de Orquestração, escolha Fargate.
6. Para Nome, insira um nome exclusivo para a sua fila de tarefas. Ele pode ter até 128 caracteres e conter letras minúsculas, maiúsculas, números e sublinhados (_).
7. Em Prioridade, insira um valor inteiro para a prioridade da fila de tarefas. As filas com prioridade mais alta serão executadas antes das filas com prioridade mais baixa associadas ao mesmo ambiente de computação. A prioridade é determinada em ordem decrescente. Por exemplo, uma fila de tarefas com valor de prioridade 10 tem preferência de programação em relação a uma fila com valor de prioridade 1.
8. (Opcional) Para Política de Agendamento do Nome do Recurso da Amazon (ARN), escolha uma política de agendamento existente.
9. Em Ambientes de Computação Conectados, selecione um ou mais ambientes de computação da lista para associar à lista de tarefas. Selecione ambientes de computação na ordem que quiser que a fila tente posicionar a fila de tarefas. O programador de trabalhos usará a ordem do ambiente de computação para determinar o ambiente de computação responsável por iniciar uma tarefa específica. Antes de poder associá-los a uma fila de tarefas, os ambientes de computação devem estar no estado VALID. Você pode associar até três ambientes de computação a uma fila de tarefas.

 Note

Todos os ambientes de computação associados a uma fila devem compartilhar o mesmo modelo de provisionamento. AWS Batch não é compatível com a mistura de modelos de provisionamento em uma única fila de tarefas.

10. Em Computar Ordem de Provisionamento, escolha as setas para cima e para baixo para configurar a ordem desejada.
11. Escolha Criar Fila de Tarefas para concluir e criar sua fila.

Criar uma fila de trabalhos Amazon EKS

Conclua as etapas a seguir para criar uma fila de trabalho para o Amazon Elastic Kubernetes Service (Amazon EKS).

Para criar uma fila de trabalhos Amazon EKS

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Filas de Tarefas.
4. Escolha Criar.
5. Em Tipo de Orquestração, escolha Amazon Elastic Kubernetes Service (Amazon EKS).
6. Para Nome, insira um nome exclusivo para a sua fila de tarefas. Ele pode ter até 128 caracteres e conter letras minúsculas, maiúsculas, números e sublinhados (_).
7. Para Prioridade, insira um valor inteiro para a prioridade da fila de tarefas. As filas com prioridade mais alta serão executadas antes das filas com prioridade mais baixa associadas ao mesmo ambiente de computação. A prioridade é determinada em ordem decrescente. Por exemplo, uma fila de tarefas com valor de prioridade 10 tem preferência de programação em relação a uma fila com valor de prioridade 1.
8. (Opcional) Para Política de Agendamento do Nome do Recurso da Amazon (ARN), escolha uma política de agendamento existente.
9. Em Ambientes de Computação Conectados, selecione um ou mais ambientes de computação da lista para associar à lista de tarefas. Selecione ambientes de computação na ordem que quiser que a fila tente posicionar a fila de tarefas. O programador de trabalhos usará a ordem

do ambiente de computação para determinar o ambiente de computação responsável por iniciar uma tarefa específica. Antes de poder associá-los a uma fila de tarefas, os ambientes de computação devem estar no estado `VALID`. Você pode associar até três ambientes de computação a uma fila de tarefas.

 Note

Todos os ambientes de computação associados a uma fila devem compartilhar o mesmo modelo de provisionamento. AWS Batch não é compatível com a mistura de modelos de provisionamento em uma única fila de tarefas.

 Note

Todos os ambientes de computação associados a uma fila de trabalhos devem compartilhar a mesma arquitetura. AWS Batch não oferece suporte à mistura de tipos de arquitetura de ambiente de computação em uma única fila de tarefas.

10. Em Computar Ordem de Provisionamento, escolha as setas para cima e para baixo para configurar a ordem desejada.
11. Escolha Criar Fila de Tarefas para concluir e criar sua fila.

Criar uma fila de trabalhos do SageMaker Training no AWS Batch

As filas de trabalhos do SageMaker Training se integram diretamente ao serviço SageMaker AI para fornecer agendamento de tarefas em tecnologia sem servidor, sem exigir que você gerencie a infraestrutura computacional subjacente.

Pré-requisitos

Antes de criar uma lista de trabalhos do SageMaker Training, certifique-se de ter:

- Ambiente de serviço — Um ambiente de serviço que define limites de capacidade. Para obter mais informações, consulte [Criar um ambiente de serviço no AWS Batch](#).
- Permissões do IAM — Permissões para criar e gerenciar filas de trabalhos AWS Batch e ambientes de serviço. Para obter mais informações, consulte [AWS Batch Políticas, funções e permissões do IAM](#).

Create a SageMaker Training job queue (AWS Batch console)

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Filas de trabalho e Criar.
3. Em Tipo de orquestração, escolha SageMaker Training.
4. Em Configuração da fila de trabalhos:
 - a. Em Nome, insira o nome da fila de trabalho.
 - b. Em Prioridade, insira um valor entre 0 e 1000. Uma fila de trabalhos com maior prioridade tem preferência para ambientes de serviço.
 - c. (Opcional) Para Política de Agendamento do Nome do Recurso da Amazon (ARN), escolha uma política de agendamento existente.
 - d. Para Ambientes de serviços conectados, selecione um ambiente de serviço da lista para associar à lista de tarefas.
5. (Opcional) Em Limites de estado de trabalho:
 - a. Em Configuração incorreta, escolha SERVICE_ENVIRONMENT_MAX_RESOURCE e insira o Tempo máximo de execução (segundos) .
 - b. Em Capacidade, escolha INSUFFICIENT_INSTANCE_CAPACITY e insira o Tempo máximo de execução (segundos).
6. Selecione Criar fila de trabalho.

Create a SageMaker Training job queue (AWS CLI)

Use o comando `create-job-queue` para criar uma fila de trabalhos do SageMaker Training.

O exemplo a seguir cria uma fila de trabalhos básica do SageMaker Training que usa um ambiente de serviço:

```
aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
  --priority 1 \
  --service-environment-order order=1,serviceEnvironment=ExampleServiceEnvironment
```

Substitua *ExampleServiceEnvironment* pelo nome do seu ambiente de serviço.

Esse comando retorna uma saída semelhante à seguinte:

```
{
  "jobQueueName": "my-sm-training-fifo-jq",
  "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq"
}
```

Depois de criar sua fila de trabalhos, verifique se ela foi criada com sucesso e se está em um estado válido.

Use o comando `describe-job-queues` para visualizar detalhes sobre a lista de trabalhos:

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

Esse comando retorna uma saída semelhante à seguinte:

```
{
  "jobQueues": [
    {
      "jobQueueName": "my-sm-training-fifo-jq",
      "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq",
      "state": "ENABLED",
      "status": "VALID",
      "statusReason": "JobQueue Healthy",
      "priority": 1,
      "computeEnvironmentOrder": [],
      "serviceEnvironmentOrder": [
        {
          "order": 1,
          "serviceEnvironment": "arn:aws:batch:region:account:service-environment/ExampleServiceEnvironment"
        }
      ],
      "jobQueueType": "SAGEMAKER_TRAINING",
      "tags": {},
      "jobStateTimeLimitActions": []
    }
  ]
}
```

Verifique se:

- O `state` é `ENABLED`

- O status é VALID
- O statusReason é JobQueue Healthy
- O jobQueueType é SAGEMAKER_TRAINING
- serviceEnvironmentOrder faz referência ao seu ambiente de serviço

Modelo de fila de trabalhos

O seguinte é um modelo de fila de tarefas vazio. Você pode usar esse modelo para criar sua fila de trabalhos. Em seguida, você pode salvar essa fila de trabalhos em um arquivo e usá-la com a AWS CLI `--cli-input-json` opção. Para obter mais informações sobre esses parâmetros, consulte [CreateJobQueue](#) no AWS Batch API Reference.

Note

Você pode gerar o modelo de fila de trabalhos com o comando da AWS CLI a seguir.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

```
{
  "computeEnvironmentOrder": [
    {
      "computeEnvironment": "",
      "order": 0
    }
  ],
  "jobQueueName": "",
  "jobStateTimeLimitActions": [
    {
      "state": "RUNNABLE",
      "action": "CANCEL",
      "maxTimeSeconds": 0,
      "reason": ""
    }
  ],
  "priority": 0,
  "schedulingPolicyArn": "",
```

```
"state": "ENABLED",
"tags": {
  "KeyName": ""
}
```

Exibir uma fila de trabalhos em AWS Batch

Depois de criar uma fila de trabalho e enviar trabalhos, é importante poder monitorar o progresso. É possível usar a página Detalhes do trabalho para revisar, gerenciar e monitorar a fila de trabalhos.

Visualizar informações sobre a fila de trabalhos

No AWS Batch console, selecione Filas de trabalhos no painel de navegação e escolha a fila de trabalhos desejada para ver seus detalhes. Nessa página, é possível revisar e gerenciar a fila de trabalhos e ver informações adicionais sobre as operações da fila, como o snapshot da fila de trabalhos, limites de estado do trabalho, a ordem do ambiente, etiquetas e o código JSON da fila de trabalhos.

Detalhes da fila de trabalhos

Esta seção fornece uma visão geral e as opções de manutenção para a fila de trabalhos.

Observação: é possível encontrar o nome do recurso da Amazon (ARN) nesta seção.

Para encontrar essas informações por meio do AWS Command Line Interface, use a [DescribeJobQueues](#) operação junto com o nome da fila de trabalhos ou o ARN correspondente.

Ações ativas

Para filas de trabalho usando agendamento de compartilhamento justo, AWS Batch fornece visibilidade de como diferentes identificadores de compartilhamento consomem capacidade. Essas informações ajudam você a entender a distribuição de recursos e identificar os compartilhamentos que podem precisar de ajustes.

Note

A guia Compartilhamentos ativos só está presente quando o algoritmo de agendamento da fila de trabalhos é Fair-share.

A seção dos 20 principais compartilhamentos ativos mostra identificadores de compartilhamento cujos trabalhos estão programados, iniciais e em execução. Essa visualização inclui:

- Nome do identificador de compartilhamento - O identificador exclusivo do compartilhamento.

Os identificadores de compartilhamento são rótulos que agrupam trabalhos para agendamento de compartilhamento justo. Quando você envia trabalhos com o mesmo identificador de compartilhamento, AWS Batch trata-os como parte da mesma carga de trabalho para fins de alocação de recursos. Os identificadores de compartilhamento ajudam a garantir a distribuição equitativa da capacidade computacional entre diferentes equipes, projetos ou tipos de carga de trabalho. Para obter mais informações, consulte [Usar identificadores de compartilhamento para identificar workloads](#).

- Utilização da capacidade - A quantidade de recursos que a tarefa está configurada para usar. Isso é medido em `instancescpu`, ou vCPU dependendo do ambiente.
- Exibir ação de vagas - Um link para ver todas as vagas desse compartilhamento.

Você pode visualizar essas informações nos formatos de lista e gráfico:

- Visualização de lista - Exibição tabular com números exatos de capacidade
- Visualização do gráfico - Gráfico de barras visual mostrando a utilização relativa

Snapshot da fila de trabalhos

Esta seção fornece uma lista estática dos primeiros 100 RUNNABLE trabalhos que estão na fila. É possível usar o campo de pesquisa para restringir a lista, procurando informações em qualquer coluna da seção de resultados. Os trabalhos na área de resultados do snapshot são classificados com base na estratégia de execução da fila de trabalhos. Para filas de trabalhos first-in-first-out (FIFO), a ordem dos trabalhos é baseada no tempo de envio. Para as filas de trabalho de [agendamento de compartilhamento justo](#), a ordem dos trabalhos se baseia na prioridade do trabalho e no uso do compartilhamento. O campo Capacidade necessária mostra a quantidade de recursos que a tarefa está configurada para usar.

Como os resultados são um snapshot da fila de trabalhos, a lista não é atualizada automaticamente. Para atualizá-la, selecione Atualizar na parte superior da seção. Selecione o hiperlink do nome do trabalho para navegar até Detalhes do trabalho e visualizar o status do trabalho e outras informações relacionadas.

Para encontrar essas informações por meio do AWS CLI, use a [GetJobQueueSnapshot](#) operação junto com o nome da fila de trabalhos ou o ARN correspondente.

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Limites de estados de trabalhos

Use essa guia para rever as informações de configuração relacionadas a quanto tempo um trabalho pode permanecer em um estado RUNNABLE antes de ser cancelado.

Para encontrar essas informações por meio do AWS CLI, use a [DescribeJobQueues](#) operação junto com o nome da fila de trabalhos ou o ARN correspondente.

Ordem do ambiente

Se a sua fila de tarefas é executada em vários ambientes, essa guia fornece a ordem e uma visão geral deles.

Para encontrar essas informações por meio do AWS CLI, use a [DescribeJobQueues](#) operação junto com o nome da fila de trabalhos ou o ARN correspondente.

Tags

Use essa guia para rever e gerenciar as etiquetas associadas a essa fila de trabalho.

JSON

Use essa guia para copiar o código JSON que está associado a essa fila de trabalho. Em seguida, você pode reutilizar o JSON para AWS CloudFormation modelos e AWS CLI scripts.

Excluir uma fila de trabalhos no AWS Batch

Quando você não precisar mais da fila de trabalhos, ela poderá ser desativada e excluída.

Delete a job queue (AWS Batch console)

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação, escolha Filas de trabalhos e, depois, uma fila de trabalhos.
3. Escolha Ações e, em seguida, Desativar.

4. Quando o estado da fila de trabalhos estiver Desativado, escolha Ações e, em seguida, Excluir.
5. Na janela modal, escolha Excluir fila de trabalhos.

Delete a job queue (AWS CLI)

1. Desative a fila de trabalhos para evitar novos envios de trabalhos:

```
aws batch update-job-queue \  
  --job-queue my-sm-training-fifo-jq \  
  --state DISABLED
```

2. Aguarde a conclusão de todos os trabalhos em execução e, em seguida, exclua a fila de trabalhos:

```
aws batch delete-job-queue \  
  --job-queue my-sm-training-fifo-jq
```

Políticas de agendamento de compartilhamento justo

O programador do AWS Batch avalia quando, onde e como executar trabalhos que são enviados a uma fila de trabalhos. Se você não especificar uma política de agendamento ao criar uma fila de trabalhos, o agendador de trabalhos do AWS Batch usará como padrão uma estratégia FIFO (por ordem de entrada). Uma estratégia FIFO (por ordem de chegada) pode fazer com que trabalhos importantes fiquem “presos” atrás de trabalhos que foram enviados anteriormente. Ao especificar uma política de agendamento diferente, você pode alocar recursos de computação de acordo com suas necessidades específicas.

Note

Se você quiser programar a ordem específica em que os trabalhos são executados, use o parâmetro [dependsOn](#) em [SubmitJob](#) para especificar as dependências de cada trabalho.

Se você criar uma política de agendamento e anexá-la a uma fila de trabalhos, o agendamento de compartilhamento justo será ativado. Se a fila de trabalhos tiver uma política de agendamento, a política de agendamento determinará a ordem em que os trabalhos serão executados. Para obter

mais informações, consulte [Usar políticas de agendamento de compartilhamento justo para atribuir identificadores de compartilhamento](#).

Tópicos

- [Usar identificadores de compartilhamento para identificar workloads](#)
- [Usar políticas de agendamento de compartilhamento justo para atribuir identificadores de compartilhamento](#)
- [Usar o agendamento de compartilhamento justo para ajudar a agendar trabalhos](#)
- [Tutorial: criar uma política de agendamento](#)
- [Referência: modelo de política de agendamento](#)

Usar identificadores de compartilhamento para identificar workloads

Você pode usar identificadores de compartilhamento para marcar trabalhos e diferenciar usuários e workloads. O agendador do AWS Batch rastreia o uso de cada identificador de compartilhamento justo usando a fórmula ($T * weightFactor$), onde T é o uso da vCPU ao longo do tempo. O agendador seleciona trabalhos com o menor uso do identificador de compartilhamento. Você pode usar um identificador de compartilhamento sem substituí-lo.

Note

Os identificadores de compartilhamento são exclusivos em uma fila de trabalhos e não são agregados nas filas de trabalhos.

Você pode definir a prioridade do agendamento de compartilhamento justo para configurar a ordem em que os trabalhos são executados em um identificador de compartilhamento. Trabalhos com prioridade de agendamento mais alta são agendados primeiro. Se você não especificar uma política de agendamento de compartilhamento justo, todos os trabalhos enviados à fila de trabalhos serão agendados na ordem FIFO. Ao enviar um trabalho, você não pode especificar um identificador de compartilhamento ou a prioridade de agendamento do compartilhamento justo.

Note

Os recursos de computação anexados são alocados igualmente entre todos os identificadores de compartilhamento, a menos que sejam explicitamente substituídos.

Usar políticas de agendamento de compartilhamento justo para atribuir identificadores de compartilhamento

Você pode usar políticas de agendamento para configurar como os recursos de computação em uma fila de trabalhos serão alocados entre usuários ou workloads. Por meio das políticas de agendamento de compartilhamento justo, você pode atribuir diferentes identificadores de compartilhamento a workloads ou usuários. O AWS Batch atribui a cada identificador de compartilhamento uma percentagem do total de recursos disponíveis durante um período de tempo.

A percentagem de compartilhamento justo é calculada usando os valores `shareDecaySeconds` e `shareDistribution`. Você pode adicionar tempo à análise de compartilhamento justo atribuindo um tempo de decaimento à política. Adicionar tempo dá mais peso ao tempo e menos ao peso definido. Você pode reservar recursos de computação para identificadores de compartilhamento que não estejam ativos especificando uma reserva de computação. Para obter mais informações, consulte [SchedulingPolicyDetail](#).

Usar o agendamento de compartilhamento justo para ajudar a agendar trabalhos

O agendamento de compartilhamento justo fornece um conjunto de controles para ajudar a agendar trabalhos.

Note

Para obter mais informações sobre parâmetros de política de agendamento, consulte [SchedulingPolicyDetail](#).

- Segundos de degradação de compartilhamento - o período de tempo (em segundos) que o agendador do AWS Batch usa para calcular uma porcentagem de compartilhamento para cada identificador de compartilhamento. Um valor zero indica que somente o uso atual é medido. Um tempo de degradação mais longo dá mais peso ao tempo.

Note

O período de tempo de degradação é calculado como: `shareDecaySeconds + OrderMinutes`, onde `OrderMinutes` é o tempo na ordem em minutos.

- Reserva de computação: impede que trabalhos em um único identificador de compartilhamento usem todos os recursos anexados à fila de trabalhos. A proporção reservada é $(computeReservation/100)^{ActiveFairShares}$ onde *ActiveFairShares* é o número de identificadores de compartilhamento ativos.

Note

Se um identificador de compartilhamento tiver trabalhos em um estado SUBMITTED, PENDING, RUNNABLE, STARTING ou RUNNING, ele será considerado um identificador de compartilhamento ativo. Após o período de degradação expirar, um identificador de compartilhamento é considerado inativo.

- Fator de ponderação: o fator de ponderação para o identificador de compartilhamento justo. O valor padrão é 1. Um valor menor permite que trabalhos do identificador de compartilhamento sejam executados ou forneça runtime adicional ao identificador de compartilhamento. Por exemplo, trabalhos que usam um identificador de compartilhamento com um fator de ponderação de 0,125 (1/8) obtêm oito vezes os recursos de computação dos trabalhos que usam um identificador de compartilhamento com um fator de ponderação 1.

Note

Você só precisa definir esse atributo quando precisar atualizar o fator de peso padrão de 1.

Quando a fila de trabalhos estiver ativa e processando trabalhos, é possível revisar uma lista dos 100 primeiros trabalhos RUNNABLE por meio do snapshot da Fila de trabalhos. Para obter mais informações, consulte [Exibir uma fila de trabalhos em AWS Batch](#).

Tutorial: criar uma política de agendamento

Antes de criar uma fila de tarefas com uma política de agendamento, você precisa criar uma política de agendamento. Ao criar uma política de agendamento justo, você associa um ou mais identificadores de compartilhamento, ou prefixos de identificadores de compartilhamento, com pesos para a fila e, opcionalmente, atribui um período de decaimento, além de calcular a reserva para a política.

Para criar uma política de agendamento

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.

2. Na barra de navegação, selecione a Região a ser usada.
3. No painel de navegação, escolha Políticas de Agendamento, Criar.
4. Em Nome, insira um nome exclusivo para a sua política de agendamento. São permitidos até 128 caracteres (caixa alta e baixa), números, hifens e sublinhados.
5. (Opcional) Para Segundos de decaimento de compartilhamento, insira um valor inteiro para o tempo de decaimento de compartilhamento da política de agendamento de compartilhamento justo. Um tempo maior de decaimento de compartilhamentos usará consideravelmente mais recursos de computação por um período maior ao agendar trabalhos. Isso permitirá que trabalhos utilizando um identificador de compartilhamento utilizem mais recursos de computação temporariamente do que o peso do identificador de compartilhamento permitiria, caso não estivesse usando recursos de computação recentes.
6. (Opcional) Em Reserva de computação, insira um valor inteiro para reserva de computação da política de agendamento de compartilhamento justo. A reserva de computação irá separar alguns vCPUs para uso por identificadores de compartilhamento justo não ativos no momento.

A proporção reservada é $(computeReservation/100)^{ActiveFairShares}$, onde *ActiveFairShares* é o número de identificadores de compartilhamento justo ativos.

Por exemplo, um valor *computeReservation* de 50 indica que AWS Batch deve reservar 50% da VCPU máxima disponível caso haja apenas um identificador de compartilhamento justo, 25% em caso de dois identificadores e 12,5% em caso de três identificadores de compartilhamento justo. Um valor *computeReservation* de 25 indica que AWS Batch deve reservar 25% da vCPU máxima disponível caso haja apenas um identificador de compartilhamento, 6,25% em caso de dois identificadores de compartilhamento e 1,56% caso haja três identificadores de compartilhamento.

7. Na seção Atributos de compartilhamento, é possível especificar o identificador de compartilhamento e peso de cada identificador de compartilhamento associado à política de agendamento.
 - a. Escolha Adicionar Identificador de Compartilhamento.
 - b. Para Identificador de compartilhamento, especifique o identificador de compartilhamento. Se a string terminar com "*", isso torna-se um prefixo de identificador de compartilhamento para equiparar identificadores de compartilhamento para trabalhos. Todos os identificadores de compartilhamento e prefixos de identificador de compartilhamento em uma política de agendamento devem ser exclusivos e não podem se sobrepor. Por exemplo, você não

- pode ter identificadores de compartilhamento justo com prefixo “UserA*” e identificador de compartilhamento “UserA1” na mesma política de agendamento de compartilhamento justo.
- c. Para o Fator de peso, especifique o peso relativo para o identificador de compartilhamento. O valor padrão é 1.0. Um valor mais baixo terá prioridade mais alta para recursos de computação. Se um prefixo de identificador de compartilhamento for usado, trabalhos com identificadores de compartilhamento que comecem com o prefixo compartilharão o fator peso. Isso aumenta efetivamente o fator peso desses trabalhos, enquanto diminui sua prioridade individual e mantém o mesmo fator para o prefixo do identificador de compartilhamento.
8. (Opcional) Na seção Tags, você pode especificar a chave e o valor de cada tag associada à política de agendamento. Para mais informações, consulte [Marcar com tag os recursos do AWS Batch](#).
 9. Escolha Enviar para finalizar e criar sua política de agendamento.

Referência: modelo de política de agendamento

Um modelo de política de agendamento vazio é mostrado abaixo. É possível usar esse modelo para criar sua política de agendamento, que pode ser salva em um arquivo e usada com a opção `--cli-input-json` da AWS CLI. Para obter mais informações sobre esses parâmetros, consulte [CreateSchedulingPolicy](#) na AWS Batch API Reference.

Note

Você pode gerar o modelo de fila de trabalhos com o comando da AWS CLI a seguir.

```
$ aws batch create-scheduling-policy --generate-cli-skeleton
```

```
{
  "name": "",
  "fairsharePolicy": {
    "shareDecaySeconds": 0,
    "computeReservation": 0,
    "shareDistribution": [
      {
        "shareIdentifier": "",
        "weightFactor": 0.0
      }
    ]
  }
}
```

```
    }  
  ]  
},  
"tags": {  
  "KeyName": ""  
}  
}
```

Agendamento com reconhecimento de recurso

AWS Batch agenda trabalhos com base na vCPU, na GPU e na disponibilidade de memória no Ambiente Computacional (CE) associado à Fila de trabalho (JQ). Mas, às vezes, apenas a disponibilidade desses recursos de CE não garante que o trabalho seja executado com sucesso, pois pode depender de outros recursos necessários, portanto, esses trabalhos são cancelados ou encerrados. Isso resulta no uso ineficiente dos recursos computacionais. Para resolver esse problema, o agendamento com reconhecimento de recursos pode verificar a disponibilidade de recursos dependentes que não sejam do CE antes de programar o trabalho para execução em um CE.

O agendamento com reconhecimento de recurso do AWS Batch permite que você agende trabalhos com base nos recursos consumíveis necessários para executar seus trabalhos: tokens de licença de terceiros, largura de banda de acesso ao banco de dados, necessidade de limitar as chamadas de controle de utilização de uma API de terceiros e assim por diante. Você especifica os recursos consumíveis necessários para a execução de um trabalho e o Batch leva essas dependências de recursos em consideração ao agendar um trabalho. Você pode evitar intervenções manuais para eliminar falhas no trabalho e longas esperas causadas pela escassez de recursos consumíveis. Você pode reduzir a subutilização dos recursos computacionais alocando somente os trabalhos que têm todos os recursos necessários disponíveis.

O agendamento com reconhecimento de recursos está disponível para políticas de agendamento FIFO e de compartilhamento justo e pode ser usado com todas as plataformas de computação compatíveis com o Batch, incluindo EKS, ECS e Fargate. Ele pode ser usado com trabalhos de matriz, trabalhos de paralelos de vários nós (MNP) e trabalhos de Batch regulares.

Para configurar o agendamento com reconhecimento de recursos, primeiro você especifica todos os recursos consumíveis necessários para executar seus trabalhos, junto com a contagem total disponível de cada recurso. Em seguida, para cada trabalho que requer um recurso consumível, você especifica o nome e as quantidades necessárias de cada recurso necessário. O Batch

acompanha quantos recursos consumíveis estão disponíveis para os trabalhos em suas filas de trabalhos e garante que um trabalho seja programado para execução somente quando todos os recursos consumíveis necessários estiverem disponíveis para que o trabalho seja executado com sucesso.

Tópicos

- [Criar recursos consumíveis](#)
- [Especificar os recursos necessários para executar um trabalho](#)
- [Verificar quantos recursos estão em uso e disponíveis](#)
- [Atualizar a quantidade de um recurso enquanto ele é usado em trabalhos](#)
- [Encontrar os trabalhos que exigem um recurso consumível específico](#)
- [Excluir um recurso consumível](#)

Criar recursos consumíveis

Primeiro, você deve criar os recursos consumíveis que representam os recursos não CE que são consumidos quando um trabalho está em execução e estão disponíveis somente em quantidades limitadas. Cada recurso consumível tem o seguinte:

- nome do recurso (`consumableResourceName`) que deve ser exclusivo no nível da conta.
- (opcional) tipo de recurso (`resourceType`) que indica se o recurso está disponível para ser reutilizado após a conclusão de um trabalho. Isto pode ser:
 - REPLENISHABLE (padrão)
 - NON_REPLENISHABLE
- quantidade total (`totalQuantity`) que especifica a quantidade total do recurso consumível disponível.

O número máximo de recursos consumíveis por conta é de 50 mil.

Console do:

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Escolha Criar recurso consumível.
3. Insira um Nome de recurso exclusivo, a Quantidade total de recursos e selecione se o Tipo de recurso é Renovável ou Não renovável.

4. Escolha Criar recurso consumível.

API do:

Use a [API CreateConsumableResource](#) para definir os recursos que você deseja.

Especificar os recursos necessários para executar um trabalho

Ao registrar um trabalho, você pode especificar o nome de um ou mais recursos criados (consumableResource) e a quantidade desse recurso que cada instância do trabalho exige (quantity).

O Batch acompanha as unidades disponíveis de cada recurso a qualquer momento. Para cada trabalho na fila de trabalhos, o agendador Batch garante que seu trabalho seja executado somente quando as dependências de recursos especificadas estiverem disponíveis.

Se um recurso consumível para o trabalho não estiver disponível quando o trabalho chegar ao topo da fila, o trabalho aguardará em estado RUNNABLE até que todos os recursos necessários estejam disponíveis ou que o limite de tempo do estado do trabalho seja atingido (consulte [Exibir uma fila de trabalhos em AWS Batch](#)). Depois que o Batch valida que todos os recursos estão disponíveis, a tarefa passa para o estado STARTING e depois para RUNNING. Os recursos são bloqueados quando o trabalho é transferido para STARTING e, em seguida, são desbloqueados quando o trabalho é transferido para SUCCEEDED ou FAILED.

Você também pode atualizar a quantidade de um recurso necessário para um trabalho específico ao enviar o trabalho.

Console do:

Para especificar recursos e suas quantidades necessárias ao definir um trabalho:

1. Defina um trabalho usando o assistente de definição de trabalhos do [Console AWS Batch](#) (Definições de trabalhos -> Criar).
2. Na Etapa 4: Configurar contêineres do assistente, em Recurso consumível, selecione o Nome de um recurso necessário na lista. No campo Valor solicitado, insira a quantidade desse recurso necessária para uma instância desse trabalho e escolha Adicionar recurso consumível.
3. Repita a etapa anterior para todos os recursos consumíveis exigidos pelo trabalho. É possível especificar até cinco recursos para cada trabalho definido.

4. Você verá uma lista dos recursos consumíveis que você criou depois de concluir o assistente de definição de trabalho, mas antes de escolher Criar definição de trabalho.

Para atualizar as quantidades necessárias de recursos ao enviar um trabalho:

1. No painel de navegação da esquerda do [Console AWS Batch](#), escolha Trabalhos e, em seguida, escolha Enviar novo trabalho.
2. Na Etapa 2: Configurar substituições do assistente, em Substituições de recursos consumíveis, insira um novo Valor solicitado para qualquer recurso consumível cuja quantidade necessária você queira substituir para o trabalho.
3. Depois de concluir todas as substituições que você deseja fazer para esse trabalho, escolha Avançar para continuar para Revisar e enviar.

API do:

Ao registrar um trabalho na [API RegisterJobDefinition](#), use o `consumableResourceList` na parte `consumableResourceProperties` da solicitação para especificar os recursos consumíveis necessários para executar uma instância do trabalho e a quantidade de cada um.

Ao enviar um trabalho com a [API SubmitJob](#), você pode substituir a lista de recursos consumíveis e a quantidade de cada um usando a parte `consumableResourcePropertiesOverride` da solicitação. Observe que isso substitui apenas a quantidade do recurso necessária para cada instância do trabalho, não a quantidade total disponível.

Verificar quantos recursos estão em uso e disponíveis

O Batch permite que você consulte o número de recursos disponíveis (`availableQuantity`), o número de recursos em uso (`inUseQuantity`) e o total de recursos (`totalQuantity`) em um determinado momento.

Quando um trabalho vai para o estado `STARTING`, os recursos consumidos serão subtraídos da quantidade disponível desse recurso. Se o recurso for `REPLENISHABLE`, o número de recursos consumidos será adicionado de volta à quantidade disponível assim que o trabalho for movido para o estado `SUCCEEDED` ou `FAILED`, e a quantidade total permanecerá a mesma. Se o recurso for `NON_REPLENISHABLE`, o número de recursos consumidos será subtraído das quantidades totais e disponíveis e não será adicionado novamente se o trabalho for transferido para o estado `SUCCEEDED` ou `FAILED`.

 Note

Essas informações podem ter um atraso de até 30 segundos.

Console do:

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Selecione a guia Renovável ou Não Renovável para visualizar os recursos desse tipo que você criou.
3. Para cada recurso Renovável, o console exibe o Nome, a quantidade Total do recurso, o número atualmente Em uso e quantos ainda estão Disponíveis, junto com um cálculo da Utilização (o número de recursos em uso dividido pela quantidade total desse recurso).

Para cada recurso Não Renovável, o console exibe o Nome, o número atualmente Em uso e quantos ainda estão Disponíveis.

Você também pode visualizar as informações atuais sobre recursos consumíveis na página de detalhes do trabalho no console.

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Trabalhos e selecione o nome de um trabalho para abrir a página de detalhes desse trabalho.
2. Informações sobre Recursos Renováveis e Recursos não Renováveis estão disponíveis para verificar se são exigidos pelo trabalho. Para ambos os tipos, o console exibe o Nome do recurso, a quantidade Solicitada para o trabalho, quantos ainda estão Disponíveis, o número atualmente Em uso, a quantidade Total do recurso, junto com um cálculo da Utilização atual (o número de recursos em uso pelo trabalho dividido pela quantidade total desse recurso).

API do:

Use a [API DescribeConsumableResource](#), que retorna as seguintes informações:

```
{
  "availableQuantity": number,
  "consumableResourceArn": "string",
  "consumableResourceName": "string",
  "createdAt": number,
  "inUseQuantity": number,
```

```
"resourceType": "string",
"tags": {
  "string" : "string"
},
"totalQuantity": number
}
```

A [API ListConsumableResources](#) também relata o número de recursos em uso (inUseQuantity) e o número total de recursos atualmente disponíveis (totalQuantity) como parte da lista de todos os recursos consumíveis que você criou em sua conta. Essa API também permite filtrar a consulta da lista de recursos consumíveis com base no nome do recurso consumível.

Atualizar a quantidade de um recurso enquanto ele é usado em trabalhos

Você pode redefinir a quantidade total de um recurso para um novo valor, adicionar à quantidade total ou subtrair dela.

Se a nova quantidade total especificada for maior do que antes, o Batch agenda mais trabalhos de acordo com isso. Se a nova quantidade total for menor do que antes e não houver unidades desse recurso em uso, o Batch apenas reduzirá a quantidade total (ou disponível). Se houver unidades em uso, o Batch reduz a quantidade disponível imediatamente e, à medida que os trabalhos são concluídos, o Batch reduz a quantidade total (disponível) para que, eventualmente, chegue ao novo número.

Console do:

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Selecione a guia Renovável ou Não Renovável para visualizar os recursos desse tipo que você criou.
3. Para recursos Renováveis:
 1. Escolha o recurso que você deseja atualizar, selecione Ações e escolha Definir recursos, Adicionar recursos ou Remover recursos.
 2. É exibida uma janela pop-up na qual você pode Definir o valor total, Adicionar recursos ou Remover recursos, dependendo da ação escolhida na etapa anterior. Insira a quantidade que você deseja definir como o novo valor total, que deseja adicionar à quantidade total ou que deseja subtrair da quantidade total e selecione OK.

Para recursos Não renováveis:

1. Escolha o recurso que você deseja atualizar, selecione Ações e escolha Definir recursos, Adicionar recursos ou Remover recursos.
2. É exibida uma janela pop-up na qual você pode Definir o valor disponível, Adicionar recursos ou Remover recursos, dependendo da ação escolhida na etapa anterior. Insira a quantidade que você deseja definir como o novo valor disponível, que deseja adicionar à quantidade disponível ou que deseja subtrair da quantidade disponível e, em seguida, selecione OK.

API do:

Use a [API UpdateConsumableResource](#) para definir uma nova quantidade total para o recurso ou para aumentar ou reduzir a quantidade total.

Encontrar os trabalhos que exigem um recurso consumível específico

O Batch permite recuperar uma lista de trabalhos que exigem um recurso consumível específico.

Console do:

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Na lista, selecione o nome do recurso consumível. A página de detalhes desse recurso é aberta.
3. Em Pesquisar trabalhos, insira os filtros que você deseja aplicar à lista de trabalhos. É possível filtrar por Nome do trabalho (“igual a”, “começa com”), por Intervalo de datas (quando o trabalho foi criado) e por Critérios adicionais (“fila de trabalhos”, “definição do trabalho”, “identificador de trabalho compartilhado”). Para cada tipo de filtro que você deseja aplicar, selecione entre as opções disponíveis na lista suspensa e insira as informações adicionais solicitadas.

Selecione a opção Pesquisar.

4. Uma lista (filtrada) de trabalhos que exigem o recurso consumível é exibida, incluindo o nome do trabalho, status, número de unidades solicitadas do recurso consumível, outros recursos consumíveis necessários e assim por diante. Nessa lista, é possível selecionar um ou mais trabalhos para Cancelar ou Encerrar. Você também pode selecionar o nome de um trabalho para abrir a respectiva página de detalhes.
5. Em Pesquisar trabalhos, agora é possível Atualizar os resultados ou Limpar a pesquisa e começar de novo.

API do:

Você pode obter uma lista de trabalhos que usam um recurso consumível específico com a [API `ListJobsByConsumableResource`](#). Essa API também permite filtrar a consulta da lista de trabalhos usando o status do trabalho ou o nome do trabalho.

Excluir um recurso consumível

Você pode excluir um recurso consumível a qualquer momento, mesmo quando os trabalhos que exigem o recurso ainda estão em execução. Depois que um recurso consumível é excluído, pode haver uma lacuna entre o momento em que o comando de excluir é recebido e quando o agendador de trabalhos honra a exclusão, portanto, é possível que os trabalhos que consomem o recurso sejam agendados logo após a chamada de exclusão. Se o recurso consumível excluído tiver o tipo de recurso (`resourceType`) `REPLENISHABLE`, isso será ignorado quando os trabalhos forem concluídos. Se você excluir um recurso consumível e recriá-lo com o mesmo nome, ele será considerado o mesmo recurso e poderá ser usado por trabalhos `RUNNABLE`.

Console do:

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Selecione a guia Renovável ou Não Renovável para visualizar os recursos desse tipo que você criou.
3. Selecione cada recurso que deseja excluir e escolha Excluir. Uma janela pop-up Excluir recurso consumível aparece. Para confirmar a exclusão, selecione Excluir.

Também é possível excluir um recurso consumível de sua página de detalhes no console.

1. No painel de navegação esquerdo do [Console AWS Batch](#), escolha Recursos consumíveis.
2. Selecione a guia Renovável ou Não Renovável para visualizar os recursos desse tipo que você criou.
3. Escolha o nome do recurso que deseja excluir. A página de detalhes do recurso consumível é exibida. Escolha Excluir. Uma janela pop-up Excluir recurso consumível aparece. Para confirmar a exclusão, selecione Excluir.

API do:

Use a [API `DeleteConsumableResource`](#) para excluir um recurso consumível.

Definições de trabalho

AWS Batch as definições de tarefas especificam como as tarefas devem ser executadas.

Embora cada tarefa deva fazer referência a uma definição de tarefa, muitos parâmetros que são especificados na definição de tarefa podem ser substituídos em runtime.

Alguns dos atributos especificados em uma definição de tarefa incluem:

- Qual imagem de docker usar com o contêiner em seu trabalho.
- Quantos v CPUs e quanta memória usar com o contêiner.
- O comando que o contêiner deve executar quando iniciado.
- Quais (se houver alguma) variáveis de ambiente devem ser passadas para o contêiner quando iniciado.
- Todos os volumes de dados que devem ser usados com o contêiner.
- Qual função do IAM (se houver) que seu trabalho deve usar para obter AWS permissões.

Conteúdo

- [Criar uma definição de tarefa de nó único](#)
- [Criar uma definição de trabalho paralelo de vários nós](#)
- [Modelo de definição de trabalho que usa ContainerProperties](#)
- [Crie definições de trabalhos usando EcsProperties](#)
- [Usar o driver de log awslogs.](#)
- [Especificar dados confidenciais](#)
- [Autenticação de registro privado para trabalhos](#)
- [Volumes Amazon EFS](#)
- [Exemplo de definição de trabalhos](#)

Criar uma definição de tarefa de nó único

Antes de poder executar trabalhos no AWS Batch, você deve criar uma definição de tarefa. Esse processo varia ligeiramente entre trabalhos em paralelo de nó único e de vários nós. Este tópico aborda especificamente como criar uma definição de trabalho para um trabalho do AWS Batch que não seja um trabalho paralelo de vários nós (também conhecido como programação de grupos).

Você pode criar uma definição de trabalho em paralelo de vários nós nos recursos do Amazon Elastic Container Service. Para obter mais informações, consulte [the section called “Criar uma definição de trabalho paralelo de vários nós”](#).

Tópicos

- [Criar uma definição de trabalho de nó único nos recursos do Amazon EC2](#)
- [Criar uma definição de trabalho de nó único nos recursos do Fargate](#)
- [Criar uma definição de trabalho de nó único nos recursos do Amazon EKS](#)
- [Criar uma definição de trabalho de nó único com vários contêineres nos recursos do Amazon EC2](#)

Criar uma definição de trabalho de nó único nos recursos do Amazon EC2

Conclua as etapas a seguir para criar uma definição de trabalho com um único nó nos recursos do Amazon Elastic Compute Cloud (Amazon EC2).

Para criar uma nova definição de trabalho em recursos do Amazon EC2:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, escolha a Região da AWS a ser usada.
3. No painel de navegação esquerdo, escolha Definições de trabalho.
4. Escolha Criar.
5. Para o Tipo de orquestração, escolha Amazon Elastic Compute Cloud (Amazon EC2).
6. Em Configuração da plataforma EC2, desative o processamento de Ativar paralelo de vários nós.
7. Em Nome, insira um nome exclusivo para a sua definição de trabalho. Os nomes podem ter até 128 caracteres. Podem conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
8. (Opcional) Em Tempo limite de execução, insira o valor do tempo limite (em segundos). O tempo limite de execução é o período de tempo antes que um trabalho não concluído seja encerrado. Se uma tentativa exceder o tempo limite, ela será interrompida e passada para um status FAILED. Para mais informações, consulte [Tempos limite de trabalho](#). O valor mínimo é 60 segundos.
9. (Opcional) Ative Prioridade de agendamento. Insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade.

10. (Opcional) Para Tentativas de Trabalho, insira o número máximo de vezes que AWS Batch tenta mover o trabalho para um status RUNNABLE. Insira um número inteiro entre 1 e 10.
11. (Opcional) Em Repetir condições da estratégia, escolha Adicionar avaliação na saída. Insira pelo menos um valor de parâmetro e escolha uma Ação. Para cada conjunto de condições, Ação deve ser definida como Tentar novamente ou Sair. Essas ações significam o seguinte:
 - Tentar novamente – O AWS Batch tenta novamente até que o número de tentativas de trabalho que você especificou seja atingido.
 - Sair: o AWS Batch para de tentar novamente o trabalho.

 Important

Se você escolher Adicionar avaliação na saída, deverá configurar pelo menos um parâmetro e ou escolher uma Ação ou escolher Remover avaliação na saída.

12. (Opcional) Expanda Tags e então, escolha Adicionar Tag para adicionar tags ao recurso. Insira uma chave e um valor opcional e então escolha Adicionar tag.
13. (Opcional) Ative Propagar tags para propagar tags do trabalho e da definição de trabalho para a tarefa do Amazon ECS.
14. Escolha Próxima página.
15. Na seção Configuração de Contêiner:
 - a. Para Imagem, escolha a imagem do Docker a ser usada em seu trabalho. Por padrão, as imagens no registro do Docker Hub estão disponíveis. Você também pode especificar outros repositórios com `repository-url/image:tag`. Os nomes podem ter até 225 caracteres. Pode conter letras maiúsculas e minúsculas, números, hifens (-), sublinhados (_), dois pontos (:), barras (/) e sinais de número (#). Esse parâmetro é mapeado para Image na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro IMAGE de [docker run](#).

 Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo, imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` ou `registry/repository[@digest]` completa (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - As imagens em repositórios do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` completa (por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Imagens em repositórios oficiais em Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
 - Imagens em outros repositórios Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
 - Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).
- b. Em Comando, insira os comandos no campo como o equivalente da matriz de string JSON.

Esse parâmetro é mapeado para `Cmd` na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro `COMMAND` de [docker run](#). Para obter mais informações sobre o parâmetro Docker CMD, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Você pode usar os valores padrão de substituição de parâmetros e espaços reservados no seu comando. Para obter mais informações, consulte [Parameters](#).

- c. (Opcional) Para Função de execução, especifique um perfil do IAM que conceda aos atendentes de contêiner do Amazon ECS permissão para fazer chamadas da API da AWS em seu nome. Esse atributo usa perfis do IAM do Amazon ECS para tarefas. Para obter mais informações, consulte [Perfis do IAM para execução de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- d. Para a configuração de Perfil de trabalho, escolha um perfil do IAM que tenha permissões para as APIs da AWS. Esse atributo usa perfis do IAM do Amazon ECS para tarefas. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

 Note

Somente funções que tenham o relacionamento de confiança Função da tarefa do Amazon Elastic Container Service são mostradas aqui. Para obter mais informações sobre como criar um perfil do IAM para os trabalhos do seu AWS Batch, consulte [Como criar um perfil do IAM e uma política para suas tarefas](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

16. Em Parâmetros, escolha Adicionar parâmetros para adicionar espaços reservados para substituição de parâmetros como pares de Chave e Valor opcionais.
 17. Na seção Configuração de Ambiente:
 - a. Para vCPUs, especifique o número de vCPUs a serem reservadas para o contêiner. Esse parâmetro é mapeado para CpuShares na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção `--cpu-shares` para [docker run](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Você deve especificar pelo menos uma vCPU.
 - b. Em Memória, insira o limite de memória disponível para o contêiner. Caso tente exceder a quantidade memória especificada aqui, o contêiner será interrompido. Esse parâmetro é mapeado para Memory na seção [Criar um Contêiner](#) da [API Remota Docker](#) e para a opção `--memory` para [docker run](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa.
-  Note
- Para maximizar o uso dos recursos, priorize a memória para trabalhos de um tipo específico de instância. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).
- c. Em Número de GPUs, escolha o número de GPUs a serem reservadas para o contêiner.
 - d. (Opcional) Em Variáveis de Ambiente, escolha Adicionar Variável de Ambiente para adicionar variáveis de ambiente como pares de nome/valor. Essas variáveis serão passadas para o contêiner.
 - e. (Opcional) Em Segredos, escolha Adicionar segredo para adicionar segredos como pares de nome-valor. Esses segredos são expostos no contêiner. Para obter mais informações, consulte [LogConfiguration:secretOptions](#).
18. Escolha Próxima página.

19. Na seção Configuração do Linux:

- a. Para Usuário, insira o nome do usuário a ser usado dentro do contêiner. Esse parâmetro é mapeado para User na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção --user para [docker run](#).
- b. (Opcional) Para oferecer ao contêiner de trabalho permissões elevadas na instância do host (semelhante ao usuário root), arraste o controle deslizante Privilegiado para a direita. Esse parâmetro é mapeado para Privileged na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção --privileged para [docker run](#).
- c. (Opcional) Ative Ativar processo init para executar um processo init dentro do contêiner. Este processo encaminha sinais e colhe processos.

20. (Opcional) Na seção Configuração do sistema de arquivos:

- a. Ative Habilitar sistema de arquivos somente para leitura para remover o acesso de gravação ao volume.
- b. Para Tamanho da memória compartilhada, insira o tamanho (em MiB) do volume /dev/shm.
- c. Em Tamanho máximo de troca, insira a quantidade total de memória de troca (em MiB) que o contêiner pode usar.
- d. Em Swappiness, insira um valor entre 0 e 100 para indicar o comportamento de troca de memória do contêiner. Se você não especificar um valor e a troca estiver ativada, o valor assumirá 60 como padrão. Para obter mais informações, consulte [LinuxParameters:swappiness](#).
- e. (Opcional) Expanda Configuração adicional.
- f. (Opcional) Para Tmpfs, escolha Adicionar tmpfs para adicionar uma montagem tmpfs.
- g. (Opcional) Em Dispositivos, escolha Adicionar dispositivo para adicionar um dispositivo:
 - i. Em Container path (Caminho do contêiner), especifique o caminho na instância de contêiner para expor o dispositivo mapeado para a instância de host. Se isso for deixado em branco, o caminho de host será usado no contêiner.
 - ii. Em Host path, especifique o caminho de um dispositivo na instância de host.
 - iii. Em Permissões, selecione uma ou mais permissões para serem aplicadas ao dispositivo. As permissões disponíveis são LER, GRAVAR e MKNOD.

- h. (Opcional) Para Configuração de volumes, escolha Adicionar volume para criar uma lista de volumes a serem passados para o contêiner. Insira Nome e Caminho de origem para o volume e escolha Adicionar volume. Você também pode optar por ativar o Habilitar EFS.
- i. (Opcional) Em Pontos de montagem, escolha Adicionar configuração de pontos de montagem para adicionar pontos de montagem para volumes de dados. Você deve especificar o volume de origem e o caminho do contêiner. Esses pontos de montagem são passados para o Docker daemon em uma instância de contêiner. Você também pode optar por tornar o volume Somente para leitura.
- j. (Opcional) Para Configuração de ulimits, escolha Adicionar ulimit para adicionar um valor de ulimits ao contêiner. Insira os valores de Nome, Limite flexível, e Limite rígido e, em seguida, escolha Adicionar ulimit.

21. Na seção Propriedades da tarefa:

- a. Em Perfil de execução - condicional, escolha um perfil para permitir que os atendentes do Amazon ECS façam chamadas de API AWS em seu nome. Para obter mais informações sobre criar um Perfil de execução, consulte [Tutorial: criar o perfil de execução do IAM](#).
- b. Escolha Habilitar o comando de execução do ECS para permitir o acesso direto ao shell do contêiner do Amazon ECS e ignorar o sistema operacional do host. Você deve escolher uma Função de tarefa.

 Important

O comando Executar ECS exige que o sistema de arquivos seja gravável.

- c. Em Perfil da tarefa, escolha um perfil do Amazon ECS Identity and Access Management (IAM) que permita que o contêiner faça chamadas de API AWS em seu nome. Para obter mais informações, consulte [Perfil do IAM de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

22. (Opcional) Na seção Configuração de registro:

- a. Em Driver de log, escolha o driver de log a ser usado. Para obter mais informações sobre os drivers de log disponíveis, consulte [LogConfiguration:logDriver](#).

 Note

Por padrão, o driver de log `awslogs` é usado.

- b. Em Opções, escolha Adicionar opção para adicionar uma opção. Insira um par nome-valor e então escolha Adicionar opção.
- c. Em Segredos, escolha Adicionar segredo. Insira um par nome-valor e escolha Adicionar segredo para adicionar um segredo.

**Tip**

Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

- 23. Escolha Próxima página.
- 24. Para a Revisão de Definição de Tarefa, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar Definição de Trabalho.

Criar uma definição de trabalho de nó único nos recursos do Fargate

Conclua as seguintes etapas para criar uma definição de trabalho de nó único em recursos do AWS Fargate.

Para criar uma nova definição de trabalho em recursos do Fargate:

- 1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
- 2. Na barra de navegação superior, escolha a Região da AWS a ser usada.
- 3. No painel de navegação esquerdo, escolha Definições de trabalho.
- 4. Escolha Criar.
- 5. Para Tipo de orquestração, escolha Fargate. Para obter mais informações, consulte [Ambientes de computação Fargate](#).
- 6. Em Nome, insira um nome exclusivo para a sua definição de trabalho. Os nomes podem ter até 128 caracteres. Podem conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
- 7. (Opcional) Em Tempo Limite de Execução, insira o valor do tempo limite (em segundos). O tempo limite de execução é o período de tempo antes que um trabalho não concluído seja encerrado. Se uma tentativa exceder o tempo limite, ela será interrompida e passada para um status FAILED. Para mais informações, consulte [Tempos limite de trabalho](#). O valor mínimo é 60 segundos.

8. (Opcional) Ative a Prioridade de Agendamento. Insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade sobre valores mais baixos.
9. (Opcional) Expanda Tags e então, escolha Adicionar Tag para adicionar tags ao recurso. Ative Propagar tags para propagar tags do trabalho e da definição do trabalho.
10. Na seção Configuração da plataforma Fargate:
 - a. Para a plataforma Runtime, escolha a arquitetura do ambiente de computação.
 - b. Em Família do sistema operacional, escolha o sistema operacional do ambiente de computação.
 - c. Em Arquitetura da CPU, escolha a arquitetura vCPU.
 - d. Para Versão da plataforma Fargate, insira LATEST ou uma versão específica do ambiente de runtime.
 - e. (Opcional) Ative a opção Atribuir IP público para atribuir um endereço IP público a uma interface de rede de tarefas do Fargate. Para um trabalho que é executado em uma sub-rede privada para enviar tráfego de saída para a Internet, a sub-rede privada exige que um gateway NAT seja anexado para rotear solicitações para a Internet. Ao fazer isso, você poderá extrair imagens de contêineres. Para obter mais informações, consulte [Rede de tarefas do Amazon ECS](#) no Guia do desenvolvedor do Serviço Amazon Elastic Container.
 - f. (Opcional) Em Armazenamento temporário, insira a quantidade de armazenamento temporário a ser alocada para a tarefa. A quantidade de armazenamento temporário deve estar entre 21 GiB e 200 GiB. Por padrão, 20 GiB de armazenamento temporário são alocados se você não inserir um valor.

 Note

O armazenamento temporário requer a versão da plataforma Fargate 1.4 ou posterior.

- g. Para Função de execução, especifique um perfil do IAM que conceda aos atendentes de contêiner do Amazon ECS e do Fargate permissão para fazer chamadas da API da AWS em seu nome. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para obter mais informações, incluindo pré-requisitos de configuração, consulte [Perfis do IAM para execução de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- h. Para Tentativas de trabalho, insira o número de vezes que o AWS Batch tenta mover o trabalho para o status de RUNNABLE. Insira um número inteiro entre 1 e 10.

- i. (Opcional) Em Repetir as condições da estratégia, escolha Adicionar avaliação na saída. Insira pelo menos um valor de parâmetro e escolha uma Ação. Para cada conjunto de condições, Ação deve ser definida como Tentar Novamente ou Sair. Essas ações significam o seguinte:
 - Tentar novamente: o AWS Batch tenta novamente até que o número de tentativas de trabalho que você especificou seja atingido.
 - Sair: o AWS Batch para de tentar novamente o trabalho.

 Important

Se você escolher Adicionar avaliação na saída, deverá configurar pelo menos um parâmetro e escolher uma Ação ou escolher Remover avaliação na saída.

11. Escolha Próxima página.

12. Na seção Configuração de Contêiner:

- a. Para Imagem, escolha a imagem do Docker a ser usada em seu trabalho. Por padrão, imagens no registro Docker Hub estarão disponíveis. Você também pode especificar outros repositórios com `repository-url/image:tag`. Os nomes podem ter até 225 caracteres. Pode conter letras maiúsculas, minúsculas, números, hifens (-), sublinhados (_), dois pontos (:), pontos (.), barras (/) e jogos da velha (#). Esse parâmetro é mapeado para Image na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro IMAGE de [docker run](#).

 Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo, imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` ou `registry/repository[@digest]` completa (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).

- As imagens em repositórios do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` completa (por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Imagens em repositórios oficiais em Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
 - Imagens em outros repositórios Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
 - Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).
- b. Em Comando, insira os comandos no campo como o equivalente da matriz de string JSON.

Esse parâmetro é mapeado para `Cmd` na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro `COMMAND` de [docker run](#). Para obter mais informações sobre o parâmetro Docker CMD, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Você pode usar os valores padrão de substituição de parâmetros e espaços reservados no seu comando. Para obter mais informações, consulte [Parameters](#).

- c. (Opcional) Adicione parâmetros à definição do trabalho como mapeamentos de nome-valor para substituir os padrões de definição do trabalho. Para adicionar um parâmetro:
- Em Parâmetros, escolha Adicionar parâmetros, insira um par nome-valor e então escolha Adicionar parâmetro.

 Important

Se você escolher Adicionar parâmetro, deverá configurar pelo menos um parâmetro ou escolher Remover parâmetro

- d. Na seção Configuração de Ambiente:
- i. Para a configuração de Perfil de trabalho, escolha um perfil do IAM que tenha permissões para as APIs da AWS. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

 Note

Somente funções que tenham o relacionamento de confiança Função da tarefa do Amazon Elastic Container Service são mostradas aqui. Para obter mais informações sobre como criar um perfil do IAM para os trabalhos do seu AWS Batch, consulte [Como criar um perfil do IAM e uma política para suas tarefas](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

- ii. Para vCPUs, especifique o número de vCPUs a serem reservadas para o contêiner. Esse parâmetro é mapeado para CpuShares na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção `--cpu-shares` para [docker run](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Você deve especificar pelo menos uma vCPU.
- iii. Em Memória, insira o limite de memória que está disponível para o contêiner. Caso seu contêiner tente exceder a memória especificada, o mesmo será interrompido. Esse parâmetro é mapeado para Memory na seção [Criar um Contêiner](#) da [API Remota Docker](#) e para a opção `--memory` para [docker run](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa.

Se você usa o GuardDuty Runtime Monitoring, há uma pequena sobrecarga de memória para o agente de segurança do GuardDuty. Por isso, o limite de memória deve incluir o tamanho do agente de segurança GuardDuty. Para obter informações sobre os limites de memória de agentes de segurança do GuardDuty, consulte [Limites de CPU e memória](#) no Guia do usuário do GuardDuty. Para obter informações sobre as práticas recomendadas, consulte [Como corrigir erros de falta de memória nas minhas tarefas do Fargate após habilitar o Monitoramento de runtime](#), no Guia do desenvolvedor do Amazon ECS.

 Note

Para maximizar o uso dos recursos, priorize a memória para trabalhos de um tipo específico de instância. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).

- e. (Opcional) Em Variáveis de Ambiente, escolha Adicionar Variável de Ambiente para adicionar variáveis de ambiente como pares de nome/valor. Essas variáveis serão passadas para o contêiner.

- f. (Opcional) Em Segredos, escolha Adicionar segredo para adicionar segredos como pares de nome-valor. Esses segredos são expostos no contêiner. Para obter mais informações, consulte [LogConfiguration:secretOptions](#).
 - g. Escolha Próxima página.
13. (Opcional) Na seção Configuração do Linux:
- a. Em Usuário, insira um nome do usuário a ser usado dentro do contêiner.
 - b. Ative Ativar processo init para executar um processo init dentro do contêiner. Este processo encaminha sinais e colhe processos.
 - c. Ative Habilitar sistema de arquivos somente para leitura para remover o acesso de gravação ao volume.
 - d. (Opcional) Expanda Configuração adicional.
 - e. Em Configuração dos pontos de montagem, escolha Adicionar configuração de pontos de montagem para adicionar pontos de montagem para volumes de dados. Você deve especificar o volume de origem e o caminho do contêiner. Esses pontos de montagem são passados para o Docker daemon em uma instância de contêiner.
 - f. Para Configuração de volumes, escolha Adicionar volume para criar uma lista de volumes a serem passados para o contêiner. Insira Nome e Caminho de origem para o volume, e então escolha Adicionar volume.
 - g. Na seção Propriedades da tarefa:
 - i. Em Perfil de execução - condicional, escolha um perfil para permitir que os atendentes do Amazon ECS façam chamadas de API AWS em seu nome. Para obter mais informações sobre criar um Perfil de execução, consulte [Tutorial: criar o perfil de execução do IAM](#).
 - ii. Escolha Habilitar o comando de execução do ECS para permitir o acesso direto ao shell do contêiner do Amazon ECS e ignorar o sistema operacional do host. Você deve escolher uma Função de tarefa.
-  **Important**

O comando Executar ECS exige que o sistema de arquivos seja gravável.
- iii. Em Perfil da tarefa, escolha um perfil do Amazon ECS Identity and Access Management (IAM) que permita que o contêiner faça chamadas de API AWS em seu

nome. Para obter mais informações, consulte [Perfil do IAM de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

h. Na seção Configuração do log:

- i. (Opcional) Em Driver de log, escolha o driver de log a ser usado. Para obter mais informações sobre os drivers de log disponíveis, consulte [LogConfiguration:logDriver](#).

 Note

Por padrão, o driver de log `awslogs` é usado.

- ii. (Opcional) Em Opções, escolha Adicionar opção para adicionar uma opção. Insira um par nome-valor e então escolha Adicionar opção.
- iii. (Opcional) Em Segredos, escolha Adicionar segredo para adicionar um segredo. Em seguida, insira um par nome-valor e escolha Adicionar segredo.

 Tip

Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

14. Escolha Próxima página.

15. Para a Revisão de Definição de Tarefa, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar Definição de Trabalho.

Criar uma definição de trabalho de nó único nos recursos do Amazon EKS

Conclua as seguintes etapas para criar uma definição de trabalho de nó único no Amazon Elastic Kubernetes Service (Amazon EKS).

Para criar uma nova definição de trabalho em recursos do Amazon EKS:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação superior, escolha a Região da AWS a ser usada.
3. No painel de navegação esquerdo, escolha Definições de trabalho.
4. Escolha Criar.
5. Em Tipo de Orquestração, escolha Elastic Kubernetes Service (EKS).

6. Em Nome, insira um nome exclusivo para a sua definição de trabalho. Os nomes podem ter até 128 caracteres. Podem conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
7. (Opcional) Em Tempo Limite de Execução, insira o valor do tempo limite (em segundos). O tempo limite de execução é o período de tempo antes que um trabalho não concluído seja encerrado. Se uma tentativa exceder o tempo limite, ela será interrompida e passada para um status FAILED. Para mais informações, consulte [Tempos limite de trabalho](#). O valor mínimo é 60 segundos.
8. (Opcional) Ative a Prioridade de Agendamento. Insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade sobre valores mais baixos.
9. (Opcional) Expanda Tags e então, escolha Adicionar Tag para adicionar tags ao recurso.
10. Escolha Próxima página.
11. Na seção Propriedades pod EKS:
 - a. Em Nome da Conta de Serviço, insira uma conta que forneça uma identidade para processos executados em pod.
 - b. Ative a rede Host para usar o modelo de rede Kubernetes pod e abrir uma porta de recepção para conexões de entrada. Desative essa configuração somente para comunicações de saída.
 - c. Em Política DNS, escolha uma das opções a seguir:
 - Sem Valor (Nulo): o pod ignora as configurações DNS do ambiente Kubernetes.
 - Padrão: o pod herda a configuração de resolução de nomes do nó onde for executado.

 Note

Se uma política DNS não for especificada, o Padrão não será a política DNS padrão. O ClusterFirst será usado no lugar.

- ClusterFirst: qualquer consulta ao DNS que não corresponda ao sufixo de domínio de cluster configurado será encaminhada para o servidor de nomes upstream herdado do nó.
 - ClusterFirstWithHostNet: utilize caso a Rede Host esteja ativada.
- d. (Opcional) Para Volumes, selecione Adicionar volume e, em seguida:
 - i. Adicione um Nome para seu volume.

- ii. (Opcional) Adicione o Caminho do host para o diretório no host.
 - iii. (Opcional) Adicione um Meio e um Limite de tamanho para configurar um [Kubernetes emptyDir](#).
 - iv. (Opcional) Forneça um Nome secreto para o pod e se o segredo é Opcional.
 - v. (Opcional) Defina um Nome de declaração para anexar uma [Declaração de volume persistente](#) do Kubernetes ao pod, e se ela é Somente leitura.
- e. (Opcional) Em Rótulos de Pod, escolha Adicionar Rótulos de Pod e então, insira um par nome-valor.

 Important

O prefixo de um rótulo de pod não pode conter `kubernetes.io/`, `k8s.io/` ou `batch.amazonaws.com/`.

- f. (Opcional) Para Anotações do pod, escolha Adicionar anotações e insira um par nome-valor.

 Important

O prefixo para anotação de pod não pode conter `kubernetes.io/`, `k8s.io/` ou `batch.amazonaws.com/`.

- g. Escolha Próxima página.
- h. Na seção Configuração de Contêiner:
- i. Em Nome, insira um nome exclusivo para o contêiner. O nome deve começar com uma letra ou número e pode conter até 63 caracteres. Pode conter letras minúsculas, maiúsculas, números e hífen (-).
 - ii. Para Imagem, escolha a imagem do Docker a ser usada em seu trabalho. Por padrão, imagens no registro Docker Hub estarão disponíveis. Você também pode especificar outros repositórios com `repository-url/image:tag`. O nome pode ter até 255 caracteres. Pode conter letras maiúsculas, minúsculas, números, hífens (-), sublinhados (_), dois pontos (:), pontos (.), barras (/) e jogos da velha (#). Esse parâmetro é mapeado para Image na seção [Criar um Contêiner](#) da [API remota Docker](#) e do parâmetro IMAGE de [docker run](#)

 Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo, imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` ou `registry/repository[@digest]` completa (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - As imagens em repositórios do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` completa (por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Imagens em repositórios oficiais em Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
 - Imagens em outros repositórios Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
 - Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).
- iii. (Opcional) Em Política de Extração de Imagem, escolha quando as imagens serão puxadas.
 - iv. (Opcional) Em Comando, insira um comando JSON para passar para o contêiner.
 - v. (Opcional) Em Argumentos, insira argumentos a serem transmitidos ao contêiner. Caso um argumento não seja fornecido, o comando de imagem de contêiner será usado.
- i. (Opcional) Você pode adicionar parâmetros a definição do trabalho como mapeamentos de nome-valor para substituir padrões da definição da tarefa. Para adicionar um parâmetro:
 - Em Parâmetros, insira um par nome-valor e então, escolha Adicionar Parâmetro.

 Important

Se escolher Adicionar Parâmetro, deverá configurar pelo menos um parâmetro, ou escolher Remover Parâmetro

j. Na seção Configuração de Ambiente:

- i. Para vCPUs, especifique o número de vCPUs a serem reservadas para o contêiner. Esse parâmetro é mapeado para CpuShares na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção `--cpu-shares` para [docker run](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Você deve especificar pelo menos uma vCPU.
- ii. Em Memória, insira o limite de memória disponível para o contêiner. Caso seu contêiner tente exceder a memória especificada, o mesmo será interrompido. Esse parâmetro é mapeado para Memory na seção [Criar um Contêiner](#) da [API Remota Docker](#) e para a opção `--memory` para [docker run](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa.

 Note

Para maximizar o uso dos recursos, priorize a memória para trabalhos de um tipo específico de instância. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).

- k. (Opcional) Em Variáveis de Ambiente, escolha Adicionar Variável de Ambiente para adicionar variáveis de ambiente como pares de nome/valor. Essas variáveis serão passadas para o contêiner.
- l. (Opcional) Em Montagem de Volume:
 - i. Escolha Adicionar Montagem de Volume.
 - ii. Insira um Nome e, em seguida, um Caminho de Montagem no contêiner onde o volume estiver montado. Insira um SubPath para especificar um subcaminho dentro do volume referenciado em vez de sua raiz.
 - iii. Escolha Somente Leitura para remover permissões de registro no volume.
 - iv. Escolha Adicionar Montagem de Volume.
- m. (Opcional) Em Executar como Usuário, insira uma ID de usuário para executar o processo do contêiner.

 Note

A ID do usuário deve existir na imagem para que o contêiner seja executado.

- n. (Opcional) Em Executar como Grupo, insira uma ID de grupo para executar o tempo de processamento runtime do contêiner.

 Note

A ID do grupo deve existir na imagem para que o contêiner seja executado.

- o. (Opcional) Para conceder ao contêiner de trabalho permissões elevadas na instância do host (semelhante ao usuário root), arraste o slider Privilegiado para a direita. Esse parâmetro é mapeado para Privileged na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção `--privileged` para [docker run](#).
- p. (Opcional) Ative o Sistema de Arquivamento Raiz Somente para Leitura para remover o acesso de registro ao sistema de arquivamento raiz.
- q. (Opcional) Ative Executar como Não raiz para executar contêineres em pod como usuário não raiz.

 Note

Caso a Execução como Não Raiz esteja ligada, o kubelet valida a imagem em runtime para verificar se a mesma não será executada como UID 0.

- r. Escolha Próxima página.
12. Para a Revisão de Definição de Tarefa, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar Definição de Trabalho.

Criar uma definição de trabalho de nó único com vários contêineres nos recursos do Amazon EC2

Conclua as seguintes etapas para criar uma definição de trabalho com um único nó com vários contêineres nos recursos do Amazon Elastic Compute Cloud (Amazon EC2).

Para criar uma nova definição de trabalho em recursos do Amazon EC2:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, escolha a Região da AWS a ser usada.
3. No painel de navegação esquerdo, escolha Definições de trabalho.
4. Escolha Criar.
5. Para o Tipo de orquestração, escolha Amazon Elastic Compute Cloud (Amazon EC2).
6. Em Estrutura de definição do trabalho, desative o processamento Usar estrutura legado containerProperties.
7. Em Configuração da plataforma EC2, desative o processamento de Ativar paralelo de vários nós.
8. Escolha Próximo.
9. Na seção Configuração geral, insira o seguinte:
 - a. Em Nome, insira um nome exclusivo para a sua definição de trabalho. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
 - b. Em Tempo limite de execução - opcional, insira o valor do tempo limite (em segundos). O tempo limite de execução é o período de tempo antes que um trabalho não concluído seja encerrado. Se uma tentativa exceder o tempo limite, ela será interrompida e passada para um status FAILED. Para mais informações, consulte [Tempos limite de trabalho](#). O valor mínimo é 60 segundos.
 - c. Ative Prioridade de agendamento - opcional. Insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade.
 - d. Expanda Tags - opcional e escolha Adicionar Tag para adicionar tags ao recurso. Insira uma chave e um valor opcional e então escolha Adicionar Tag.
 - e. Ative Propagar tags para propagar tags do trabalho e da definição de trabalho para a tarefa do Amazon ECS.
10. Na seção Estratégia de repetição - opcional, insira o seguinte:
 - a. Para Tentativas de trabalho, insira o número de vezes que o AWS Batch tenta mover o trabalho para o status de RUNNABLE. Insira um número inteiro entre 1 e 10.
 - b. Em Repetir as condições da estratégia, escolha Adicionar avaliação na saída. Insira pelo menos um valor de parâmetro e escolha uma Ação. Para cada conjunto de condições, Ação deve ser definida como Tentar Novamente ou Sair. Essas ações significam o seguinte:

- Tentar novamente: o AWS Batch tenta novamente até que o número de tentativas de trabalho que você especificou seja atingido.
- Sair: o AWS Batch para de tentar novamente o trabalho.

 Important

Se você escolher Adicionar avaliação na saída, deverá configurar pelo menos um parâmetro e ou escolher uma Ação ou escolher Remover avaliação na saída.

11. Na seção Propriedades da tarefa, insira o seguinte:

- a. Em Perfil de execução - condicional, escolha uma função para permitir que agentes do Amazon ECS façam chamadas de API AWS em seu nome. Para obter mais informações sobre criar um Perfil de execução, consulte [Tutorial: criar o perfil de execução do IAM](#).
- b. Escolha Habilitar o comando de execução do ECS para permitir o acesso direto ao shell do contêiner do Amazon ECS e ignorar o sistema operacional do host. Você deve escolher uma Função de tarefa.

 Important

O comando Executar ECS exige que o sistema de arquivos seja gravável.

- c. Em Perfil da tarefa, escolha um perfil do Amazon ECS Identity and Access Management (IAM) que permita que o contêiner faça chamadas de API AWS em seu nome. Para obter mais informações, consulte [Perfil do IAM de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- d. Para o Modo IPC, escolha host, task ou none. Se o host for especificado, todos os contêineres dentro das tarefas que especificaram o modo IPC de host na mesma instância de contêiner compartilham os mesmos recursos IPC com a instância do Amazon EC2 do host. Se uma tarefa for especificada, todos os contêineres dentro da tarefa especificada compartilharão os mesmos recursos IPC. Se nada 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.

- e. Para o modo PID, escolha `host` ou `task`. Por exemplo, o monitoramento de arquivos associados pode precisar que `pidMode` acesse informações sobre outros contêineres em execução na mesma tarefa. Se `host` for especificado, todos os contêineres das tarefas que especificaram o modo PID `host` na mesma instância de contêiner compartilharão namespace de processo com a instância do Amazon EC2 do `host`. 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 cada contêiner.
12. Na seção **Recurso consumível**, insira o seguinte:
- a. Insira um Nome exclusivo e o Valor solicitado.
 - b. Você pode adicionar mais recursos consumíveis escolhendo **Adicionar recurso consumível**.
13. Na seção **Armazenamento**, faça o seguinte:
- a. Insira um Nome e Caminho de origem para o volume e, então, escolha **Adicionar volume**. Você também pode optar por ativar o **Habilitar EFS**.
 - b. Você pode adicionar mais Volumes escolhendo **Adicionar volume**.
14. Em **Parâmetros**, escolha **Adicionar parâmetros** para adicionar espaços reservados para substituição de parâmetros como pares de **Chave** e **Valor** opcionais.
15. Escolha **Próxima página**.
16. Na seção **Configuração de Contêiner**:
- a. Em **Name (Nome)**, insira um nome para o contêiner.
 - b. Em **Contêiner essencial**, ative se o contêiner for essencial.
 - c. Para **Imagem**, escolha a imagem do Docker a ser usada em seu trabalho. Por padrão, as imagens no registro do Docker Hub estão disponíveis. Você também pode especificar outros repositórios com `repository-url/image:tag`. Os nomes podem ter até 225 caracteres. Pode conter letras maiúsculas e minúsculas, números, hifens (-), sublinhados (_), dois pontos (:), barras (/) e sinais de número (#). Esse parâmetro é mapeado para **Image** na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro **IMAGE** de [docker run](#).

 Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo,

imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` ou `registry/repository[@digest]` completa (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - As imagens em repositórios do Amazon ECR usam as convenções de nomenclatura `registry/repository[:tag]` completa (por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`).
 - Imagens em repositórios oficiais em Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
 - Imagens em outros repositórios Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
 - Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).
- d. Em Requisitos de recursos, configure cada um dos seguintes:
- i. Em vCPUs, escolha o número de CPUs para o contêiner.
 - ii. Em Memória, escolha a quantidade de memória para o contêiner.
 - iii. Em GPU - opcional, escolha o número de GPUs para o contêiner.
- e. Para Usuário, insira o nome do usuário a ser usado dentro do contêiner.
- f. Ative Habilitar sistema de arquivos somente para leitura para remover o acesso de gravação ao volume.
- g. Ative Privilegiado para oferecer ao contêiner de trabalho permissões elevadas na instância do host, semelhante ao usuário-raiz.
- h. Em Comando, insira os comandos no campo como o equivalente da matriz de string JSON.

Esse parâmetro é mapeado para `Cmd` na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro `COMMAND` de [docker run](#). Para obter mais informações sobre o parâmetro Docker CMD, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Você pode usar os valores padrão de substituição de parâmetros e espaços reservados no seu comando. Para obter mais informações, consulte [Parameters](#).

- i. Em Credenciais do repositório - opcional, insira o ARN do segredo que contém suas credenciais.
- j. Em Variáveis de ambiente - opcional, escolha Adicionar variáveis de ambiente para adicionar variáveis de ambiente a serem passadas para o contêiner.
- k. Na seção Parâmetros do Linux - opcional:
 - i. Ative Ativar processo init para executar um processo init dentro do contêiner.
 - ii. Em Tamanho da memória compartilhada, insira o tamanho (em MiB) do volume `/dev/shm`
 - iii. Em Tamanho máximo de troca, insira a quantidade total de memória de troca (em MiB) que o contêiner pode usar.
 - iv. Em Swappiness, insira um valor entre 0 e 100 para indicar o comportamento de troca de memória do contêiner. Se você não especificar um valor e a troca estiver ativada, o valor assumirá 60 como padrão.
 - v. Em Dispositivos, escolha Adicionar dispositivo para adicionar um dispositivo:
 - A. Em Container path (Caminho do contêiner), especifique o caminho na instância de contêiner para expor o dispositivo mapeado para a instância de host. Se isso for deixado em branco, o caminho de host será usado no contêiner.
 - B. Em Host path, especifique o caminho de um dispositivo na instância de host.
 - C. Em Permissões, selecione uma ou mais permissões para serem aplicadas ao dispositivo. As permissões disponíveis são LER, GRAVAR e MKNOD.
 - vi. Em Tmpfs, escolha Adicionar tmpfs para adicionar uma montagem `tmpfs`.

l.

 Note

O registro em logs do Firelens deve ser feito em um contêiner dedicado. Para configurar o registro em log do Firelens:

- Em cada contêiner, exceto seu contêiner dedicado do Firelens, defina o Driver de registro em log como `awsfirelens`

- Em seu contêiner Firelens, defina a Configuração Firelens - opcional e a Configuração de registro em log - opcional para o destino de registro em log.

Na seção Configuração Direlens - opcional:

 Important

O AWS Batch impõe o modo de rede host em trabalhos Amazon ECS não MNP e não Fargate. [O usuário-raiz é necessário](#) para o Amazon ECS Firelens. Ao executar tarefas que usam o modo de rede host, a Amazon ECS aconselha a não executar contêineres usando o usuário raiz (UID 0) para [melhor segurança](#). Portanto, todos os trabalhos não MNP e não Fargate ECS com log Firelens não atenderão às melhores práticas de segurança.

- Para Tipo, escolha fluentd ou fluentbit.
 - Em Opções, insira o par nome/valor da opção. Você pode adicionar mais Opções usando Opção adicionada.
- m. Na seção Configuração de registro em log - opcional:
- Em Driver de log, escolha o driver de log a ser usado. Para obter mais informações sobre os drivers de log disponíveis, consulte [LogConfiguration:logDriver](#).

 Note

Por padrão, o driver de log awslogs é usado.

- Em Opções, escolha Adicionar opção para adicionar uma opção. Insira um par nome-valor e então escolha Adicionar opção.
- Em Segredos, escolha Adicionar segredo. Insira um par nome-valor e escolha Adicionar segredo para adicionar um segredo.

 Tip

Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

- n. Em Pontos de montagem - opcional, escolha Adicionar pontos de montagem para adicionar pontos de montagem para volumes de dados. Você deve especificar o volume de origem e o caminho do contêiner.
- o. Em Segredos - opcional, escolha Adicionar segredo para adicionar um segredo. Em seguida, insira um par nome-valor e escolha Adicionar segredo.

 Tip

Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

- p. Em Ulimits - opcional, escolha Adicionar ulimit para adicionar um valor de `ulimits` ao contêiner. Insira os valores de Nome, Limite flexível, e Limite rígido e, em seguida, escolha Adicionar ulimit.
 - q. Em Dependências - opcional, escolha Adicionar dependências de contêiner. Escolha o nome do contêiner e seu estado para determinar quando o contêiner é iniciado.
17. Se você tiver apenas um contêiner configurado, deverá escolher Adicionar contêiner e concluir a configuração do novo contêiner. Caso contrário, escolha Avançar para revisar.

Criar uma definição de trabalho paralelo de vários nós

Antes de poder executar trabalhos no AWS Batch, você deve criar uma definição de tarefa. Esse processo varia ligeiramente entre trabalhos em paralelo de nó único e de vários nós. Este tópico aborda especificamente como criar uma definição de trabalho para um trabalho paralelo de vários nós do AWS Batch (também conhecido como programação de grupos). Para obter mais informações, consulte [Trabalhos em paralelo de vários nós](#).

 Note

O AWS Fargate não oferece suporte a trabalhos paralelos de vários nós.

Conteúdo

- [Tutorial: criar uma definição de trabalho em paralelo de vários nós nos recursos do Amazon EC2](#)

Tutorial: criar uma definição de trabalho em paralelo de vários nós nos recursos do Amazon EC2

Para criar uma definição de trabalho paralelo de vários nós nos recursos do Amazon Elastic Compute Cloud (Amazon EC2).

Note

Para criar uma definição de trabalho de nó único, consulte [Criar uma definição de trabalho de nó único nos recursos do Amazon EC2](#).

Para criar uma definição de trabalho em paralelo de vários nós nos recursos do Amazon EC2:

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Definições de trabalho.
4. Escolha Criar.
5. Para o Tipo de orquestração, escolha Amazon Elastic Compute Cloud (Amazon EC2).
6. Em Habilitar paralelo de vários nós, ative o paralelo de vários nós.
7. Em Nome, insira um nome exclusivo para a sua definição de trabalho. O nome pode ter até 128 caracteres e pode conter letras minúsculas e maiúsculas, números, hifens (-) e sublinhados (_).
8. (Opcional) Em Execution timeout, especifique o número máximo de segundos permitidos para as execuções das tentativas de trabalho. Se uma tentativa exceder o tempo limite, ela será interrompida e passada para um status FAILED. Para obter mais informações, consulte [Tempos limite de trabalho](#).
9. (Opcional) Ative a Prioridade de Agendamento. Insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade sobre valores mais baixos.
10. (Opcional) Para Tentativas de Trabalho, insira o número máximo de vezes que AWS Batch tenta mover o trabalho para um status RUNNABLE. Insira um número inteiro entre 1 e 10.
11. (Opcional) Em Repetir condições da estratégia, escolha Adicionar avaliação na saída. Insira pelo menos um valor de parâmetro e escolha uma Ação. Para cada conjunto de condições, Ação deve ser definida como Tentar Novamente ou Sair. Essas ações significam o seguinte:
 - Tentar novamente: o AWS Batch tenta novamente até que o número de tentativas de trabalho que você especificou seja atingido.

- Sair: o AWS Batch para de tentar novamente o trabalho.

 Important

Se você escolher Adicionar avaliação na saída, deverá configurar pelo menos um parâmetro e ou escolher uma Ação ou escolher Remover avaliação na saída.

12. (Opcional) Expanda Tags e então, escolha Adicionar Tag para adicionar tags ao recurso. Insira uma chave e um valor opcional e, em seguida, selecione Add tag. Você também pode ativar a opção Propagate tags para propagá-las do trabalho e da definição de trabalho para a tarefa do Amazon ECS.
13. Escolha Próxima página.
14. Em Number of nodes, insira o número total de nós a serem usados em sua tarefa.
15. Em Main node, insira o índice do nó a ser usado para o nó principal. O índice do nó principal padrão é 0.
16. Em Instance Type, selecione um tipo de instância.

 Note

O tipo de instância que você escolher se aplica a todos os nós.

17. Em Parâmetros, escolha Adicionar parâmetros para adicionar espaços reservados para substituição de parâmetros como pares de Chave e Valor opcionais.
18. Na seção Node ranges:
 - a. Selecione Add node range. Isso cria uma seção Node range.
 - b. Em Target nodes, especifique o intervalo para o seu grupo de nós usando a anotação *range_start:range_end*.

Você pode criar até cinco intervalos de nós para os nós especificados para seu trabalho. Os intervalos de nós usam o valor do índice para um nó e o índice do nó começa em 0. Certifique-se de que o valor do índice final do intervalo do seu grupo final de nós seja um a menos do que o número de nós que você especificou. Por exemplo, suponha que você especificou 10 nós e deseja usar um único grupo de nós. Então, seu intervalo final é 9.

- c. Para Imagem, escolha a imagem do Docker a ser usada em seu trabalho. Por padrão, as imagens no registro do Docker Hub estão disponíveis. Você também pode especificar

outros repositórios com `repository-url/image:tag`. O nome pode ter até 225 caracteres. Pode conter letras maiúsculas e minúsculas, números, hifens (-), sublinhados (_), dois pontos (:), barras (/) e sinais de número (#). Esse parâmetro é mapeado para Image na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro IMAGE de [docker run](#).

 Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo, imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura completa `registry/repository[:tag]` ou `registry/repository[@digest]` (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).
 - As imagens nos repositórios do Amazon ECR usam a convenção de `registry/repository[:tag]` nomenclatura completa. Por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`
 - As imagens em repositórios oficiais no Docker Hub usam um único nome (por exemplo, `ubuntu` ou `mongo`).
 - Imagens em outros repositórios Docker Hub são qualificadas com um nome de organização (por exemplo, `amazon/amazon-ecs-agent`).
 - Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).
- d. Em Comando, insira os comandos no campo como o equivalente da matriz de string JSON.

Esse parâmetro é mapeado para `Cmd` na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro `COMMAND` de [docker run](#). Para obter mais informações sobre o parâmetro Docker `CMD`, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

Você pode usar os valores padrão de substituição de parâmetros e espaços reservados no seu comando. Para obter mais informações, consulte [Parameters](#).

- e. Para vCPUs, especifique o número de vCPUs a serem reservadas para o contêiner. Esse parâmetro é mapeado para CpuShares na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção `--cpu-shares` para [docker run](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Você deve especificar pelo menos uma vCPU.
- f. Para Memória, especifique o limite rígido (em MiB) de memória a ser apresentado ao contêiner do trabalho. Caso seu contêiner tente exceder a memória especificada, o mesmo será interrompido. Esse parâmetro é mapeado para Memory na seção [Criar um Contêiner](#) da [API Remota Docker](#) e para a opção `--memory` para [docker run](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa.

 Note

Para maximizar a utilização de recursos, você pode fornecer aos seus trabalhos o máximo de memória possível para um determinado tipo de instância. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).

- g. (Opcional) Em Number of GPUs, especifique o número de GPUs que a tarefa vai usar. A tarefa será executada em um contêiner com o número especificado de GPUs fixadas a esse contêiner.
- h. (Opcional) Para Job role, é possível especificar uma função do perfil do IAM que fornece o contêiner no trabalho com permissões para usar as APIs da AWS. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para obter mais informações, consulte [IAM Roles for Tasks](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

 Note

Para trabalhos que são executados nos recursos do Fargate, é necessário um perfil de trabalho.

 Note

Somente funções que tenham o relacionamento de confiança Função da tarefa do Amazon Elastic Container Service são mostradas aqui. Para obter mais informações sobre como criar um perfil do IAM para os trabalhos do seu AWS Batch, consulte [Como criar um perfil do IAM e uma política para suas tarefas](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

- i. (Opcional) Para Função de execução, especifique um perfil do IAM que conceda aos atendentes de contêiner do Amazon ECS permissão para fazer chamadas da API da AWS em seu nome. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para obter mais informações, consulte [Perfis do IAM para execução de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

19. (Opcional) Expanda Configuração adicional:

- a. Em Environment variables, escolha Adicionar Variável de Ambiente para adicionar variáveis de ambiente como pares de nome e valor. Essas variáveis serão passadas para o contêiner.
- b. Para Job role, é possível especificar uma função do perfil do IAM que fornece o contêiner no seu trabalho com permissões para usar as APIs da AWS. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para obter mais informações, consulte [IAM Roles for Tasks](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

 Note

Para trabalhos que são executados nos recursos do Fargate, é necessário um perfil de trabalho.

 Note

Somente funções que tenham o relacionamento de confiança Função da tarefa do Amazon Elastic Container Service são mostradas aqui. Para obter mais informações sobre como criar um perfil do IAM para os trabalhos do seu AWS

Batch, consulte [Como criar um perfil do IAM e uma política para suas tarefas](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

- c. Para Execution role, especifique um perfil do IAM que conceda aos agentes de contêiner do Amazon ECS permissão para fazer chamadas de API AWS em seu nome. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para obter mais informações, consulte [Perfis do IAM para execução de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

20. Na seção Configuração de segurança:

- a. (Opcional) Para oferecer ao contêiner da tarefa privilégios elevados na instância do host (semelhante ao usuário root), selecione Privileged. Esse parâmetro é mapeado para Privileged na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção --privileged para [docker run](#).
- b. (Opcional) Para Usuário, insira o nome do usuário a ser usado dentro do contêiner. Esse parâmetro é mapeado para User na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção --user para [docker run](#).
- c. (Opcional) Em Segredos, escolha Adicionar segredo para adicionar segredos como pares de nome-valor. Esses segredos são expostos no contêiner. Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

21. Na seção Configuração do Linux:

- a. Ative Habilitar sistema de arquivos somente para leitura para remover o acesso de gravação ao volume.
- b. (Opcional) Ative o processo init para executar um processo `init` dentro do contêiner. Este processo encaminha sinais e colhe processos.
- c. Para Tamanho da memória compartilhada, insira o tamanho (em MiB) do volume `/dev/shm`.
- d. Em Tamanho máximo de troca, insira a quantidade total de memória de troca (em MiB) que o contêiner pode usar.
- e. Em Swappiness, insira um valor entre 0 e 100 para indicar o comportamento de troca de memória do contêiner. Se você não especificar um valor e a troca estiver ativada, o valor padrão será 60. Para obter mais informações, consulte [LinuxParameters:swappiness](#).
- f. (Opcional) Em Dispositivos, escolha Adicionar dispositivo para adicionar um dispositivo:

- i. Em Container path (Caminho do contêiner), especifique o caminho na instância de contêiner para expor o dispositivo mapeado para a instância de host. Se isso for deixado em branco, o caminho de host será usado no contêiner.
 - ii. Em Host path, especifique o caminho de um dispositivo na instância de host.
 - iii. Em Permissões, selecione uma ou mais permissões para serem aplicadas ao dispositivo. As permissões disponíveis são LER, GRAVAR e MKNOD.
22. (Opcional) Em Pontos de montagem, escolha Adicionar configuração de pontos de montagem para adicionar pontos de montagem para volumes de dados. Você deve especificar o volume de origem e o caminho do contêiner. Esses pontos de montagem são passados para o Docker daemon em uma instância de contêiner. Você também pode optar por tornar o volume somente para leitura.
23. (Opcional) Para Configuração de ulimits, escolha Adicionar ulimit para adicionar um valor de `ulimits` ao contêiner. Insira os valores de Nome, Limite flexível, e Limite rígido e, em seguida, escolha Adicionar ulimit.
24. (Opcional) Para Configuração de volumes, escolha Adicionar volume para criar uma lista de volumes a serem passados para o contêiner. Insira Nome e Caminho de origem para o volume e escolha Adicionar volume. Você também pode optar por ativar o Habilitar EFS.
25. (Opcional) Para Tmpfs, escolha Adicionar tmpfs para adicionar uma montagem `tmpfs`.
26. Na seção Propriedades da tarefa:
- a. Em Perfi de execução - condicional, escolha um perfil para permitir que os atendentes do Amazon ECS façam chamadas de API AWS em seu nome. Para obter mais informações sobre criar um Perfil de execução, consulte [Tutorial: criar o perfil de execução do IAM](#).

b.

 Important

Para usar o Comando de execução ECS, seu ambiente computacional deve atender às [considerações do ambiente computacional para trabalhos paralelos de vários nós](#).

Escolha Habilitar o comando de execução do ECS para permitir o acesso direto ao shell do contêiner do Amazon ECS e ignorar o sistema operacional do host. Você deve escolher uma Função de tarefa.

 Important

O comando Executar ECS exige que o sistema de arquivos seja gravável.

- c. Em Perfil da tarefa, escolha um perfil do Amazon ECS Identity and Access Management (IAM) que permita que o contêiner faça chamadas de API AWS em seu nome. Para obter mais informações, consulte [Perfil do IAM de tarefa do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

27. (Opcional) Na seção Configuração de registro:

- a. Em Driver de log, escolha o driver de log a ser usado. Para obter mais informações sobre os drivers de log disponíveis, consulte [LogConfiguration:logDriver](#).

 Note

Por padrão, o driver de log `awslogs` é usado.

- b. Em Opções, escolha Adicionar opção para adicionar uma opção. Insira um par nome-valor e então escolha Adicionar opção.
- c. Em Segredos, escolha Adicionar segredo. Insira um par nome-valor e escolha Adicionar segredo para adicionar um segredo.

 Tip

Para obter mais informações, consulte [LogConfiguration:secretOptions](#).

28. Escolha Próxima página.

29. Para a Revisão de Definição de Tarefa, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar Definição de Trabalho.

Modelo de definição de trabalho que usa ContainerProperties

Veja um modelo de definição de trabalho vazio que inclui um só contêiner. É possível usá-lo para criar sua definição de tarefa, que pode ser salva em um arquivo e usada com a AWS CLI `--cli-input-json` opção. Para obter mais informações sobre esses parâmetros, consulte [JobDefinition](#).

Note

Você pode gerar um modelo de definição de trabalho de contêiner único com o seguinte comando AWS CLI:

```
$ aws batch register-job-definition --generate-cli-skeleton
```

```
{
  "jobDefinitionName": "",
  "type": "container",
  "parameters": {
    "KeyName": ""
  },
  "schedulingPriority": 0,
  "containerProperties": {
    "image": "",
    "vcpus": 0,
    "memory": 0,
    "command": [
      ""
    ],
    "jobRoleArn": "",
    "executionRoleArn": "",
    "volumes": [
      {
        "host": {
          "sourcePath": ""
        },
        "name": "",
        "efsVolumeConfiguration": {
          "fileSystemId": "",
          "rootDirectory": "",
          "transitEncryption": "ENABLED",
          "transitEncryptionPort": 0,
          "authorizationConfig": {
            "accessPointId": "",
            "iam": "DISABLED"
          }
        }
      }
    ]
  },
}
```

```
"environment": [
  {
    "name": "",
    "value": ""
  }
],
"mountPoints": [
  {
    "containerPath": "",
    "readOnly": true,
    "sourceVolume": ""
  }
],
"readonlyRootFilesystem": true,
"privileged": true,
"ulimits": [
  {
    "hardLimit": 0,
    "name": "",
    "softLimit": 0
  }
],
"user": "",
"instanceType": "",
"resourceRequirements": [
  {
    "value": "",
    "type": "MEMORY"
  }
],
"linuxParameters": {
  "devices": [
    {
      "hostPath": "",
      "containerPath": "",
      "permissions": [
        "WRITE"
      ]
    }
  ],
  "initProcessEnabled": true,
  "sharedMemorySize": 0,
  "tmpfs": [
    {
```

```

        "containerPath": "",
        "size": 0,
        "mountOptions": [
            ""
        ]
    },
    ],
    "maxSwap": 0,
    "swappiness": 0
},
"logConfiguration": {
    "logDriver": "syslog",
    "options": {
        "KeyName": ""
    },
    "secretOptions": [
        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"secrets": [
    {
        "name": "",
        "valueFrom": ""
    }
],
"networkConfiguration": {
    "assignPublicIp": "DISABLED"
},
"fargatePlatformConfiguration": {
    "platformVersion": ""
}
},
"nodeProperties": {
    "numNodes": 0,
    "mainNode": 0,
    "nodeRangeProperties": [
        {
            "targetNodes": "",
            "container": {
                "image": "",
                "vcpus": 0,

```

```

    "memory": 0,
    "command": [
        ""
    ],
    "jobRoleArn": "",
    "executionRoleArn": "",
    "volumes": [
        {
            "host": {
                "sourcePath": ""
            },
            "name": "",
            "efsVolumeConfiguration": {
                "fileSystemId": "",
                "rootDirectory": "",
                "transitEncryption": "DISABLED",
                "transitEncryptionPort": 0,
                "authorizationConfig": {
                    "accessPointId": "",
                    "iam": "ENABLED"
                }
            }
        }
    ],
    "environment": [
        {
            "name": "",
            "value": ""
        }
    ],
    "mountPoints": [
        {
            "containerPath": "",
            "readOnly": true,
            "sourceVolume": ""
        }
    ],
    "readonlyRootFilesystem": true,
    "privileged": true,
    "ulimits": [
        {
            "hardLimit": 0,
            "name": "",
            "softLimit": 0
        }
    ]
}

```

```

    }
  ],
  "user": "",
  "instanceType": "",
  "resourceRequirements": [
    {
      "value": "",
      "type": "MEMORY"
    }
  ],
  "linuxParameters": {
    "devices": [
      {
        "hostPath": "",
        "containerPath": "",
        "permissions": [
          "WRITE"
        ]
      }
    ],
    "initProcessEnabled": true,
    "sharedMemorySize": 0,
    "tmpfs": [
      {
        "containerPath": "",
        "size": 0,
        "mountOptions": [
          ""
        ]
      }
    ],
    "maxSwap": 0,
    "swappiness": 0
  },
  "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
      "KeyName": ""
    },
    "secretOptions": [
      {
        "name": "",
        "valueFrom": ""
      }
    ]
  }
}

```

```

        ]
      },
      "secrets": [
        {
          "name": "",
          "valueFrom": ""
        }
      ],
      "networkConfiguration": {
        "assignPublicIp": "DISABLED"
      },
      "fargatePlatformConfiguration": {
        "platformVersion": ""
      }
    }
  ]
},
"retryStrategy": {
  "attempts": 0,
  "evaluateOnExit": [
    {
      "onStatusReason": "",
      "onReason": "",
      "onExitCode": "",
      "action": "RETRY"
    }
  ]
},
"propagateTags": true,
"timeout": {
  "attemptDurationSeconds": 0
},
"tags": {
  "KeyName": ""
},
"platformCapabilities": [
  "EC2"
],
"eksProperties": {
  "podProperties": {
    "serviceAccountName": "",
    "hostNetwork": true,
    "dnsPolicy": "",

```

```

    "containers": [
      {
        "name": "",
        "image": "",
        "imagePullPolicy": "",
        "command": [
          ""
        ],
        "args": [
          ""
        ],
        "env": [
          {
            "name": "",
            "value": ""
          }
        ],
        "resources": {
          "limits": {
            "KeyName": ""
          },
          "requests": {
            "KeyName": ""
          }
        },
        "volumeMounts": [
          {
            "name": "",
            "mountPath": "",
            "readOnly": true
          }
        ],
        "securityContext": {
          "runAsUser": 0,
          "runAsGroup": 0,
          "privileged": true,
          "readOnlyRootFilesystem": true,
          "runAsNonRoot": true
        }
      }
    ],
    "volumes": [
      {
        "name": "",

```

```
        "hostPath": {
            "path": ""
        },
        "emptyDir": {
            "medium": "",
            "sizeLimit": ""
        },
        "secret": {
            "secretName": "",
            "optional": true
        }
    ]
}
}
```

Parâmetros de definição de trabalho para ContainerProperties

Definições de trabalho que usam [ContainerProperties](#) são divididas em várias partes:

- O nome da definição do trabalho
- O tipo de definição do trabalho
- Os padrões do espaço reservado para substituição de parâmetros
- As propriedades de contêiner para o trabalho
- As propriedades do Amazon EKS para a definição do trabalho que são necessárias para trabalhos executados em recursos do Amazon EKS
- As propriedades do nó que são necessárias para um trabalho paralelo de vários nós
- As funcionalidades da plataforma necessárias para trabalhos executados em recursos do Fargate
- Os detalhes padrão de propagação de tag da definição do trabalho
- A estratégia de repetição padrão para a definição do trabalho
- A prioridade de agendamento padrão para a definição do trabalho
- As tags padrão para a definição do trabalho
- O tempo limite padrão para a definição do trabalho

Sumário

- [Nome da definição do trabalho](#)
- [Tipo](#)
- [Parameters](#)
- [Propriedades do contêiner](#)
- [Amazon EKS properties](#)
- [Funcionalidades da plataforma](#)
- [Propagar tags](#)
- [Propriedades de nó](#)
- [Estratégia de repetição](#)
- [Prioridade de agendamento](#)
- [Tags](#)
- [Tempo limite](#)

Nome da definição do trabalho

`jobDefinitionName`

Ao registrar uma definição de trabalho, você especifica um nome. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_). A primeira definição de trabalho que está registrada com esse nome recebe uma revisão de 1. Todas as definições de trabalho subsequentes registrados com esse nome recebem um número de revisão incremental.

Tipo: Sequência

Obrigatório: Sim

Tipo

`type`

Ao registrar uma definição de trabalho, você especifica o tipo de trabalho. Se o trabalho for executado em recursos do Fargate, então `multinode` não será compatível. Para obter mais informações sobre trabalho em paralelo com vários nós, consulte [Criar uma definição de trabalho paralelo de vários nós](#).

Tipo: Sequência

Valores válidos: `container` | `multinode`

Obrigatório: Sim

Parameters

parameters

Quando você envia um trabalho, pode especificar os parâmetros que devem substituir os marcadores ou os parâmetros de definição de trabalho padrão. Os parâmetros nas solicitações de envio de trabalho têm precedência sobre os padrões em uma definição de trabalho. Isso significa que você pode usar a mesma definição de trabalho para vários outros que usam o mesmo formato. Você também pode alterar programaticamente os valores no comando no momento do envio.

Tipo: Mapa de string para string

Obrigatório: Não

Ao registrar uma definição de trabalho, você pode usar espaços reservados para substituição de parâmetros no campo `command` de propriedades de contêiner de um trabalho. A sintaxe é a seguinte.

```
"command": [  
    "ffmpeg",  
    "-i",  
    "Ref::inputfile",  
    "-c",  
    "Ref::codec",  
    "-o",  
    "Ref::outputfile"  
]
```

No exemplo acima, há os espaços reservados de substituição de parâmetro *Ref::inputfile*, *Ref::codec* e *Ref::outputfile* no comando. Você pode usar o objeto `parameters` na definição do trabalho para configurar valores padrão para esses espaços reservados. Por exemplo, para definir um padrão para o espaço reservado *Ref::codec*, você especificará o seguinte na definição do trabalho:

```
"parameters" : {"codec" : "mp4"}
```

Quando essa definição de trabalho for enviada para execução, o argumento *Ref::codec* no comando para o contêiner será substituído pelo valor padrão, mp4.

Propriedades do contêiner

Ao registrar uma definição de trabalho você deve especificar uma lista de propriedades de contêiner que são passadas para o daemon do Docker em uma instância de contêiner quando o trabalho é feito. As seguintes propriedades de contêiner são permitidas em uma definição de trabalho. Para trabalhos de nó único, essas propriedades de contêiner são configuradas no nível de definição do trabalho. Para trabalhos em paralelo de vários nós, as propriedades de contêiner são definidas no nível de [Propriedades de nó](#) para cada grupo de nós.

command

O comando que é passado para o contêiner. Esse parâmetro é mapeado para Cmd na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro COMMAND de [docker run](#). Para obter mais informações sobre o parâmetro CMD da Docker, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

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

Tipo: Matriz de strings

Obrigatório: Não

environment

As variáveis de ambiente a serem passadas para um contêiner. Esse parâmetro é mapeado para Env na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção --env para [docker run](#).

Important

Não recomendamos que você use variáveis de ambiente de texto simples para informações confidenciais, como dados de credenciais.

 Note

Variáveis de ambiente não podem começar com `AWS_BATCH`. Essa convenção de nomenclatura é reservada para variáveis definidas pelo serviço AWS Batch.

Tipo: Matriz de pares de valores-chave

Obrigatório: Não

`name`

O nome da variável de ambiente.

Tipo: Sequência

Obrigatório: Sim, quando `environment` for usado.

`value`

O valor da variável de ambiente.

Tipo: Sequência

Obrigatório: Sim, quando `environment` for usado.

```
"environment" : [  
  { "name" : "envName1", "value" : "envValue1" },  
  { "name" : "envName2", "value" : "envValue2" }  
]
```

`executionRoleArn`

Ao registrar uma definição de trabalho, você pode especificar um perfil do IAM. O perfil fornece ao agente de contêiner do Amazon ECS permissões para chamar as ações da API especificadas nas políticas colaboradoras em seu nome. Para trabalhos executados em recursos do Fargate, você deve fornecer um perfil de execução. Para obter mais informações, consulte [AWS Batch Função de execução do IAM](#).

Tipo: Sequência

Obrigatório: Não

fargatePlatformConfiguration

A configuração da plataforma para trabalhos que são executados nos recursos do Fargate. Os trabalhos que são executados nos recursos do EC2 não devem especificar esse parâmetro.

Tipo: Objeto [FargatePlatformConfiguration](#)

Obrigatório: Não

platformVersion

Use a versão da plataforma do AWS Fargate para os trabalhos ou LATEST para usar uma versão recente e aprovada da plataforma do AWS Fargate.

Tipo: Sequência

Padrão: LATEST

Obrigatório: Não

image

A imagem usada para iniciar um trabalho. 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*. São permitidos até 255 letras (caixa alta e baixa), números, hifens, sublinhados, dois pontos, ponto, barras e sinais numéricos. Esse parâmetro é mapeado para Image na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro IMAGE de [docker run](#).

Note

A arquitetura da imagem do Docker deve corresponder à arquitetura do processador dos recursos de computação nos quais eles foram programados. Por exemplo, imagens do Docker baseadas no ARM só podem ser executadas em recursos de computação baseados no ARM.

- As imagens em repositórios públicos do Amazon ECR usam as convenções de nomenclatura `completa:registry/repository[:tag]` ou `registry/repository[@digest]` (por exemplo, `public.ecr.aws/registry_alias/my-web-app:latest`).

- As imagens nos repositórios do Amazon ECR usam a convenção de `registry/repository:[tag]` nomenclatura completa. Por exemplo, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest`.
- 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`).
- Imagens em outros repositórios online também são qualificadas por um nome de domínio (por exemplo, `quay.io/assemblyline/ubuntu`).

Tipo: Sequência

Obrigatório: Sim

`instanceType`

O tipo de instância a ser usado para trabalhos paralelos de vários nós. Atualmente, todos os grupos de nós em um trabalho em paralelo de vários nós devem usar o mesmo tipo de instância. Esse parâmetro não é válido para trabalhos de contêiner de nó único ou para trabalhos que são executados em recursos Fargate.

Tipo: Sequência

Obrigatório: Não

`jobRoleArn`

Ao registrar uma definição de trabalho, você pode especificar um perfil do IAM. A função fornece ao contêiner de trabalho permissões para chamar as ações da API especificadas em suas políticas colaboradoras em seu nome. Para obter mais informações, consulte [Perfis do IAM para tarefas](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Tipo: Sequência

Obrigatório: Não

`linuxParameters`

Modificações específicas do Linux que são aplicadas ao contêiner, como detalhes para mapeamentos do dispositivo.

```
"linuxParameters": {
```

```
"devices": [  
  {  
    "hostPath": "string",  
    "containerPath": "string",  
    "permissions": [  
      "READ", "WRITE", "MKNOD"  
    ]  
  }  
],  
"initProcessEnabled": true/false,  
"sharedMemorySize": 0,  
"tmpfs": [  
  {  
    "containerPath": "string",  
    "size": integer,  
    "mountOptions": [  
      "string"  
    ]  
  }  
],  
"maxSwap": integer,  
"swappiness": integer  
}
```

Tipo: Objeto [LinuxParameters](#)

Obrigatório: Não

devices

Lista de dispositivos mapeados para o contêiner. Esse parâmetro é mapeado para Devices na seção [Criar um contêiner](#) da [Docker Remote API](#) e na opção --device de [execução do docker](#).

 Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Tipo: Matriz de objetos [Device](#)

Obrigatório: Não

hostPath

Caminho onde está o dispositivo disponível na instância de contêiner host.

Tipo: Sequência

Obrigatório: Sim

containerPath

Caminho em que o dispositivo é exposto no contêiner. Se isso não for especificado, o dispositivo será exposto ao mesmo caminho do host.

Tipo: Sequência

Obrigatório: Não

permissions

Permissões para o dispositivo no contêiner. Se não for especificado, as permissões serão definidas como READ, WRITE e MKNOD.

Tipo: Matriz de strings

Obrigatório: Não

Valores válidos: READ | WRITE | MKNOD

initProcessEnabled

Se verdadeiro, execute um processo `init` dentro do contêiner que encaminha sinais e colhe processos. Esse parâmetro mapeia para a opção `--init` para [execução do docker](#). Este parâmetro requer a versão 1.25 ou posterior da Docker Remote API ou posterior em sua instância de contêiner. Para verificar a versão da Docker Remote API na instância de contêiner, faça login na instância de contêiner e execute o seguinte comando: `sudo docker version | grep "Server API version"`

Tipo: Booleano

Obrigatório: Não

maxSwap

Especifica a quantidade total de memória de troca (em MiB) que um trabalho pode usar. Esse parâmetro será convertido na opção `--memory-swap` para [execução do docker](#) em

que o valor é a soma da memória do contêiner mais o valor de `maxSwap`. Para obter mais informações, consulte [Detalhes do `--memory-swap`](#) na documentação do Docker.

Se um valor `maxSwap` de 0 for especificado, o contêiner não usará a troca. Os valores aceitos são 0 ou qualquer número inteiro positivo. Se o parâmetro `maxSwap` for omitido, o contêiner usará a configuração de troca para a instância de contêiner na qual ele está sendo executado. Um valor `maxSwap` deve ser definido para que o parâmetro `swappiness` seja usado.

 Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Tipo: Inteiro

Obrigatório: Não

`sharedMemorySize`

O valor do tamanho (em MiB) do volume `/dev/shm`. Esse parâmetro mapeia para a opção `--shm-size` para [execução do docker](#).

 Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Tipo: Inteiro

Obrigatório: Não

`swappiness`

É possível usar isso para ajustar o comportamento de troca de memória de um contêiner. Um valor `swappiness` de 0 fará com que a troca não ocorra, a menos que seja absolutamente necessário. Um valor `swappiness` de 100 fará com que as páginas sejam trocadas de forma agressiva. Os valores aceitos são números inteiros entre 0 e 100. Se o parâmetro `swappiness` não for especificado, será usado um valor padrão de 60. Se nenhum valor for especificado para `maxSwap`, esse parâmetro será ignorado. Se `maxSwap` estiver definido como 0, o contêiner não usará troca. Esse parâmetro mapeia para a opção `--memory-swappiness` para [execução do docker](#).

Considere o seguinte ao usar uma configuração de troca de contêiner.

- O espaço de troca deve ser habilitado e alocado na instância de contêiner para os contêineres usarem.

 Note

As AMIs otimizadas do Amazon ECS não têm a troca habilitada por padrão. É necessário habilitar a troca na instância para usar esse recurso. Para obter mais informações, consulte [Volumes de troca de armazenamento de instâncias](#) no Manual do usuário do Amazon EC2 ou [Como alocar memória para funcionar como espaço de troca em uma instância do Amazon EC2 usando uma partição no meu disco rígido?](#).

- Os parâmetros de espaço de troca só são suportados para definições de trabalho usando recursos do EC2.
- Se os parâmetros `maxSwap` e `swappiness` forem omitidos de uma definição de trabalho, cada contêiner terá um valor de `swappiness` padrão de 60. Além disso, o uso total de troca tem um limite de duas vezes a reserva de memória do contêiner.

 Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Tipo: Inteiro

Obrigatório: Não

`tmpfs`

O caminho do contêiner, as opções de montagem e o tamanho da montagem do `tmpfs`.

Tipo: Matriz de objetos [Tmpfs](#)

 Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Obrigatório: Não

containerPath

O caminho absoluto do arquivo no contêiner onde o volume tmpfs é montado.

Tipo: Sequência

Obrigatório: Sim

mountOptions

A lista de opções de montagem do volume tmpfs.

Valores válidos: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

Tipo: Matriz de strings

Obrigatório: Não

size

O tamanho (em MiB) do volume tmpfs.

Tipo: Inteiro

Obrigatório: Sim

logConfiguration

A especificação de configuração do log para o contêiner.

Esse parâmetro é mapeado LogConfig na seção [Criar um contêiner](#) da [Docker Remote API](#) e na opção `--log-driver` de [execução do docker](#). Por padrão, os contêineres usam o mesmo driver de registro que o daemon do Docker usa. No entanto, o contêiner pode usar um driver de registro diferente do daemon do Docker especificando um driver de log com esse parâmetro na definição de contêiner. Para usar um driver de registro diferente para um contêiner, o sistema de registro deve ser configurado na instância de contêiner ou em outro servidor de registro para fornecer opções de registro remoto. Para obter mais informações sobre as opções para drivers de log compatíveis diferentes, consulte [Configure logging drivers](#) na documentação do Docker.

Note

No momento, o AWS Batch é compatível com um subconjunto dos drivers de registro em log disponíveis para o daemon do Docker (mostrado no tipo de dados [LogConfiguration](#)).

Este parâmetro requer a versão 1.18 da Docker Remote API ou posterior em sua instância de contêiner. Para verificar a versão da Docker Remote API na instância de contêiner, faça login na instância de contêiner e execute o seguinte comando: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
  "devices": [
    {
      "logDriver": "string",
      "options": {
        "optionName1" : "optionValue1",
        "optionName2" : "optionValue2"
      }
      "secretOptions": [
        {
          "name" : "secretOptionName1",
          "valueFrom" : "secretOptionArn1"
        },
        {
          "name" : "secretOptionName2",
          "valueFrom" : "secretOptionArn2"
        }
      ]
    }
  ]
}
```

Tipo: Objeto [LogConfiguration](#)

Obrigatório: Não

logDriver

O driver de log a ser usado para o contêiner. Por padrão, o AWS Batch ativa o driver de log `awslogs`. Os valores válidos listados para esse parâmetro são drivers de log com os quais o agente de contêiner do Amazon ECS pode se comunicar por padrão.

Esse parâmetro é mapeado LogConfig na seção [Criar um contêiner](#) da [Docker Remote API](#) e na opção `--log-driver` de [execução do docker](#). Por padrão, os trabalhos usam o mesmo driver de registro que o daemon do Docker usa. No entanto, o trabalho pode usar um driver de registro diferente do daemon do Docker especificando um driver de log com esse parâmetro na definição do trabalho. Se você quiser especificar outro driver de registro em log para um trabalho, o sistema de log deve ser configurado na instância de contêiner no ambiente de computação. Ou, como alternativa, configure-o em outro servidor de log para fornecer opções de registro remoto. Para obter mais informações sobre as opções para drivers de log compatíveis diferentes, consulte [Configure logging drivers](#) na documentação do Docker.

 Note

AWS Batch atualmente é compatível com um subconjunto de drivers de registro disponíveis para o daemon do Docker. Os drivers de log adicionais podem estar disponíveis em versões futuras do agente de contêiner do Amazon ECS.

Os drivers de log compatíveis são `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` e `splunk`.

 Note

Os trabalhos em execução nos recursos do Fargate são restritos aos drivers de log `awslogs` e `splunk`.

Este parâmetro requer a versão 1.18 da Docker Remote API ou posterior em sua instância de contêiner. Para verificar a versão da Docker Remote API na instância de contêiner, faça login na instância de contêiner e execute o seguinte comando: `sudo docker version | grep "Server API version"`

 Note

O agente de contêiner do Amazon ECS que é executado em uma instância de contêiner deve registrar os drivers de registro que estão disponíveis nessa instância com o variável de ambiente `ECS_AVAILABLE_LOGGING_DRIVERS`. Caso contrário, os contêineres colocados nessa instância não poderão usar essas opções de configuração de log. Para obter mais informações, consulte [Configuração do Agente](#)

[de Contêineres do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

awslogs

Especifica o driver de registro em log do Amazon CloudWatch Logs. Para obter mais informações, consulte [Usar o driver de log awslogs](#), e [Driver de registro do Amazon CloudWatch Logs](#) na documentação do Docker.

fluentd

Especifica o driver de registro do Fluentd. Para obter mais informações, incluindo uso e opções, consulte [Driver de registro Fluentd](#) na documentação do Docker.

gelf

Especifica o driver de registro Graylog Extended Format (GELF). Para obter mais informações, incluindo uso e opções, consulte [Driver de registro Graylog Extended Format](#) na documentação do Docker.

journald

Especifica o driver de registro journald. Para obter mais informações, incluindo uso e opções, consulte [Driver de registro Journald](#) na documentação do Docker.

json-file

Especifica o driver de registro de arquivos JSON. Para obter mais informações, incluindo uso e opções, consulte [driver de registro de arquivo JSON](#) na documentação do Docker.

splunk

Especifica o driver de registro do Splunk. Para obter mais informações, incluindo uso e opções, consulte [driver de registro do Splunk](#) na documentação do Docker.

syslog

Especifica o driver de registro em log do syslog. Para obter mais informações, incluindo uso e opções, consulte [driver de registro de log do Syslog](#) na documentação do Docker.

Tipo: Sequência

Obrigatório: Sim

Valores válidos: awslogs | fluentd | gelf | journald | json-file | splunk | syslog

 Note

Se você tiver um driver personalizado não listado acima que você deseja que funcione no agente de contêiner do Amazon ECS, divida o projeto do agente de contêiner do Amazon ECS [disponível no GitHub](#) e personalize-o para funcionar com esse driver. Incentivamos você a enviar solicitações pull para alterações que você gostaria de ter incluído. No entanto, o Amazon Web Services atualmente não oferece suporte para solicitações a execução de cópias modificadas desse software.

options

Opções de configuração de log para enviar para um driver de log personalizado para o contêiner.

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

Tipo: Mapa de string para string

Obrigatório: Não

secretOptions

Um objeto que representa o segredo a ser passado para a configuração de log. Para obter mais informações, consulte [Especificar dados confidenciais](#).

Tipo: Matriz de objeto

Obrigatório: Não

name

O nome da opção do driver de log a ser definida no trabalho.

Tipo: Sequência

Obrigatório: Sim

valueFrom

O nome do recurso da Amazon (ARN) do segredo a ser exposto à configuração de log do contêiner. Os valores compatíveis são o ARN completo do segredo do Secrets Manager ou o ARN completo do parâmetro na SSM Parameter Store.

Note

Se o parâmetro do Parameter Store do SSM existir na mesma região Região da AWS 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.

Tipo: Sequência

Obrigatório: Sim

memory

Este parâmetro está obsoleto, use [resourceRequirements](#) alternativamente.

O número de MiB de memória a ser reservado para o contêiner.

Como exemplo de como usar [resourceRequirements](#), se sua definição de trabalho contiver uma sintaxe semelhante à seguinte.

```
"containerProperties": {  
  "memory": 512  
}
```

A sintaxe equivalente usando [resourceRequirements](#) é a seguinte.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "512"  
    }  
  ]  
}
```

Tipo: Inteiro

Obrigatório: Sim

mountPoints

Os pontos de montagem para volumes de dados no contêiner. Esse parâmetro é mapeado para Volumes na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção `--volume` para [docker run](#).

```
"mountPoints": [  
  {  
    "sourceVolume": "string",  
    "containerPath": "string",  
    "readOnly": true/false  
  }  
]
```

Tipo: Matriz de objeto

Obrigatório: Não

sourceVolume

O nome do volume a ser montado.

Tipo: Sequência

Obrigatório: Sim, quando mountPoints for usado.

containerPath

O caminho no contêiner no qual montar o volume de host.

Tipo: Sequência

Obrigatório: Sim, quando mountPoints for usado.

readOnly

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

Tipo: Booleano

Obrigatório: Não

Padrão: Falso

networkConfiguration

A configuração de rede para trabalhos que estão em execução nos recursos do Fargate. Os trabalhos que são executados nos recursos do EC2 não devem especificar esse parâmetro.

```
"networkConfiguration": {  
  "assignPublicIp": "string"  
}
```

Tipo: Matriz de objeto

Obrigatório: Não

assignPublicIp

Indica se o trabalho tem um endereço IP público. Isso é necessário se o trabalho precisar de acesso externo à rede.

Tipo: Sequência

Valores válidos: ENABLED | DISABLED

Obrigatório: Não

Padrão: DISABLED

privileged

Quando esse parâmetro é verdadeiro, o contêiner recebe permissões elevadas na instância de contêiner host (semelhante ao usuário root). Esse parâmetro é mapeado para Privileged na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção `--privileged` para [docker run](#). Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate. Não o forneça nem o especifique como falso.

```
"privileged": true/false
```

Tipo: Booleano

Obrigatório: Não

readonlyRootFilesystem

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

```
"readonlyRootFilesystem": true/false
```

Tipo: Booleano

Obrigatório: Não

resourceRequirements

O tipo e a quantidade de um recurso a serem atribuídos a um contêiner. Os recursos suportados incluem GPU, MEMORY e VCPU.

```
"resourceRequirements" : [  
  {  
    "type": "GPU",  
    "value": "number"  
  }  
]
```

Tipo: Matriz de objeto

Obrigatório: Não

type

O tipo de recurso a ser atribuído a um contêiner. Os recursos suportados incluem GPU, MEMORY e VCPU.

Tipo: Sequência

Obrigatório: Sim, quando `resourceRequirements` for usado.

value

A quantidade do recurso especificado a ser reservado para o contêiner. Os valores variam de acordo com o `type` especificado.

Tipo="GPU"

O número de GPUs físicas a serem reservadas para o contêiner. O número de GPUs reservadas para todos os contêineres em um trabalho não deve exceder o número de GPUs disponíveis no recurso de computação no qual o trabalho é executado.

Tipo="MEMORY"

O limite rígido (em MiB) de memória a ser apresentado ao contêiner. Caso tente exceder a memória especificada aqui, o contêiner será excluído. Esse parâmetro é mapeado Memory na seção [Criar um contêiner](#) da [Docker Remote API](#) e na opção --memory de [execução do docker](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa. Isso é necessário, mas pode ser especificado em vários lugares para trabalhos paralelos de vários nós (MNP). Ele deve ser especificado para cada nó pelo menos uma vez. Esse parâmetro é mapeado Memory na seção [Criar um contêiner](#) da [Docker Remote API](#) e na opção --memory de [execução do docker](#).

Note

Se você deseja maximizar a utilização de recursos fornecendo aos trabalhos o máximo de memória possível para um determinado tipo de instância, consulte [Gerenciamento de memória de recursos de computação](#).

Para trabalhos executados em recursos do Fargate, value deve corresponder a um dos valores compatíveis. Além disso, os valores VCPU devem ser um dos valores compatíveis com esse valor de memória.

VCPU	MEMORY
0.25 vCPU	512, 1024 e 2048 MiB
0.5 vCPU	1024-4096 MiB em incrementos de 1024 MiB
1 vCPU	2048-8192 MiB em incrementos de 1024 MiB
2 vCPU	4096-16384 MiB em incrementos de 1024 MiB
4 vCPU	8192-30720 MiB em incrementos de 1024 MiB

VCPU	MEMORY
8 vCPU	16384-61440 MiB em incrementos de 4096 MiB
16 vCPU	32768-122880 MiB em incrementos de 8192 MiB

type="VCPU"

O número de vCPUs reservado para o trabalho. Esse parâmetro é mapeado para CpuShares na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção `--cpu-shares` para [execução do docker](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Para recursos do EC2, você deve especificar pelo menos uma vCPU. Isso é necessário, mas pode ser especificado em vários lugares. Ele deve ser especificado para cada nó pelo menos uma vez.

Para trabalhos que são executados em recursos do Fargate, `value` deve corresponder a um dos valores compatíveis, e os valores MEMORY devem ser um dos valores compatíveis para esse valor de VCPU. Os valores compatíveis são 0,25, 0,5, 1, 2, 4, 8 e 16.

O padrão para a cota de contagem de recursos de vCPU sob demanda do Fargate é 6 vCPUs. Para obter mais informações sobre cotas de serviço, consulte [AWS quotas](#) na Referência geral da Amazon Web Services.

Tipo: Sequência

Obrigatório: Sim, quando `resourceRequirements` for usado.

secrets

Os segredos do trabalho que são expostos como variáveis de ambiente. Para obter mais informações, consulte [Especificar dados confidenciais](#).

```
"secrets": [
  {
    "name": "secretName1",
    "valueFrom": "secretArn1"
  },
  {
    "name": "secretName2",
    "valueFrom": "secretArn2"
  }
]
```

```
    ...  
  ]
```

Tipo: Matriz de objeto

Obrigatório: Não

name

O nome da variável de ambiente que contém o segredo.

Tipo: Sequência

Obrigatório: Sim, quando `secrets for` usado.

valueFrom

O segredo a ser exposto ao contêiner. Os valores com suporte são o nome do recurso da Amazon (ARN) completo do segredo do Secrets Manager ou o ARN completo do parâmetro no SSM Parameter Store.

 Note

Se o parâmetro do SSM Parameter Store existir na mesma Região da AWS do trabalho que está sendo iniciado, 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.

Tipo: Sequência

Obrigatório: Sim, quando `secrets for` usado.

ulimits

Uma lista de valores `ulimits` a ser definida no contêiner. Esse parâmetro é mapeado para `Ulimits` na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção `--ulimit` para [docker run](#).

```
"ulimits": [  
  {  
    "name": string,    ...  
  }  
]
```

```
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

Tipo: Matriz de objeto

Obrigatório: Não

name

O type do ulimit.

Tipo: Sequência

Obrigatório: Sim, quando ulimits for usado.

hardLimit

O limite rígido do tipo ulimit.

Tipo: Inteiro

Obrigatório: Sim, quando ulimits for usado.

softLimit

O limite flexível do tipo ulimit.

Tipo: Inteiro

Obrigatório: Sim, quando ulimits for usado.

user

O nome de usuário a ser usado dentro do contêiner. Esse parâmetro é mapeado para User na seção [Criar um contêiner](#) da [Docker Remote API](#) e a opção `--user` para [docker run](#).

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

Tipo: Sequência

Obrigatório: Não

vcpus

Este parâmetro está obsoleto, use [resourceRequirements](#) alternativamente.

O número de vCPUs reservado para o contêiner.

Como exemplo de como usar `resourceRequirements`, se sua definição de trabalho contiver linhas semelhantes a esta:

```
"containerProperties": {  
  "vcpus": 2  
}
```

As linhas equivalentes usadas [resourceRequirements](#) são as seguintes.

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "VCPU",  
      "value": "2"  
    }  
  ]  
}
```

Tipo: Inteiro

Obrigatório: Sim

volumes

Ao registrar uma definição de trabalho, você pode especificar uma lista de volumes que são passados para o daemon do Docker em uma instância de contêiner. Os parâmetros a seguir são permitidos na propriedades de contêiner:

```
"volumes": [  
  {  
    "name": "string",  
    "host": {  
      "sourcePath": "string"  
    },  
    "efsVolumeConfiguration": {  
      "authorizationConfig": {  
        "accessPointId": "string",  

```



```
    "iam": "string"
  },
  "fileSystemId": "string",
  "rootDirectory": "string",
  "transitEncryption": "string",
  "transitEncryptionPort": number
}
}
```

name

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

Tipo: Sequência

Obrigatório: Não

host

O conteúdo do parâmetro `host` determina se o seu volume de dados persiste na instância de contêiner `host` e onde ele é armazenado. Se o parâmetro `host` estiver vazio, o daemon do Docker atribuirá um caminho de `host` para o volume de dados. No entanto, não há garantia de que os dados persistirão após os contêineres associados a eles deixarem de ser executados.

Note

Este parâmetro não é aplicável a trabalhos executados em recursos do Fargate.

Tipo: Objeto

Obrigatório: Não

sourcePath

O caminho na instância de contêiner `host` apresentada ao contêiner. Caso esse parâmetro esteja vazio, o daemon do Docker atribui um caminho `host` para você.

Caso o parâmetro `host` contenha um local de arquivo `sourcePath`, o volume de dados persiste no local especificado na instância de contêiner `host` até você excluí-lo

manualmente. Caso o valor `sourcePath` não exista na instância de contêiner do host, o daemon do Docker o criará. Caso o local exista, o conteúdo da pasta do caminho de origem é exportado.

Tipo: Sequência

Obrigatório: Não

`efsVolumeConfiguration`

Esse parâmetro é especificado quando você estiver usando um sistema de arquivos do Amazon Elastic File System para o armazenamento de tarefas. Para obter mais informações, consulte [Volumes Amazon EFS](#).

Tipo: Objeto

Obrigatório: Não

`authorizationConfig`

Detalhes de configuração de autorização do sistema de arquivamento Amazon EFS.

Tipo: Sequência

Obrigatório: Não

`accessPointId`

O ID do ponto de acesso do Amazon EFS a ser usado. Se um ponto de acesso for especificado, o valor do diretório raiz especificado no `EFSVolumeConfiguration` deve ser omitido ou definido como `/`. Isso impõe o caminho definido no ponto de acesso EFS. Se um ponto de acesso for usado, a criptografia em trânsito deverá ser habilitada em `EFSVolumeConfiguration`. Para mais informações, consulte [Trabalhando com Pontos de Acesso Amazon EFS](#) no Guia de Usuário Amazon Elastic File System.

Tipo: Sequência

Obrigatório: Não

`iam`

Se o perfil do IAM de trabalho do AWS Batch, estabelecido em uma definição de trabalho ao montar o sistema de arquivos do Amazon EFS, deve ou não ser usado. Em caso positivo, a criptografia de trânsito deve estar habilitada em

EFSVolumeConfiguration. Se o parâmetro for omitido, o valor padrão DISABLED será usado. Para obter mais informações, consulte [Uso dos pontos de acesso do Amazon EFS](#).

Tipo: Sequência

Valores válidos: ENABLED | DISABLED

Obrigatório: Não

fileSystemId

A ID do sistema de arquivamento Amazon EFS a ser usada.

Tipo: string

Obrigatório: Não

rootDirectory

O diretório dentro do sistema de arquivamento Amazon EFS a ser montado como diretório raiz dentro do host. Caso esse parâmetro seja omitido, a raiz do volume Amazon EFS será usada. Especificar / tem o mesmo efeito que omitir esse parâmetro. O tamanho máximo é de 4.096 caracteres.

Important

Se um ponto de acesso EFS for especificado em authorizationConfig, o parâmetro do diretório raiz deverá ser omitido ou definido para /. Isso impõe o caminho definido no ponto de acesso do Amazon EFS.

Tipo: Sequência

Obrigatório: Não

transitEncryption

Determina se a criptografia deve ou não ser habilitada para dados do Amazon EFS em trânsito entre o host do Amazon ECS e o servidor do Amazon EFS. A criptografia de trânsito deverá ser habilitada caso a autorização IAM Amazon EFS for usada. Se o parâmetro for omitido, o valor padrão DISABLED será usado. Para mais informações, consulte [Criptografando Dados em Trânsito](#) no Guia de Usuário Amazon Elastic File System.

Tipo: Sequência

Valores válidos: ENABLED | DISABLED

Obrigatório: Não

`transitEncryptionPort`

A porta a ser usada ao enviar dados criptografados entre o host do Amazon ECS e o servidor do Amazon EFS. Caso não especifique uma porta de criptografia em trânsito, a estratégia de seleção de porta usada pelo assistente de montagem Amazon EFS será utilizada. O valor precisa estar compreendido entre 0 e 65.535. Para mais informações, consulte [Auxiliar de Montagem EFS](#) no Guia de Usuário Amazon Elastic File System.

Tipo: Inteiro

Obrigatório: Não

Amazon EKS properties

Um objeto com diversas propriedades específicas de trabalhos baseados no Amazon EKS. Isso não pode ser especificado para definições de trabalho baseados no Amazon ECS.

`podProperties`

As propriedades dos recursos do pod Kubernetes de um trabalho.

Tipo: Objeto [EksPodProperties](#)

Obrigatório: Não

`containers`

As propriedades do contêiner que é usado no pod do Amazon EKS.

Tipo: Objeto [EksContainer](#)

Obrigatório: Não

`args`

Uma matriz de argumentos para o ponto de entrada. Se isso não for especificado, o CMD da imagem do contêiner será usado. Isso corresponde ao membro `args` na parte

do [Entrypoint](#) do [Pod](#) em Kubernetes. As referências de variáveis de ambiente são expandidas usando o ambiente do contêiner.

Se a variável de ambiente referenciada não existir, a referência no comando não será alterada. Por exemplo, se a referência for a "\$ (NAME1)" e a variável de ambiente NAME1 não existir, a string de comando continuará a ser "\$ (NAME1)". \$\$ será substituído por \$ e a string resultante não será expandida. Por exemplo, \$\$ (VAR_NAME) é passado como \$ (VAR_NAME), independentemente da variável de ambiente VAR_NAME existir ou não. Para obter mais informações, consulte [CMD](#) na referência do Dockerfile e [Definir um comando e argumentos para um pod](#) na Kubernetes documentação.

Tipo: Matriz de strings

Obrigatório: Não

command

O ponto de entrada para o contêiner. Isso não é executado dentro de uma shell. Se isso não for especificado, o ENTRYPOINT da imagem do contêiner será usado. As referências de variáveis de ambiente são expandidas usando o ambiente do contêiner.

Se a variável de ambiente referenciada não existir, a referência no comando não será alterada. Por exemplo, se a referência for a "\$ (NAME1)" e a variável de ambiente NAME1 não existir, a string de comando continuará a ser "\$ (NAME1)". \$\$ será substituído por \$ e a string resultante não será expandida. Por exemplo, \$\$ (VAR_NAME) será passado como \$ (VAR_NAME) independentemente da variável de ambiente VAR_NAME existir ou não. O ponto de entrada não pode ser atualizado. Para obter mais informações, consulte [ENTRYPOINT](#) na referência do Dockerfile e [Definir um comando e argumentos para um contêiner](#) e [Entrypoint](#) na Kubernetes documentação.

Tipo: Matriz de strings

Obrigatório: Não

env

As variáveis de ambiente a serem passadas para um contêiner.

 Note

Variáveis de ambiente não podem começar com "AWS_BATCH". Essa convenção de nomenclatura é reservada para as variáveis que o AWS Batch define.

Tipo: matriz de objetos [EKS ContainerEnvironmentVariable](#)

Obrigatório: Não

name

O nome da variável de ambiente.

Tipo: Sequência

Obrigatório: Sim

value

O valor da variável de ambiente.

Tipo: Sequência

Obrigatório: Não

image

A imagem do Docker usada para iniciar um contêiner.

Tipo: Sequência

Obrigatório: Sim

imagePullPolicy

A política de extração de imagens para o contêiner. Os valores compatíveis são Always, IfNotPresent e Never. Por padrão, esse parâmetro é IfNotPresent. No entanto, se a tag :latest for especificada, o padrão será Always. Para obter mais informações, consulte [Atualizando imagens](#) na Kubernetes documentação.

Tipo: Sequência

Obrigatório: Não

name

O nome do contêiner. Se o nome não for especificado, o nome padrão "Default" será usado. Cada contêiner em um pod deve ter um nome exclusivo.

Tipo: Sequência

Obrigatório: Não

resources

O tipo e a quantidade de um recurso a serem atribuídos a um contêiner. Os recursos suportados incluem `memory`, `cpu` e `nvidia.com/gpu`. Para obter mais informações, consulte [Resource de recursos para pods e contêineres](#) na Kubernetes documentação.

Tipo: Objeto [EksContainerResourceRequirements](#)

Obrigatório: Não

limits

O tipo e a quantidade dos recursos a reservar para o contêiner. Os valores variam de acordo com o nome especificado. Os recursos podem ser solicitados usando o `limits` ou os objetos do `requests`.

memória

O limite rígido de memória (em MiB) para o contêiner, usando inteiros, com um sufixo "Mi". Caso o container tente exceder a memória especificada, o contêiner será encerrado. Você deve especificar pelo menos 4 MiB de memória para um trabalho. `memory` pode ser especificada em `limits`, `requests` ou em ambos. Se `memory` for especificado nos dois lugares, o valor especificado em `limits` deverá ser igual ao valor especificado em `requests`.



Note

Para maximizar a utilização de recursos, forneça aos trabalhos o máximo de memória possível para o tipo de instância específico que você está usando. Para saber como, consulte [Gerenciamento de memória de recursos de computação](#).

cpu

O número de CPUs reservado para o contêiner. Os valores devem ser um múltiplo par de 0.25. `cpu` pode ser especificado em `limits`, `requests` ou em ambos. Se `cpu` for especificada em ambos, o valor especificado em `limits` deverá ser no mínimo igual ao valor especificado em `requests`.

nvidia.com/gpu

O número de GPUs reservado para o contêiner. Os valores devem ser um inteiro par. `memory` pode ser especificado em `limits`, `requests` ou em ambos. Se `memory` for especificado nos dois lugares, o valor especificado em `limits` deverá ser igual ao valor especificado em `requests`.

Tipo: Mapa de string para string

Restrições de tamanho do valor: tamanho mínimo 1. Tamanho máximo de 256.

Obrigatório: Não

`requests`

O tipo e a quantidade dos recursos a solicitar para o contêiner. Os valores variam de acordo com o nome especificado. Os recursos podem ser solicitados usando o `limits` ou os objetos do `requests`.

memória

O limite rígido de memória (em MiB) para o contêiner, usando inteiros, com um sufixo "Mi". Caso o container tente exceder a memória especificada, o contêiner será encerrado. Você deve especificar pelo menos 4 MiB de memória para um trabalho. `memory` pode ser especificada em `limits`, `requests` ou em ambos. Se `memory` for especificado em ambos, o valor especificado em `limits` deverá ser igual ao valor especificado em `requests`.

 Note

Se você deseja maximizar a utilização de recursos fornecendo aos trabalhos o máximo de memória possível para um determinado tipo de instância, consulte [Gerenciamento de memória de recursos de computação](#).

`cpu`

O número de CPUs reservado para o contêiner. Os valores devem ser um múltiplo par de 0.25. `cpu` pode ser especificado em `limits`, `requests` ou em ambos. Se `cpu` for especificada em ambos, o valor especificado em `limits` deverá ser no mínimo igual ao valor especificado em `requests`.

`nvidia.com/gpu`

O número de GPUs reservado para o contêiner. Os valores devem ser um inteiro par. `nvidia.com/gpu` pode ser especificado em `limits`, `requests` ou em ambos. Se `nvidia.com/gpu` for especificado em ambos, o valor especificado em `limits` deverá ser igual ao valor especificado em `requests`.

Tipo: Mapa de string para string

Restrições de tamanho do valor: tamanho mínimo 1. Tamanho máximo de 256.

Obrigatório: Não

`securityContext`

O contexto de segurança de um trabalho. Para obter mais informações, consulte [Configure um contexto de segurança para um pod ou contêiner](#) na Kubernetes documentação.

Tipo: Objeto [EKSContainerSecurityContext](#)

Obrigatório: Não

`privileged`

Quando esse parâmetro é `true`, o contêiner recebe permissões elevadas na instância de contêiner host. O nível das permissões é semelhante às permissões do usuário `root`. O valor padrão é `false`. Esse parâmetro é mapeado para a `privileged` política nas [políticas de segurança do pod Privileged](#) na Kubernetes documentação.

Tipo: Booleano

Obrigatório: Não

`readOnlyRootFilesystem`

Quando esse parâmetro é `true`, o contêiner recebe acesso somente para leitura ao sistema de arquivos raiz. O valor padrão é `false`. Esse parâmetro é mapeado para a `ReadOnlyRootFilesystem` política nas [políticas de segurança do pod de sistemas de arquivos e volumes](#) na documentação Kubernetes.

Tipo: Booleano

Obrigatório: Não

runAsGroup

Quando esse parâmetro é especificado, o contêiner é executado como o ID de grupo especificado (gid). Se esse parâmetro não for especificado, o padrão será o grupo que for especificado nos metadados da imagem. Esse parâmetro mapeia para RunAsGroup e MustRunAs política nas [políticas de segurança do pod de grupos e usuários](#) na Kubernetes documentação.

Tipo: Long

Obrigatório: Não

runAsNonRoot

Quando esse parâmetro é especificado, o contêiner é executado como um usuário com um valor de uid diferente de 0. Se esse parâmetro não for especificado, essa regra será imposta. Esse parâmetro mapeia para RunAsUser e MustRunAsNonRoot política nas [políticas de segurança do pod de grupos e usuários](#) na Kubernetes documentação.

Tipo: Long

Obrigatório: Não

runAsUser

Quando esse parâmetro é especificado, o contêiner é executado como o ID de usuário especificado (uid). Se esse parâmetro não for especificado, o padrão será o usuário que for especificado nos metadados da imagem. Esse parâmetro mapeia para RunAsUser e MustRunAs política nas [políticas de segurança do pod de grupos e usuários](#) na Kubernetes documentação.

Tipo: Long

Obrigatório: Não

volumeMounts

O volume é montado para um contêiner para um trabalho do Amazon EKS. Para obter mais informações sobre volumes e montagens de volumes em Kubernetes, consulte [Volumes](#) na Kubernetes documentação.

Tipo: Matriz de objetos [EKSContainerVolumeMount](#)

Obrigatório: Não

`mountPath`

O caminho no contêiner em que o volume está montado.

Tipo: Sequência

Obrigatório: Não

`name`

O nome do volume a ser montado. Isso deve corresponder ao nome de um dos volumes no pod.

Tipo: Sequência

Obrigatório: Não

`readOnly`

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

Tipo: Booleano

Obrigatório: Não

`dnsPolicy`

A política de DNS para o pod. O valor padrão é `ClusterFirst`. Se o parâmetro `hostNetwork` não for especificado, o padrão será `ClusterFirstWithHostNet`. `ClusterFirst` indica que qualquer consulta ao DNS que não corresponda ao sufixo de domínio de cluster configurado será encaminhada para o servidor de nomes upstream herdado do nó. Se nenhum valor foi especificado para `dnsPolicy` na operação da API [RegisterJobDefinition](#), nenhum valor será retornado para `dnsPolicy` pelas operações da API [DescribeJobDefinitions](#) ou [DescribeJobs](#). A configuração de especificação do pod conterá `ClusterFirst` ou `ClusterFirstWithHostNet`, dependendo do valor do parâmetro `hostNetwork`. Para obter mais informações, consulte [política Pod's DNS](#) na Kubernetes documentação.

Valores válidos: `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

Tipo: Sequência

Obrigatório: Não

`hostNetwork`

Indica se o pod usa o endereço IP da rede dos hosts. O valor padrão é `true`. Definir essa opção como `false` ativa o modelo de rede de pods Kubernetes. A maioria das cargas AWS Batch de trabalho é somente de saída e não requer a sobrecarga de alocação de IP para cada pod para conexões de entrada. Para obter mais informações, consulte [Espaços de nomes de Host](#) e [rede de Pods](#) na Kubernetes documentação.

Tipo: Booleano

Obrigatório: Não

`serviceAccountName`

O nome da conta de serviço que é usada para executar o pod. Para obter mais informações, consulte [Kubernetes contas de serviço](#) e [Configure uma conta de serviço Kubernetes para assumir um perfil do IAM](#) no Amazon EKS User Guide e [Configure contas de serviço para pods](#) na Kubernetes documentação.

Tipo: Sequência

Obrigatório: Não

`volumes`

Especifica os volumes para uma definição de trabalho que usa recursos do Amazon EKS.

Tipo: Matriz de objetos [EksVolume](#)

Obrigatório: Não

`emptyDir`

Especifica a configuração de um volume Kubernetes `emptyDir`. Um volume `emptyDir` é criado pela primeira vez quando um pod é atribuído a um nó. Ele existirá enquanto o pod estiver sendo executado nesse nó. O volume `emptyDir` está inicialmente vazio. Todos os contêineres no pod podem ler e gravar os arquivos no volume `emptyDir`. No entanto, o volume `emptyDir` pode ser montado no mesmo caminho ou em caminhos diferentes em cada contêiner. Quando um pod é removido de um nó por qualquer motivo, os dados

no `emptyDir` são excluídos permanentemente. Para obter mais informações, consulte [emptyDir](#) na Kubernetes documentação.

Tipo: Objeto [ekEmptyDir](#)

Obrigatório: Não

medium

A mídia para armazenar o volume. O valor padrão é uma string vazia, que usa o armazenamento do nó.

""

(Padrão) Usar o armazenamento em disco do nó.

"Memory" (Memória)

Usar o volume `tmpfs` que tem o suporte da RAM do nó. O conteúdo do volume é perdido quando o nó é reinicializado e qualquer armazenamento no volume é contabilizado no limite de memória do contêiner.

Tipo: Sequência

Obrigatório: Não

sizeLimit

O tamanho máximo do volume. Por padrão, não há tamanho máximo definido.

Tipo: Sequência

Restrições de tamanho: tamanho mínimo 1. Tamanho máximo de 256.

Obrigatório: Não

hostPath

Especifica a configuração de um volume Kubernetes `hostPath`. Um volume `hostPath` monta um arquivo ou diretório existente do sistema de arquivos do nó host no pod. Para obter mais informações, consulte [hostPath](#) na Kubernetes documentação.

Tipo: Objeto [EKSHostPath](#)

Obrigatório: Não

path

O caminho do arquivo ou diretório no host a ser montado em contêineres no pod.

Tipo: Sequência

Obrigatório: Não

nome

O nome do volume. O nome deve ser permitido como um nome de subdomínio do DNS. Para obter mais informações, consulte [Nomes de subdomínios DNS](#) na Kubernetes documentação.

Tipo: Sequência

Obrigatório: Sim

secret

Especifica a configuração de um volume Kubernetes `secret`. Para obter mais informações, consulte [secret](#) na Kubernetes documentação.

Tipo: Objeto [EksSecret](#)

Obrigatório: Não

optional

Especifica se o segredo ou as chaves do segredo devem ser definidas.

Tipo: Booleano

Obrigatório: Não

secretName

O nome do segredo. O nome deve ser permitido como um nome de subdomínio do DNS. Para obter mais informações, consulte [Nomes de subdomínios DNS](#) na Kubernetes documentação.

Tipo: Sequência

Obrigatório: Sim

Funcionalidades da plataforma

`platformCapabilities`

As funcionalidades da plataforma necessárias pela definição do cargo. Se nenhum valor for especificado, o padrão será EC2. Para trabalhos executados em recursos do Fargate, FARGATE é especificado.

Note

Se o trabalho for executado em recursos do Amazon EKS, você não deve especificar `platformCapabilities`.

Tipo: Sequência

Valores válidos: EC2 | FARGATE

Obrigatório: Não

Propagar tags

`propagateTags`

Especifica se as tags devem ser propagadas a partir do trabalho ou da definição de trabalho para a tarefa correspondente do Amazon ECS. Se nenhum valor for especificado, as tags não serão propagadas. As tags só podem ser propagadas para as tarefas quando as tarefas são criadas. Para tags com o mesmo nome, as tags de trabalho têm prioridade sobre as tags de definições de trabalho. Se o número total de tags combinadas do trabalho e da definição do trabalho for superior a 50, o trabalho será movido para o estado FAILED.

Note

Se o trabalho for executado em recursos do Amazon EKS, você não deve especificar `propagateTags`.

Tipo: Booleano

Obrigatório: Não

Propriedades de nó

nodeProperties

Ao registrar uma definição de trabalho paralela de vários nós, você deve especificar uma lista de propriedades de nós. Essas propriedades do nó definem o número de nós a serem usados em seu trabalho, o índice do nó principal e os diferentes intervalos de nós a serem usados. Se o trabalho for executado em recursos do Fargate, não especifique `nodeProperties`. Em seu lugar, use `containerProperties`. As seguintes propriedades de nó são permitidas em uma definição de trabalho. Para obter mais informações, consulte [Trabalhos em paralelo de vários nós](#).

Note

Se o trabalho for executado em recursos do Amazon EKS, você não deve especificar `nodeProperties`.

Tipo: Objeto [NodeProperties](#)

Obrigatório: Não

`mainNode`

Especifica o índice de nó para o nó principal de um trabalho em paralelo de vários nós. Esse valor de índice do nó deve ser menor que o número de nós.

Tipo: Inteiro

Obrigatório: Sim

`numNodes`

O número de nós que são associados a um trabalho em paralelo de vários nós.

Tipo: Inteiro

Obrigatório: Sim

`nodeRangeProperties`

Uma lista de intervalos de nós e suas propriedades que são associadas a uma trabalho em paralelo de vários nós.

 Note

Um grupo de nós é um grupo idêntico de nós de trabalhos que compartilham as mesmas propriedades do contêiner. Você pode usar o AWS Batch para especificar até cinco grupos de nós distintos para cada trabalho.

Tipo: Matriz de objetos [NodeRangeProperty](#)

Obrigatório: Sim

`targetNodes`

O intervalo de nós, usando valores de índice de nó. Um intervalo de `0:3` indica nós com valores de índice de `0` a `3`. Se o valor do intervalo inicial for omitido (`:n`), então `0` será usado para iniciar o intervalo. Se o valor do intervalo final for omitido (`n:`), o índice de nó mais alto possível será usado para finalizar o intervalo. Seus intervalos de nós acumulativos devem considerar todos os nós (`0:n`). Você pode aninhar intervalos de nós (por exemplo, `0:10` e `4:5`). Nesse caso, as propriedades do intervalo `4:5` substituem as propriedades de `0:10`.

Tipo: Sequência

Obrigatório: Não

`container`

Os detalhes do contêiner para o intervalo de nós. Para obter mais informações, consulte [Propriedades do contêiner](#).

Tipo: objeto [ContainerProperties](#)

Obrigatório: Não

Estratégia de repetição

`retryStrategy`

Ao registrar uma definição de trabalho, você pode opcionalmente especificar uma estratégia de nova tentativa a ser usada para trabalhos com falha que são enviados com essa definição de trabalho. Qualquer estratégia de repetição especificada durante uma operação [SubmitJob](#)

substitui a estratégia de repetição definida aqui. Por padrão, cada trabalho é tentado uma vez. Se você especificar mais de uma tentativa, o trabalho será repetido se falhar. Exemplos de uma tentativa de falha incluem o trabalho retornar um código de saída diferente de zero ou a instância de contêiner ser encerrada. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).

Tipo: Objeto [RetryStrategy](#)

Obrigatório: Não

`attempts`

O número de vezes para mover um trabalho para o status `RUNNABLE`. Você pode especificar entre 1 e 10 tentativas. Se `attempts` for maior que um, o trabalho será repetido essa quantidade de vezes se falhar, até ser movido para `RUNNABLE`.

```
"attempts": integer
```

Tipo: inteiro

Obrigatório: Não

`evaluateOnExit`

Matriz de até 5 objetos que especificam condições sob as quais o trabalho deve ser repetido ou considerado com falha. Se esse parâmetro for especificado, o parâmetro `attempts` também deverá ser especificado. Se `evaluateOnExit` for especificado, mas nenhuma das entradas corresponder, o trabalho será repetido.

```
"evaluateOnExit": [  
  {  
    "action": "string",  
    "onExitCode": "string",  
    "onReason": "string",  
    "onStatusReason": "string"  
  }  
]
```

Tipo: Matriz de objetos [EvaluateOnExit](#)

Obrigatório: Não

action

Especifica a ação a ser executada se todas as condições especificadas (onStatusReason, onReason, e onExitCode) forem atendidas. Os valores não diferenciam maiúsculas e minúsculas

Tipo: Sequência

Obrigatório: Sim

Valores válidos: RETRY | EXIT

onExitCode

Contém um padrão glob para corresponder à representação decimal do ExitCode retornado para um trabalho. O padrão pode ter até 512 caracteres de comprimento. Ele pode conter somente números. Ele não pode conter letras ou caracteres especiais. Ele pode, opcionalmente, terminar com um asterisco (*) para que apenas o início da string precise ser uma correspondência exata.

Tipo: Sequência

Obrigatório: Não

onReason

Contém um padrão glob para corresponder ao Reason retornado para um trabalho. O padrão pode ter até 512 caracteres de comprimento. Ele pode conter letras, números, pontos (.), dois pontos (:) e espaço em branco (incluindo espaços ou tabulações). Ele pode, opcionalmente, terminar com um asterisco (*) para que apenas o início da string precise ser uma correspondência exata.

Tipo: Sequência

Obrigatório: Não

onStatusReason

Contém um padrão glob para corresponder ao StatusReason retornado para um trabalho. O padrão pode ter até 512 caracteres de comprimento. Ele pode conter letras, números, pontos (.), dois pontos (:) e espaço em branco (incluindo espaços ou tabulações). Ele pode, opcionalmente, terminar com um asterisco (*) para que apenas o início da string precise ser uma correspondência exata.

Tipo: Sequência

Obrigatório: Não

Prioridade de agendamento

`schedulingPriority`

A prioridade de agendamento para trabalhos enviados com essa definição de trabalho. Isso afeta apenas os trabalhos nas filas de trabalhos com uma política de compartilhamento justo. Trabalhos com prioridade de agendamento mais alta são agendados antes de trabalhos com prioridade de agendamento mais baixa.

O valor mínimo compatível é 0 e o valor máximo compatível é 9999.

Tipo: Inteiro

Obrigatório: Não

Tags

`tags`

Etiquetas de pares de valores-chave a serem associadas à definição do trabalho. Para obter mais informações, consulte [Marcar com tag os recursos do AWS Batch](#).

Tipo: Mapa de string para string

Obrigatório: Não

Tempo limite

`timeout`

Você pode configurar uma duração de tempo limite para os seus trabalhos para que, caso uma execução de trabalho seja maior, AWS Batch possa encerrar o trabalho. Para obter mais informações, consulte [Tempos limite de trabalho](#). Se um trabalho for encerrado devido a um tempo limite, ele não será tentado novamente. Qualquer configuração de tempo limite especificada durante uma operação [SubmitJob](#) substituirá a configuração de tempo limite definida aqui. Para obter mais informações, consulte [Tempos limite de trabalho](#).

Tipo: Objeto [JobTimeout](#)

Obrigatório: Não

`attemptDurationSeconds`

O tempo de duração em segundos (medido a partir do registro de data e hora `startedAt` da tentativa de trabalho) após o encerramento de trabalhos inacabados AWS Batch. O valor mínimo do tempo limite é 60 segundos.

Em trabalhos de matriz, o tempo limite se aplica aos trabalhos secundários, não ao trabalho de matriz principal.

Para trabalhos paralelos de vários nós (MNP), o tempo limite se aplica ao trabalho inteiro, não aos nós individuais.

Tipo: Inteiro

Obrigatório: Não

Crie definições de trabalhos usando `EcsProperties`

Com definições de trabalho do AWS Batch usando [EcsProperties](#), você pode modelar componentes de hardware, sensores, ambientes 3D e outras simulações em contêineres separados. É possível usar esse recurso para organizar logicamente componentes de workload e separá-los da aplicação principal. Esse recurso pode ser usado com o AWS Batch no Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) e AWS Fargate.

Definições de trabalho `ContainerProperties` versus `EcsProperties`

Você pode optar por usar [ContainerProperties](#) ou definições de trabalho [EcsProperties](#) de acordo com seu caso de uso. Em alto nível, executar trabalhos do AWS Batch com o `EcsProperties` é semelhante a executar trabalhos com um `ContainerProperties`.

A estrutura antiga de definição de tarefas, usando `ContainerProperties`, permanece compatível. Se você já tem fluxos de trabalho usando essa estrutura, pode continuar a executá-los.

A principal diferença é que há um novo objeto adicionado à definição do trabalho para acomodar as definições com base em `EcsProperties`.

Por exemplo, uma definição de trabalho que usa o `ContainerProperties` no Amazon ECS e no Fargate tem esta estrutura:

```
{
  "containerProperties": {
    ...
    "image": "my_ecr_image1",
    ...
  },
  ...
}
```

Uma definição de trabalho que usa o `EcsProperties` no Amazon ECS e no Fargate tem esta estrutura:

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
          ...
        },
      ],
    }
  ]
}
```

Alterações gerais nas APIs do AWS Batch

Descrevemos algumas das principais diferenças ao usar `ContainerProperties` e os tipos de dados da API `EcsProperties`:

- Muitos dos parâmetros usados em `ContainerProperties` aparecem em `TaskContainerProperties`. Alguns exemplos incluem `command`, `image`, `privileged`, `secrets` e `users`. Todos eles podem ser encontrados em [TaskContainerProperties](#).
- Alguns dos parâmetros `TaskContainerProperties` não têm equivalentes funcionais na estrutura legada. Alguns exemplos incluem `dependsOn`, `essential`, `name`, `ipcMode` e `pidMode`. Para obter mais informações, consulte [EcstTaskDetails](#) e [TaskContainerProperties](#).

Além disso, alguns parâmetros `ContainerProperties` não têm equivalentes ou aplicação na estrutura `EcsProperties`. Em [taskProperties](#), `container` foi substituído por `containers`, para que o novo objeto possa aceitar até dez elementos. Para obter mais informações, consulte [RegisterJobDefinition:containerProperties](#) e [EcsTaskProperties:containers](#).

- `taskRoleArn` é funcionalmente equivalente a `jobRoleArn`. [Para obter mais informações, consulte EcsTaskProperties:taskRoleArn e ContainerProperties:jobRoleArn.](#)
- Você pode incluir de um (1) a dez (10) contêineres na estrutura `EcsProperties`. Para obter mais informações, consulte [EcsTaskProperties:containers](#).
- Os objetos `taskProperties` e `instanceTypes` são matrizes, mas aceitam apenas um elemento. Por exemplo, [EcsProperties:taskProperties](#) e [NodeRangeProperty:instanceTypes](#).

Definições de trabalho de vários contêineres para o Amazon ECS

Para acomodar a estrutura de vários contêineres do Amazon ECS, alguns dos tipos de dados da API são diferentes. Por exemplo,

- [ecsProperties](#) é o mesmo nível de `containerProperties` na definição de contêiner único. Para obter mais informações, consulte [EcsProperties](#), no Guia de referência da API do AWS Batch.
- [taskProperties](#) contém as propriedades definidas para a tarefa do Amazon ECS. Para obter mais informações, consulte [EcsProperties](#), no Guia de referência da API do AWS Batch.
- [containers](#) inclui informações semelhantes a `containerProperties` na definição de contêiner único. A principal diferença é que `containers` permite definir até dez contêineres. Para obter mais informações, consulte [ECSTaskProperties:containers](#) no Guia de referência de APIs do AWS Batch.
- O parâmetro [essential](#) indica como o contêiner afeta o trabalho. Todos os contêineres essenciais devem ser concluídos com sucesso (encerrar como 0) para que o trabalho possa progredir. Se um contêiner marcado como essencial falhar (encerrar como diferente de 0), o trabalho falhará.

O valor padrão é `true`, e pelo menos um contêiner deve ser marcado como `essential`. Para obter mais informações, consulte [essential](#) no AWS Batch API Reference Guide (Guia de referência da API do &CDSlong;).

- Com o parâmetro [dependsOn](#), você pode definir uma lista de dependências de contêineres. Para obter mais informações, consulte [dependsOn](#) no AWS Batch API Reference Guide (Guia de referência da API do &CDSlong;).

 Note

A complexidade da lista `dependsOn` e do runtime do contêiner associado podem afetar o horário de início do seu trabalho. Se dependências demorarem muito para serem executadas, o trabalho permanecerá em estado `STARTING` até que seja concluído.

[Para obter mais informações sobre `ecsProperties` e a estrutura, consulte a sintaxe de solicitação `RegisterJobDefinition` para `ecsProperties`.](#)

Definições de trabalho de vários contêineres para o Amazon EKS

Para acomodar a estrutura de vários contêineres do Amazon EKS, alguns dos tipos de dados da API são diferentes. Por exemplo,

- [name](#) é um identificador exclusivo para o contêiner. Esse objeto não é necessário para um único contêiner, mas é necessário ao definir vários contêineres em um pod. Quando `name` não está definido para contêineres únicos, o nome padrão, `default`, é aplicado.
- [initContainers](#) são definidos no tipo de dados [eksPodProperties](#). Eles são executados antes dos contêineres da aplicação, sempre são executados até a conclusão e devem ser concluídos com êxito antes do início do próximo contêiner.

Esses contêineres são registrados no agente Amazon EKS Connector e mantêm as informações de registro no armazenamento de dados de backend do Amazon Elastic Kubernetes Service. O objeto `initContainers` pode aceitar até dez (10) elementos. Para obter mais informações, consulte [Contêineres de inicialização](#), na documentação do Kubernetes.

 Note

O objeto `initContainers` pode afetar a hora de início do seu trabalho. Se `initContainers` demorarem muito para serem executados, o trabalho permanecerá em estado `STARTING` até a conclusão.

- [shareProcessNamespace](#) indica se os contêineres no pod podem compartilhar o mesmo namespace de processo. O valor padrão é `false`. Defina isso como `true` para permitir que os contêineres vejam e sinalizem processos em outros contêineres localizados no mesmo pod.
- Cada contêiner tem importância. Todos os contêineres devem ser concluídos com êxito (sair como 0) para que o trabalho seja bem-sucedido. Se um contêiner falhar (sair como diferente de 0), o trabalho falhará.

Para obter mais informações sobre o `eksProperties` e a estrutura, consulte a sintaxe de solicitação [RegisterJobDefinition](#) para [eksProperties](#).

Referência: cenários de trabalho do AWS Batch usando EcsProperties

Para ilustrar como as definições de trabalho do AWS Batch que usam `EcsProperties` podem ser estruturadas com base nas suas necessidades, este tópico apresenta as seguintes cargas úteis de [RegisterJobDefinition](#). É possível copiar esses exemplos em um arquivo, personalizá-los de acordo com suas necessidades e, em seguida, usar o AWS Command Line Interface (AWS CLI) para chamar `RegisterJobDefinition`.

Trabalho do AWS Batch para o Amazon ECS no Amazon EC2

Veja a seguir um exemplo de um trabalho do AWS Batch para o Amazon Elastic Container Service no Amazon Elastic Compute Cloud:

```
{
  "jobDefinitionName": "multicontainer-ecs-ec2",
  "type": "container",
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
          {
            "name": "c1",
            "essential": false,
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
```

```
        "type": "VCPU",
        "value": "2"
    },
    {
        "type": "MEMORY",
        "value": "4096"
    }
]
},
{
    "name": "c2",
    "essential": false,
    "command": [
        "echo",
        "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "resourceRequirements": [
        {
            "type": "VCPU",
            "value": "2"
        },
        {
            "type": "MEMORY",
            "value": "4096"
        }
    ]
},
{
    "name": "c3",
    "essential": true,
    "command": [
        "echo",
        "hello world"
    ],
    "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
    "firelensConfiguration": {
        "type": "fluentbit",
        "options": {
            "enable-ecs-log-metadata": "true"
        }
    },
    "resourceRequirements": [
        {
```

```
        "type": "VCPU",  
        "value": "6"  
    },  
    {  
        "type": "MEMORY",  
        "value": "12288"  
    }  
]  
}  
]  
}  
]  
}
```

Trabalho do AWS Batch para o Amazon ECS no Fargate

Veja a seguir um exemplo de um trabalho do AWS Batch para o Amazon Elastic Container Service no AWS Fargate:

```
{
  "jobDefinitionName": "multicontainer-ecs-fargate",
  "type": "container",
  "platformCapabilities": [
    "FARGATE"
  ],
  "ecsProperties": {
    "taskProperties": [
      {
        "containers": [
          {
            "name": "c1",
            "command": [
              "echo",
              "hello world"
            ],
            "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
            "resourceRequirements": [
              {
                "type": "VCPU",
                "value": "2"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```

        "type": "MEMORY",
        "value": "4096"
    }
  ],
},
{
  "name": "c2",
  "essential": true,
  "command": [
    "echo",
    "hello world"
  ],
  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
  "resourceRequirements": [
    {
      "type": "VCPU",
      "value": "6"
    },
    {
      "type": "MEMORY",
      "value": "12288"
    }
  ]
}
],
"executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
}
]
}
}

```

Trabalho do AWS Batch para o Amazon EKS

Veja a seguir um exemplo de um trabalho do AWS Batch para o Amazon Elastic Kubernetes Service:

```

{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {

```

```
    "name": "init-container",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
        "echo"
    ],
    "args": [
        "hello world"
    ],
    "resources": {
        "requests": {
            "cpu": "1",
            "memory": "512Mi"
        }
    }
},
{
    "name": "init-container-2",
    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
    "command": [
        "echo",
        "my second init container"
    ],
    "resources": {
        "requests": {
            "cpu": "1",
            "memory": "512Mi"
        }
    }
}
],
"containers": [
    {
        "name": "c1",
        "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
        "command": [
            "echo world"
        ],
        "resources": {
            "requests": {
                "cpu": "1",
                "memory": "512Mi"
            }
        }
    }
],
},
```

```

    {
      "name": "sleep-container",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
      "command": [
        "sleep",
        "20"
      ],
      "resources": {
        "requests": {
          "cpu": "1",
          "memory": "512Mi"
        }
      }
    }
  ]
}

```

Trabalho MNP do AWS Batch com vários contêineres por nó

Veja a seguir um exemplo de um trabalho paralelo de vários nós (MNP) do AWS Batch com vários contêineres por nó:

```

{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}

```

```
"image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
"resourceRequirements": [
  {
    "type": "VCPU",
    "value": "2"
  },
  {
    "type": "MEMORY",
    "value": "4096"
  }
],
},
{
  "name": "range05-c2",
  "command": [
    "echo",
    "hello world"
  ],
  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
  "resourceRequirements": [
    {
      "type": "VCPU",
      "value": "2"
    },
    {
      "type": "MEMORY",
      "value": "4096"
    }
  ]
}
]
}
]
}
]
}
]
}
```

Usar o driver de log awslogs.

Por padrão, o AWS Batch permite que o driver de log `awslogs` envie informações de log ao CloudWatch Logs. Você pode usar esse atributo para visualizar logs diferentes dos contêineres

em um local indicado e evita que os logs de contêiner ocupem espaço em disco nas instâncias de contêiner. Este tópico ajuda você a configurar o driver de log `awslogs` em suas definições de trabalho.

 Note

No console do AWS Batch, você pode configurar o driver de log `awslogs` registro na seção de configuração de log ao criar uma definição de trabalho.

 Note

O tipo de informações registradas em log pelos contêineres em seu trabalho depende principalmente do comando `ENTRYPOINT`. Por padrão, os logs capturados mostram a saída do comando que você normalmente vê 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 envia 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.

Para enviar logs de sistema a partir das instâncias de contêiner para o CloudWatch Logs, consulte [Usando CloudWatch registros com AWS Batch](#). Para obter mais informações sobre o CloudWatch Logs, consulte [Monitorar arquivos de log](#) e [Cotas do CloudWatch Logs](#) no Guia do usuário do Amazon CloudWatch Logs.

Opções do driver de log `awslogs` no tipo de dados AWS Batch JobDefinition

O driver de log `awslogs` dá suporte às seguintes opções em definições de trabalho do AWS Batch. Para obter mais informações, consulte [Driver de registro do CloudWatch Logs](#) na documentação do Docker.

`awslogs-region`

Obrigatório: Não

Especifique a região para a qual o driver de log `awslogs` deve enviar os logs do Docker. Por padrão, a região usada é a mesma do trabalho. É possível optar por enviar todos os logs de

trabalhos em regiões distintas para uma única região no CloudWatch Logs. Isso permite que todos sejam visíveis em um único local. Como alternativa, você pode separá-los por região para uma abordagem mais granular. No entanto, quando escolher esta opção, certifique-se de que os grupos de logs especificados existam na região especificada.

`awslogs-group`

Obrigatório: opcional

Com a opção `awslogs-group`, você pode especificar o grupo de logs para o qual o driver de log `awslogs` envia seus fluxos de logs. Se esse campo não for especificado, será usado `aws/batch/job`.

`awslogs-stream-prefix`

Obrigatório: opcional

Com a opção `awslogs-stream-prefix`, é possível associar um fluxo de logs ao prefixo especificado e a ID da tarefa do Amazon ECS do trabalho do AWS Batch à qual o contêiner pertence. Caso você especifique um prefixo com essa opção, o fluxo de log utiliza o seguinte formato:

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

`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. Isso pode ter um impacto negativo na performance 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](#) na documentação do Docker.

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 na performance do registro em log.

`awslogs-create-group`

Obrigatório: Não

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

 Warning

Essa opção não é recomendada. Recomendamos que você crie o grupo de logs com antecedência usando a ação da API [CreateLogGroup](#) do CloudWatch Logs, pois cada trabalho tenta criar o grupo de logs, aumentando a chance de falha do trabalho.

Note

A política do IAM para seu perfil de execução deve incluir a permissão `logs:CreateLogGroup` antes que você tente usar `awslogs-create-group`.

Especificar uma configuração de log na definição de trabalho

Por padrão, o AWS Batch ativa o driver de log `awslogs`. Esta seção descreve como personalizar a configuração de log `awslogs` de um trabalho. Para obter mais informações, consulte [Criar uma definição de tarefa de nó único](#).

Os trechos JSON de configuração de log a seguir têm um objeto `logConfiguration` especificado para cada trabalho. Um é para um trabalho do WordPress que envia logs para um grupo de logs chamado `awslogs-wordpress` e o outro é para um contêiner MySQL que envia logs para um grupo de logs chamado `awslogs-mysql`. Ambos os contêineres usam o prefixo de fluxo de log `awslogs-example`.

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-wordpress",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "awslogs-mysql",
    "awslogs-stream-prefix": "awslogs-example"
  }
}
```

No console do AWS Batch, a configuração de log da definição do trabalho do `wordpress` é especificada conforme mostrado na imagem.

Log configuration

Log driver

awslogs ▼

Options

Name	Value	
awslogs-group ▼	awslogs-wordpress	Remove option
awslogs-stream-prefix ▼	awslogs-example	Remove option

Add option

Secrets

Add secret

Depois que tiver registrado uma definição de tarefa com o driver de log `awslogs` em uma configuração de log de definição de trabalho, você poderá enviar um trabalho com essa definição de trabalho para começar a enviar logs ao CloudWatch Logs. Para obter mais informações, consulte [Tutorial: enviar um trabalho](#).

Especificar dados confidenciais

Com AWS Batch, você pode injetar dados confidenciais em seus trabalhos armazenando seus dados confidenciais em AWS Secrets Manager segredos ou em parâmetros do AWS Systems Manager Parameter Store e, em seguida, referenciá-los na definição de seu trabalho.

Dados sigilosos podem ser expostos a uma tarefa das seguintes formas:

- Para injetar dados confidenciais em seus contêineres como variáveis de ambiente, use o parâmetro `secrets` de definição de tarefa.
- Para fazer referência a informações confidenciais na configuração de log de uma tarefa, use o parâmetro `secretOptions` de definição de trabalho.

Tópicos

- [Especificar dados confidenciais com o Secrets Manager](#)
- [Especificar dados sigilosos com o Systems Manager Parameter Store](#)

Especificar dados confidenciais com o Secrets Manager

Com AWS Batch, você pode injetar dados confidenciais em seus trabalhos armazenando seus dados confidenciais em AWS Secrets Manager segredos e, em seguida, referenciando-os na definição de seu trabalho. Os dados sigilosos armazenados em segredos do Secrets Manager podem ser expostos a um trabalho como variáveis de ambiente, ou como parte da configuração do log.

Ao injetar um segredo como uma variável de ambiente, você pode especificar uma chave JSON ou versão de um segredo a ser injetado. Esse processo ajuda a controlar os dados sigilosos expostos a sua tarefa. Para mais informações sobre versionamento de segredos, consulte [Termos e Conceitos Importantes para AWS Secrets Manager](#) no Guia do Usuário AWS Secrets Manager .

Considerações ao especificar dados sigilosos usando o Secrets Manager

As informações a seguir devem ser consideradas quando o Secrets Manager for usado para especificar dados sigilosos para tarefas.

- Para injetar um segredo usando uma chave JSON específica ou versão de um segredo, sua instância de contêiner no ambiente de computação do agente de contêiner do Amazon ECS instalado deve ser 1.37.0 ou superior. Recomendamos usar a versão mais recente do atendente de contêiner. Para mais informações sobre como verificar a versão do agente e atualizá-la para a versão mais recente, consulte [Atualizando Agente de Contêiner Amazon ECS](#) no Guia do Desenvolvedor Amazon Elastic Container Service.

Para injetar o conteúdo completo de um segredo como variável de ambiente, ou injetar um segredo em uma configuração de registro de log, sua instância do agente de contêiner deve ser versão 1.23.0 ou superior.

- Somente segredos que armazenam dados de texto, que são segredos criados com o `SecretString` parâmetro da [CreateSecret](#) API, são suportados. Segredos que armazenam dados binários, que são segredos criados com o `SecretBinary` parâmetro da [CreateSecret](#) API, não são compatíveis.
- Ao usar uma definição de trabalho que faça referência a segredos Secrets Manager para recuperar dados confidenciais para suas tarefas, caso também esteja usando VPC endpoint de interface, você deverá criar endpoints de VPC de interface para Secrets Manager. Para mais informações,

consulte [Usando Secrets Manager com endpoints VPC](#) no Guia de Usuário AWS Secrets Manager

- Os dados sigilosos são injetados na tarefa quando a mesma é iniciada. Caso o segredo seja subsequentemente atualizado ou alternado, a tarefa não receberá o valor atualizado automaticamente. Você deve iniciar uma nova tarefa para forçar o serviço a iniciá-la com o valor do segredo atualizado.

Permissões do IAM necessárias para AWS Batch segredos

Para este atributo, você precisa possuir a função de execução e referenciá-la em sua definição de trabalho. Isso permite que o agente de contêiner puxe os recursos Secrets Manager necessários. Para mais informações, consulte [AWS Batch Função de execução do IAM](#).

Para prover acesso aos segredos Secrets Manager criados por você, adicione manualmente as permissões a seguir como política em linha ao papel de execução. Para mais informações, consulte [Adicionando e Removendo Políticas do IAM](#) no Guia de Usuário do IAM.

- `secretsmanager:GetSecretValue`— obrigatório caso faça referência a um segredo Secrets Manager.
- `kms:Decrypt`— obrigatório apenas se o seu segredo usar uma chave KMS personalizada, não a chave padrão. O ARN da sua chave personalizada deve ser adicionado como recurso.

O exemplo de política em linha a seguir adiciona as permissões obrigatórias.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-2:777777777777:secret:<secret_name>",

```

```

    "arn:aws:kms:us-east-2:777777777777:key/<key_id>"
  ]
}
]
}

```

Injetando dados sigilosos como uma variável de ambiente

Você pode especificar o seguinte em sua definição de trabalho:

- Objeto `secrets` contendo o nome da variável de ambiente a ser definido no campo
- Nome do recurso da Amazon (ARN) do segredo Secrets Manager
- Parâmetros adicionais contendo dados sigilosos a serem apresentados a tarefa

O exemplo a seguir mostra a sintaxe completa a ser especificada para o segredo Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

A seção a seguir descreve os parâmetros adicionais. Esses parâmetros são opcionais. No entanto, caso opte por não utilizá-los, você precisará incluir dois pontos `:` para valores padrão. Abaixo, exemplos provendo mais contexto.

json-key

Especifica o nome da chave em um par de chave/ valor, com o valor que deseja definir como o valor da variável de ambiente. Somente valores formato JSON são compatíveis. Caso não especifique uma chave JSON, o conteúdo completo do segredo será utilizado.

version-stage

Especifique o rótulo de preparação da versão do segredo que deseja usar. Caso um rótulo de preparação de versão seja especificado, você não poderá especificar um ID da versão. Se nenhum estágio de versão for especificado, o comportamento padrão será recuperar o segredo com o rótulo de preparação `AWSCURRENT`.

Rótulos de preparação são usados para monitoramento de diferentes versões de um segredo quando eles forem atualizados ou alternados. Cada versão de um segredo tem um ou mais

rótulos de preparação e uma ID. Para obter mais informações, consulte [Key Terms and AWS Concepts for Secrets Manager](#) no Guia AWS Secrets Manager do Usuário.

version-id

Especifica o identificador exclusivo da versão do segredo que você deseja usar. Se uma ID da versão for especificada, você não poderá especificar um rótulo de preparação da versão. Se nenhuma ID de versão for especificada, o comportamento padrão será recuperar o segredo com o rótulo de preparação AWSCURRENT.

IDs As versões são usadas para rastrear diferentes versões de um segredo quando elas são atualizadas ou alteradas. Cada versão de um segredo tem uma ID. Para obter mais informações, consulte [Key Terms and AWS Concepts for Secrets Manager](#) no Guia AWS Secrets Manager do Usuário.

Exemplo de definições de contêiner

Os exemplos a seguir mostram maneiras de fazer referência a segredos Secrets Manager nas suas definições de contêiner.

Example Referenciando um segredo completo

O seguinte é um trecho de definição de tarefa mostrando formato, ao fazer referência ao texto completo de um segredo Secrets Manager.

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

Example Referenciando uma chave específica dentro de um segredo

Veja a seguir um exemplo de saída de um get-secret-value comando $\geq$ que exibe o conteúdo de um segredo junto com o rótulo de teste da versão e o ID da versão associado a ele.

```
{
```



```

    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
    "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\",
    \"username3\":\"password3\"}",
    "VersionStages": [
        "AWSCURRENT"
    ],
    "CreateDate": 1581968848.921
}

```

Faça referência a uma chave específica de saída anterior em uma definição de contêiner especificando o nome da chave no fim do ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1:~"
    }]
  }]
}

```

Example Referenciando uma versão secreta específica

O seguinte é um exemplo de saída de comando [>describe-secret](#) que exibe o conteúdo não criptografado de um segredo junto aos metadados de todas as versões do segredo.

```

{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "Description": "Example of a secret containing application authorization data.",
  "RotationEnabled": false,
  "LastChangedDate": 1581968848.926,
  "LastAccessedDate": 1581897600.0,
  "Tags": [],
  "VersionIdsToStages": {
    "871d9eca-18aa-46a9-8785-981dd39ab30c": [
      "AWSCURRENT"
    ],
    "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [

```

```

        "AWSPREVIOUS"
    ]
}

```

Faça referência a um rótulo específico de preparação de versão de saída anterior em uma definição de contêiner especificando o nome da chave no fim do ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::AWSPREVIOUS:"
    }]
  }]
}

```

Faça referência a uma ID da versão específica da saída anterior em uma definição de contêiner especificando o nome da chave no final do ARN.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}

```

Example Referenciando uma chave específica e um rótulo de estágio de versão de um segredo

O seguinte é um exemplo de como fazer referência a uma chave específica dentro de um segredo e rótulo de estágio de versão específico.

```

{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}

```

```
    ]]
  ]]
}
```

Para uma chave específica e ID da versão, use a sintaxe a seguir.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Injetar dados sigilosos em uma configuração de registro em log

Ao especificar um `logConfiguration` na definição de trabalho, você pode especificar `secretOptions` com o nome da opção do driver de log a ser definida no contêiner e o ARN completo do segredo do Secrets Manager que contém os dados confidenciais a serem apresentados ao contêiner.

O seguinte é um trecho de uma definição de trabalho mostrando formato, ao referenciar um segredo Secrets Manager.

```
{
  "containerProperties": [{
    "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"
      }]
    }]
  }]
}
```

Crie um AWS Secrets Manager segredo

Você pode usar o console Secrets Manager para criar um segredo para seus dados sigilosos. Para mais informações, consulte [Criando um Segredo Básico](#) no Guia do Usuário AWS Secrets Manager .

Para criar um segredo básico

Use Secrets Manager para criar um segredo para seus dados sigilosos.

1. Abra o console do Secrets Manager em <https://console.aws.amazon.com/secretsmanager/>.
2. Escolha Armazenar Novo Segredo.
3. Em Selecionar Tipo de Segredo, selecione Outro Tipo de Segredo.
4. Especifique os detalhes do seu segredo personalizado como pares de Chave e Valor. Por exemplo, você pode especificar uma chave de `Username` e, em seguida, fornecer o nome do usuário apropriado como valor. Adicione uma segunda chave com o nome de `Password` e texto da senha como valor. Você também pode adicionar entradas para nome de banco de dados, endereço de servidor ou porta TCP. Adicione quantos pares quiser para armazenar as informações obrigatórias.

Como alternativa, você pode escolher a guia Texto Simples e inserir o valor do segredo como desejar.

5. Escolha a chave de AWS KMS criptografia que você deseja usar para criptografar o texto protegido no segredo. Caso não escolha uma opção, o Secrets Manager verificará a existência de uma chave padrão para a conta e a utilizará, caso exista. Caso não exista uma chave padrão, o Secrets Manager criará uma automaticamente. Você também pode escolher Adicionar Nova Chave para criar uma chave personalizada KMS específica para esse segredo. Para criar sua própria chave do KMS, você deve ter permissões para criar chaves do KMS na sua conta.
6. Escolha Próximo.
7. Em Nome do Segredo, digite nome e caminho opcionais, como **production/MyAwesomeAppSecret** ou **development/TestSecret**, e, em seguida, escolha Avançar. Você pode adicionar opcionalmente uma descrição para ajudá-lo a lembrar o objetivo desse segredo mais tarde.

O nome do segredo deve ter apenas letras ASCII, dígitos, ou qualquer um dos seguintes caracteres: `/_+=.@-`

8. (Opcional) Nesse momento, você poderá configurar a rotação para o seu segredo. Para esse procedimento, deixe em Desabilitar Rotação Automática e escolha Avançar.

Para obter informações sobre como configurar a rotação em segredos novos ou existentes, consulte Como [girar seus AWS Secrets Manager segredos](#).

9. Analise suas configurações e, em seguida, escolha Armazenar Segredo para salvar tudo que inseriu como novo segredo no Secrets Manager.

Especificar dados sigilosos com o Systems Manager Parameter Store

Com AWS Batch, você pode injetar dados confidenciais em seus contêineres armazenando seus dados confidenciais nos parâmetros do AWS Systems Manager Parameter Store e, em seguida, referenciando-os na definição do contêiner.

Tópicos

- [Considerações ao especificar dados confidenciais usando o Systems Manager Parameter Store](#)
- [Permissões do IAM necessárias para AWS Batch segredos](#)
- [Injetar dados sigilosos como uma variável de ambiente](#)
- [Injetar dados sigilosos em uma configuração de registro em log](#)
- [Criar um AWS Systems Manager parâmetro do Parameter Store](#)

Considerações ao especificar dados confidenciais usando o Systems Manager Parameter Store

As informações a seguir devem ser consideradas ao especificar dados confidenciais para contêineres usando parâmetros Systems Manager Parameter Store.

- Esse atributo exige que a instância de contêiner tenha versão 1.23.0 ou acima do agente de contêiner. Recomendamos usar a versão mais recente do atendente de contêiner. Para mais informações sobre como verificar a versão do agente e atualizá-la para a versão mais recente, consulte [Atualizando Agente de Contêiner Amazon ECS](#) no Guia do Desenvolvedor Amazon Elastic Container Service.
- Os dados confidenciais são injetados no contêiner do seu trabalho o mesmo for iniciado. Caso o segredo ou parâmetro Parameter Store seja posteriormente atualizado ou alternado, o contêiner não receberá o valor atualizado automaticamente. Você deve iniciar uma nova tarefa para forçar seu início com segredos atualizados.

Permissões do IAM necessárias para AWS Batch segredos

Para este atributo, você precisa possuir a função de execução e referenciá-la em sua definição de trabalho. Isso permite que o agente de contêineres do Amazon ECS extraia os AWS Systems Manager recursos necessários. Para obter mais informações, consulte [AWS Batch Função de execução do IAM](#).

Para fornecer acesso aos AWS Systems Manager parâmetros do Parameter Store que você cria, adicione manualmente as seguintes permissões como uma política embutida à função de execução. Para mais informações, consulte [Adicionando e Removendo Políticas do IAM](#) no Guia de usuário do IAM.

- `ssm:GetParameters`—Obrigatório se fizer referência a um parâmetro Systems Manager Parameter Store em uma definição de tarefa.
- `secretsmanager:GetSecretValue`—Obrigatório se fizer referência a um segredo Secrets Manager diretamente, ou se o parâmetro Systems Manager Parameter Store fizer referência a um segredo Secrets Manager em uma definição de tarefa.
- `kms:Decrypt`—obrigatório somente se o segredo usar uma chave personalizada KMS e não a chave padrão. O ARN da chave personalizada deve ser adicionado como recurso.

O exemplo de política alinhada a seguir adiciona as permissões necessárias:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-2:999999999999:parameter/<parameter_name>",
        "arn:aws:secretsmanager:us-east-2:999999999999:secret:<secret_name>",
        "arn:aws:kms:us-east-2:999999999999:key/<key_id>"
      ]
    }
  ]
}
```

```

    ]
  }
}

```

Injetar dados sigilosos 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 parâmetro Systems Manager Parameter Store contendo os dados sigilosos a serem apresentados.

O seguinte é um trecho de uma definição de tarefa mostrando formato ao referenciar um parâmetro Systems Manager Parameter Store. Se o parâmetro Systems Manager Parameter Store existir na mesma Região da tarefa sendo iniciada, você poderá usar o ARN completo ou nome do parâmetro. Se o parâmetro existir em uma Região diferente, o ARN completo deverá ser especificado.

```

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

```

Injetar dados sigilosos em uma configuração de registro em log

Em sua definição de contêiner, ao especificar `logConfiguration`, você poderá especificar `secretOptions` com o nome da opção de registro de log a ser definida no contêiner e o ARN completo do parâmetro Systems Manager Parameter Store contendo os dados confidenciais a serem apresentados.

Important

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

O seguinte é um trecho de uma definição de tarefa mostrando formato ao referenciar um parâmetro Systems Manager Parameter Store.

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

Criar um AWS Systems Manager parâmetro do Parameter Store

Você pode usar o AWS Systems Manager console para criar um parâmetro do Systems Manager Parameter Store para seus dados confidenciais. Para mais informações, consulte [Demonstração: Crie e Use um Parâmetro em um Comando \(Console\)](#) no Guia de Usuário AWS Systems Manager .

Para criar um parâmetro Parameter Store

1. Abra o AWS Systems Manager console em <https://console.aws.amazon.com/systems-manager/>.
2. No painel de navegação, escolha Parameter Store, Criar Parâmetro.
3. Para Nome, digite uma hierarquia e nome de parâmetro. Por exemplo, digite test/database_password.
4. Para Descrição, digite uma descrição opcional.
5. Em Tipo, escolha Cadeia de caracteres StringList, ou SecureString.

Note

- Se você escolher SecureString, o campo ID da chave KMS será exibido. Se não fornecer o ID da chave KMS, o ARN da chave KMS, um apelido ou apelido ARN, o sistema usará `alias/aws/ssm`. Essa é a chave KMS padrão para o Systems Manager. Para evitar o uso dessa chave, escolha uma personalizada. Para mais

informações, consulte [Use Parâmetros de Strings Seguros](#) no Guia de Usuário AWS Systems Manager .

- Ao criar um parâmetro de string seguro no console usando o parâmetro `key-id` com um apelido personalizado de chave KMS ou apelido ARN, você deve especificar o prefixo `alias/` antes do apelido. O seguinte é um exemplo de ARN:

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

O seguinte é um exemplo de apelido:

```
alias/MyAliasName
```

6. Em Valor, digite um valor. Por exemplo, `.MyFirstParameter` Se você escolher `SecureString`, o valor será mascarado exatamente como você o inseriu.
7. Escolha Criar Parâmetro.

Autenticação de registro privado para trabalhos

A autenticação de registro privado para uso de trabalhos AWS Secrets Manager permite que você armazene suas credenciais com segurança e, em seguida, faça referência a elas na definição de seu trabalho. Isso fornece uma forma de referenciar imagens de contêiner que existem em registros privados fora dos quais AWS é necessária autenticação em suas definições de trabalho. Esse recurso é suportado por trabalhos hospedados em EC2 instâncias da Amazon e no Fargate.

Important

Se a definição de trabalho faz referência a uma imagem armazenada no Amazon ECR, esse tópico não se aplica. Para obter mais informações, consulte [Usar imagens do Amazon ECR com o Amazon ECS](#) no Guia do usuário do Amazon Elastic Container Registry.

Para trabalhos hospedados em EC2 instâncias da Amazon, esse recurso requer uma versão 1.19.0 ou posterior do agente de contêiner. Recomendamos usar a versão mais recente do atendente de contêiner. Para mais informações sobre como verificar a versão do agente e atualizá-la para a versão mais recente, consulte [Atualizando Agente de Contêiner Amazon ECS](#) no Guia do Desenvolvedor Amazon Elastic Container Service.

Para trabalhos hospedados no Fargate, esse recurso requer a versão da plataforma 1.2.0 ou posterior. Para obter informações, consulte [Versões da plataforma AWS Fargate Linux](#), no Guia do desenvolvedor do Amazon Elastic Container Service.

Em sua definição do contêiner, especifique o objeto `repositoryCredentials` com os detalhes do segredo que você criou. O segredo que você menciona pode ser de uma conta diferente Região da AWS ou diferente do trabalho que o usa.

Note

Ao usar a AWS Batch API ou o AWS SDK, se o segredo existir da Região da AWS mesma forma que o trabalho que você está iniciando, você poderá usar o ARN completo ou o nome do segredo. AWS CLI Se o segredo existir em outra conta, o ARN completo do segredo deve ser especificado. Ao usar o Console de gerenciamento da AWS, o ARN completo do segredo deve ser especificado sempre.

Veja a seguir um trecho de uma definição de trabalho que mostra os parâmetros necessários.

```
"containerProperties": [
  {
    "image": "private-repo/private-image",
    "repositoryCredentials": {
      "credentialsParameter":
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name"
    }
  }
]
```

Permissões do IAM necessárias para a autenticação de registro privado

A perfil de execução é necessário para usar esse recurso. Isso permite que o agente de contêiner obtenha a imagem do contêiner. Para obter mais informações, consulte [AWS Batch Função de execução do IAM](#).

Para fornecer acesso aos segredos criados por você, adicione as permissões a seguir como uma política em linha ao perfil de execução. Para obter mais informações, consulte [Adicionar e remover políticas do IAM](#).

- `secretsmanager:GetSecretValue`

- `kms:Decrypt`: exigido somente se a chave usar uma chave do KMS personalizada e não a chave padrão. O nome do recurso da Amazon (ARN) da chave personalizada deve ser adicionado como um recurso.

Veja a seguir um exemplo de política em linha que adiciona as permissões.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:123456789012:secret:secret_name",
        "arn:aws:kms:us-east-1:123456789012:key/key_id"
      ]
    }
  ]
}
```

Tutorial: criar um segredo para a autenticação de registro privado

Conclua as etapas a seguir para criar um segredo para suas credenciais de registro privado com AWS Secrets Manager.

Criar um segredo básico

1. Abra o AWS Secrets Manager console em <https://console.aws.amazon.com/secretsmanager/>.
2. Escolha Armazenar Novo Segredo.
3. Em Selecionar Tipo de Segredo, selecione Outro Tipo de Segredo.
4. Selecione Plaintext (Texto simples) e insira suas credenciais de registro privado usando o seguinte formato:

```
{
  "username" : "privateRegistryUsername",
  "password" : "privateRegistryPassword"
}
```

5. Escolha Próximo.
6. Em Secret name (Nome de segredo), insira um nome e caminho opcionais, como **production/MyAwesomeAppSecret** ou **development/TestSecret** e escolha Next (Avançar). Você pode adicionar opcionalmente uma descrição para ajudá-lo a lembrar o objetivo desse segredo mais tarde.

O nome do segredo deve ter somente letras ASCII, números ou qualquer um dos seguintes caracteres: `/_+=.@-`.

7. (Opcional) Nesse momento, será possível configurar a rotação para o segredo. Para esse procedimento, deixe em Desabilitar Rotação Automática e escolha Avançar.

Para obter instruções sobre como configurar a rotação em segredos novos ou existentes, consulte Como [gitar seus AWS Secrets Manager segredos](#).

8. Reveja suas configurações e, em seguida, escolha Store secret (Armazenar segredo) para salvar tudo o que você inseriu como novo segredo no Secrets Manager.

Registre uma definição de trabalho e, em Registro privado, ative a Autenticação de registro privado. Em seguida, em ARN ou nome do Secrets Manager, insira o nome do recurso da Amazon (ARN) do segredo. Para obter mais informações, consulte [Permissões do IAM necessárias para a autenticação de registro privado](#).

Volumes Amazon EFS

O Amazon Elastic File System (Amazon EFS) oferece armazenamento simples e escalável de arquivos para uso com suas tarefas AWS Batch. Com Amazon EFS, a capacidade de armazenamento é elástica. Ela é escalada automaticamente a medida que arquivos são adicionados e removidos. Suas aplicações podem ter o armazenamento de que precisarem, quando precisarem.

Use os sistemas de arquivamento Amazon EFS com AWS Batch para exportar dados do sistema de arquivamento em sua frota de instâncias de contêiner. Assim, seus trabalhos têm acesso ao mesmo armazenamento persistente. Contudo, você deve configurar a AMI da instância de contêiner para montar o sistema de arquivos Amazon EFS antes que o daemon Docker seja iniciado. Além disso,

as definições de tarefa devem referenciar montagens de volume na instância de contêiner para poderem usar o sistema de arquivamento. As seções a seguir lhe ajudarão nesse início de uso do Amazon EFS com AWS Batch.

Considerações de volume Amazon EFS

As seguintes informações devem ser consideradas ao usar volumes Amazon EFS:

- Para trabalhos usando recursos EC2, o suporte ao sistema de arquivamento Amazon EFS foi adicionado como visualização prévia pública com a versão AMI otimizada Amazon ECS 20191212 com a versão 1.35.0 do agente de contêiner. No entanto, o suporte ao sistema de arquivamento Amazon EFS tornou-se disponível para o público com a versão AMI otimizada Amazon ECS 20200319 com a versão 1.38.0 do agente de contêiner, que incluía atributos de ponto de acesso Amazon EFS e autorização IAM. Recomendamos o uso da versão otimizada AMI Amazon ECS 20200319 ou superior para aproveitamento desses recursos. Para mais informações, consulte [Versões AMI otimizadas Amazon ECS](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

Note

Caso crie sua própria AMI, você deverá usar o agente de contêiner 1.38.0 ou superior, `ecs-init` versão 1.38.0-1 ou superior e executar os comandos a seguir em sua instância Amazon EC2. Tudo isso para habilitar o plug-in de volume Amazon ECS. Os comandos dependem do uso do Amazon Linux 2 ou Amazon Linux como imagem base.

Amazon Linux 2

```
$ yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
$ yum install amazon-efs-utils
sudo shutdown -r now
```

- Para tarefas utilizando recursos Fargate, o suporte de sistema de arquivamento Amazon EFS foi adicionado ao utilizar a versão 1.4.0 ou superior da plataforma. Para mais informações, consulte [Versões de Plataforma AWS Fargate](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

- Ao especificar volumes Amazon EFS em tarefas usando recursos Fargate, o mesmo cria um contêiner supervisor responsável pelo gerenciamento do volume Amazon EFS. O contêiner supervisor usa uma pequena quantidade de memória da tarefa. O contêiner supervisor fica visível ao consultar a versão 4 do endpoint de metadados de tarefa. Para mais informações, consulte [Endpoint de Metadados de Tarefas Versão 4](#) do Guia de Usuário Amazon Elastic Container Service para AWS Fargate.

Uso dos pontos de acesso do Amazon EFS

Os pontos de acesso Amazon EFS são pontos de entrada específicos da aplicação em um sistema de arquivamento EFS que ajudam a gerenciar o acesso a aplicações a conjuntos de dados compartilhados. Para mais informações sobre pontos de acesso Amazon EFS e como controlar o acesso a eles, consulte [Trabalhando com Pontos de Acesso Amazon EFS](#) no Guia de Usuário Amazon Elastic File System.

Os pontos de acesso podem impor uma identidade de usuário, inclusive grupos POSIX do usuário, para todas as solicitações do sistema de arquivamento feitas por meio do ponto de acesso. Os pontos de acesso também podem impor um diretório raiz diferente para o sistema de arquivamento fazendo com que clientes só possam acessar dados no diretório especificado ou em seus subdiretórios.

Note

Ao criar um ponto de acesso EFS, você especifica um caminho no sistema de arquivamento para servir como diretório raiz. Ao fazer referência ao sistema de arquivos EFS com uma ID de ponto de acesso na definição de trabalho AWS Batch, o diretório raiz deve ser omitido ou definido para / Isso impõe o caminho definido no ponto de acesso EFS.

É possível usar um perfil do IAM de uma tarefa AWS Batch para forçar aplicativos específicos a usarem um ponto de acesso específico. Ao combinar políticas do IAM com pontos de acesso, é possível oferecer acesso fácil e seguro a conjuntos de dados específicos para seus aplicativos. Esse atributo usa perfis do IAM Amazon ECS para funcionalidade de tarefa. Para mais informações, consulte [Funções do IAM para Tarefas](#) no Guia de Desenvolvedor Amazon Elastic Container Service.

Especificação de um sistema de arquivos do Amazon EFS na definição de trabalho

Para usar volumes do sistema de arquivamento Amazon EFS em seus contêineres, você deve especificar as configurações de volume e ponto de montagem na definição de tarefa. O trecho JSON de definição de tarefa a seguir mostra a sintaxe para os objetos `volumes` e `mountPoints` de um contêiner:

```
{
  "containerProperties": [
    {
      "image": "amazonlinux:2",
      "command": [
        "ls",
        "-la",
        "/mount/efs"
      ],
      "mountPoints": [
        {
          "sourceVolume": "myEfsVolume",
          "containerPath": "/mount/efs",
          "readOnly": true
        }
      ],
      "volumes": [
        {
          "name": "myEfsVolume",
          "efsVolumeConfiguration": {
            "fileSystemId": "fs-12345678",
            "rootDirectory": "/path/to/my/data",
            "transitEncryption": "ENABLED",
            "transitEncryptionPort": integer,
            "authorizationConfig": {
              "accessPointId": "fsap-1234567890abcdef1",
              "iam": "ENABLED"
            }
          }
        }
      ]
    }
  ]
}
```

efsVolumeConfiguration

Tipo: Objeto

Obrigatório: Não

Esse parâmetro é especificado quando volumes Amazon EFS são usados.

fileSystemId

Tipo: string

Obrigatório: Sim

A ID do sistema de arquivamento Amazon EFS a ser usada.

rootDirectory

Tipo: string

Obrigatório: Não

O diretório dentro do sistema de arquivamento Amazon EFS a ser montado como diretório raiz dentro do host. Caso esse parâmetro seja omitido, a raiz do volume Amazon EFS será usada. Especificar / tem o mesmo efeito que omitir esse parâmetro. Ele pode ter até 4.096 caracteres de comprimento.

Important

Se um ponto de acesso EFS for especificado em `authorizationConfig`, o parâmetro do diretório raiz deverá ser omitido ou definido para /. Isso impõe o caminho definido no ponto de acesso EFS.

transitEncryption

Tipo: String

Valores válidos: ENABLED | DISABLED

Obrigatório: não

Determina se a criptografia deve ou não ser habilitada para dados Amazon EFS em trânsito entre host AWS Batch e servidor Amazon EFS. A criptografia de trânsito deverá ser habilitada

caso a autorização IAM Amazon EFS for usada. Se o parâmetro for omitido, o valor padrão DISABLED será usado. Para mais informações, consulte [Criptografando Dados em Trânsito](#) no Guia de Usuário Amazon Elastic File System.

transitEncryptionPort

Tipo: inteiro

Obrigatório: Não

A porta a ser usada ao enviar dados criptografados do host AWS Batch para o servidor Amazon EFS. Caso não especifique uma porta de criptografia em trânsito, a estratégia de seleção de porta usada pelo assistente de montagem Amazon EFS será utilizada. O valor precisar estar compreendido entre 0 e 65.535. Para mais informações, consulte [Auxiliar de Montagem EFS](#) no Guia de Usuário Amazon Elastic File System.

authorizationConfig

Tipo: Objeto

Obrigatório: Não

Detalhes de configuração de autorização do sistema de arquivamento Amazon EFS.

accessPointId

Tipo: Sequência

Obrigatório: Não

ID do ponto de acesso a ser usado. Se um ponto de acesso for especificado, o valor do diretório raiz em `efsVolumeConfiguration` deverá ser omitido, ou definido como `/`. Isso impõe o caminho definido no ponto de acesso EFS. Se um ponto de acesso for usado, a criptografia em trânsito deverá ser habilitada em `EFSTransitEncryptionConfiguration`. Para mais informações, consulte [Trabalhando com Pontos de Acesso Amazon EFS](#) no Guia de Usuário Amazon Elastic File System.

iam

Tipo: String

Valores válidos: ENABLED | DISABLED

Obrigatório: Não

Determina se é necessário utilizar o perfil do IAM de tarefa AWS Batch estabelecido em uma definição de tarefa ao montar o sistema de arquivamento Amazon EFS. Em caso positivo, a criptografia de trânsito deve estar habilitada em `EFSVolumeConfiguration`. Se o parâmetro for omitido, o valor padrão `DISABLED` será usado. Para mais informações sobre perfis de execução do IAM, consulte [AWS Batch Função de execução do IAM](#).

Exemplo de definição de trabalhos

Os exemplos de definições de trabalhos a seguir ilustram como usar padrões comuns, como variáveis de ambiente, substituição de parâmetros e montagens do volume.

Conteúdo

- [Variáveis de ambiente](#)
- [Substituição de parâmetros](#)
- [Testar funcionalidade de GPU](#)
- [Trabalho em paralelo de vários nós](#)

Variáveis de ambiente

O exemplo de definição de tarefa a seguir usa variáveis de ambiente para especificar um tipo de arquivo e o URL do Amazon S3. Esse exemplo específico é da postagem do blog de computação [Criar uma tarefa simples "Buscar e executar" do AWS Batch](#). O script [fetch_and_run.sh](#) que é descrito na postagem do blog usa essas variáveis do ambiente para fazer download do script `myjob.sh` do S3 e declarar seu tipo de arquivo.

Embora o comando e as variáveis do ambiente sejam criptografados na definição de tarefa neste exemplo, você pode especificar as substituições de comando e variável de ambiente para tornar a definição de tarefa mais versátil.

```
{
  "jobDefinitionName": "fetch_and_run",
  "type": "container",
  "containerProperties": {
    "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
    "resourceRequirements": [
      {
        "type": "MEMORY",
```

```

        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "myjob.sh",
      "60"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
    "environment": [
      {
        "name": "BATCH_FILE_S3_URL",
        "value": "s3://amzn-s3-demo-source-bucket/myjob.sh"
      },
      {
        "name": "BATCH_FILE_TYPE",
        "value": "script"
      }
    ],
    "user": "nobody"
  }
}

```

Substituição de parâmetros

O exemplo de definição de tarefa a seguir ilustra como permitir a substituição de parâmetros e definir valores padrão.

As declarações `Ref::` na seção `command` são usadas para definir marcadores para a substituição de parâmetros. Quando envia um trabalho com essa definição de tarefa, você especifica as substituições de parâmetro para preencher esses valores, como `inputfile` e `outputfile`. A seção `parameters` a seguir define um padrão para o codec, mas você pode substituir esse parâmetro conforme necessário.

Para obter mais informações, consulte [Parameters](#).

```

{
  "jobDefinitionName": "ffmpeg_parameters",
  "type": "container",

```

```
{
  "parameters": {"codec": "mp4"},
  "containerProperties": {
    "image": "my_repo/ffmpeg",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "2000"
      },
      {
        "type": "VCPU",
        "value": "2"
      }
    ],
    "command": [
      "ffmpeg",
      "-i",
      "Ref::inputfile",
      "-c",
      "Ref::codec",
      "-o",
      "Ref::outputfile"
    ],
    "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
    "user": "nobody"
  }
}
```

Testar funcionalidade de GPU

O exemplo de definição de tarefa a seguir testa se a AMI de workload de GPU descrita em [Usar uma AMI de workload de GPU](#) está configurada corretamente. Este exemplo de definição de trabalho executa o [exemplo](#) do classificador de MNIST profundo do Tensorflow do GitHub.

```
{
  "containerProperties": {
    "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32000"
      },
      {
        "type": "VCPU",
```

```

        "value": "8"
      }
    ],
    "command": [
      "sh",
      "-c",
      "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
    ]
  },
  "type": "container",
  "jobDefinitionName": "tensorflow_mnist_deep"
}

```

É possível criar um arquivo com o texto JSON anterior chamado `tensorflow_mnist_deep.json` e registrar uma definição de trabalho do AWS Batch com o seguinte comando:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Trabalho em paralelo de vários nós

O exemplo de definição de trabalho ilustra um trabalho em paralelo de vários nós. Para obter mais detalhes, consulte [Criar um fluxo de trabalho de dinâmica molecular fortemente acoplado com trabalhos paralelos de vários nós no AWS Batch](#) no blog AWS Compute.

```

{
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
            {

```

```
        "type": "MEMORY",
        "value": "24000"
    },
    {
        "type": "VCPU",
        "value": "8"
    }
],
"command": [],
"jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
"ulimits": [],
"instanceType": "p3.2xlarge"
}
}
]
```

Tarefas

Empregos são a unidade de trabalho iniciada por AWS Batch. Os trabalhos podem ser invocados como aplicações em contêineres executáveis em instâncias de contêiner do Amazon ECS em um cluster do ECS.

Os trabalhos em contêineres podem fazer referência a uma imagem de contêiner, comando e parâmetros de contêiner. Para obter mais informações, consulte [JobDefinition](#).

Você pode enviar um grande número de trabalhos simples e independentes.

Tópicos

- [Tutorial: enviar um trabalho](#)
- [Empregos de serviços em AWS Batch](#)
- [Estados do trabalho](#)
- [AWS Batch variáveis do ambiente de trabalho](#)
- [Repetições de trabalho automatizadas](#)
- [Dependências do trabalho](#)
- [Tempos limite de trabalho](#)
- [Trabalhos do Amazon EKS](#)
- [Trabalhos em paralelo de vários nós](#)
- [Trabalhos paralelos de vários nós no Amazon EKS](#)
- [Trabalhos de matriz](#)
- [Executar trabalhos de GPU](#)
- [Exibir AWS Batch trabalhos em uma fila de trabalhos](#)
- [AWS Batch Procure empregos em uma fila de empregos](#)
- [Modos de rede para AWS Batch trabalhos](#)
- [Exibir registros de AWS Batch trabalhos em CloudWatch Registros](#)
- [Revise as informações do AWS Batch trabalho](#)

Tutorial: enviar um trabalho

Depois de registrar uma definição de trabalho, você pode enviá-la como um trabalho para uma fila de AWS Batch trabalhos. Você pode substituir muitos dos parâmetros especificados no campo de definição de trabalho em runtime.

Para enviar um trabalho

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione o Região da AWS a ser usado.
3. No painel de navegação, escolha Tarefas.
4. Escolha Enviar Novo Trabalho.
5. Em Nome, insira um nome exclusivo para a sua definição de trabalho. Os nomes podem ter até 128 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).
6. Para Definição de Trabalho, escolha uma definição de trabalho existente para o seu trabalho. Para obter mais informações, consulte [Criar uma definição de tarefa de nó único](#).
7. Para Fila de Trabalhos, escolha uma fila de trabalhos existente. Para obter mais informações, consulte [Crie uma fila de trabalhos](#).
8. Em Dependências do Trabalho, escolha Adicionar Dependência do Trabalho.
 - Em ID do Trabalho, insira a ID do trabalho para todas as dependências. Em seguida, escolha Adicionar Dependências do Trabalho. Um trabalho pode ter até 20 dependências. Para obter mais informações, consulte [Dependências do trabalho](#).
9. (Somente para trabalhos de matriz) Em Tamanho da Matriz, especifique um tamanho entre 2 e 10.000.
10. (Opcional) Expanda Tags e então, escolha Adicionar Tag para adicionar tags ao recurso. Insira uma chave e um valor opcional e então escolha Adicionar Tag.
11. Escolha Próxima página.
12. Na seção Substituições de Trabalho:
 - a. (Opcional) Em Prioridade de agendamento, insira um valor de prioridade de agendamento entre 0 e 100. Valores mais altos têm maior prioridade.

- b. (Opcional) Para Tentativas de Trabalho, insira o número máximo de vezes que AWS Batch tenta mover o trabalho para um status `RUNNABLE`. Você pode inserir um número inteiro entre 1 e 10. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).
- c. (Opcional) Em Tempo Limite de Execução, insira o valor do tempo limite (em segundos). O tempo limite de execução é o período de tempo antes que um trabalho não concluído seja encerrado. Se uma tentativa exceder o tempo limite de duração, ela será parada e o status será alterado para `FAILED`. Para obter mais informações, consulte [Tempos limite de trabalho](#). O valor mínimo é 60 segundos.

 Important

Não confie em trabalhos executados com recursos do Fargate por mais de 14 dias. Depois de 14 dias, os recursos do Fargate podem não estar mais disponíveis e é provável que o trabalho ser encerrado.

- d. (Opcional) Ative Propagar Tags para propagar tags do trabalho e da definição de trabalho para a tarefa do Amazon ECS.

13. Expanda Configuração Adicional.

14. (Opcional) Em Repetir condições da estratégia, escolha Adicionar avaliação na saída. Insira pelo menos um valor de parâmetro e escolha uma Ação. Para cada conjunto de condições, Ação deve ser definida como Tentar Novamente ou Sair. Essas ações significam o seguinte:

- Tentar novamente — AWS Batch tenta novamente até que o número de tentativas de trabalho que você especificou seja atingido.
- Sair — AWS Batch para de tentar novamente o trabalho.

 Important

Se você escolher Adicionar Avaliação na Saída, configure pelo menos um parâmetro e uma Ação, ou escolha Remover Avaliação na Saída.

15. Em Parâmetros, escolha Adicionar Parâmetros para adicionar espaços reservados de substituição de parâmetros. Em seguida, insira uma Chave e um Valor opcional.

16. Na seção Substituição de Contêineres:

- a. Em Comando, insira os comandos no campo como o equivalente da matriz de string JSON.

Esse parâmetro é mapeado para Cmd na seção [Criar um contêiner](#) da [Docker Remote API](#) e o parâmetro COMMAND de [docker run](#). Para obter mais informações sobre o CMD parâmetro Docker, consulte <https://docs.docker.com/engine/reference/builder/#cmd>.

 Note

O parâmetro não pode conter uma string vazia.

- b. Para v CPUs, insira o número de v CPUs a ser reservado para o contêiner. Esse parâmetro é mapeado para CpuShares na seção [Criar um Contêiner](#) da [API remota do Docker](#) e a opção --cpu-shares para [docker run](#). Cada vCPU equivale a 1.024 compartilhamentos de CPU. Você deve especificar pelo menos uma vCPU.
- c. Em Memória, insira o limite de memória que está disponível para o contêiner. Caso seu contêiner tente exceder a memória especificada, o mesmo será interrompido. Esse parâmetro é mapeado para Memory na seção [Criar um Contêiner](#) da [API Remota Docker](#) e para a opção --memory para [docker run](#). Você deve especificar pelo menos 4 MiB de memória para uma tarefa.

 Note

Para maximizar o uso dos recursos, priorize a memória para trabalhos de um tipo específico de instância. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).

- d. (Opcional) Em Número de GPUs, escolha o número de GPUs a ser reservado para o contêiner.
- e. (Opcional) Em Variáveis de Ambiente, escolha Adicionar Variável de Ambiente para adicionar variáveis de ambiente como pares de nome/valor. Essas variáveis serão passadas para o contêiner.
- f. Escolha Próxima página.
- g. Em Revisão de Trabalho, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Criar Definição de Trabalho.

Empregos de serviços em AWS Batch

AWS Batch os trabalhos de serviço permitem que você envie solicitações aos AWS serviços por meio de filas de AWS Batch trabalhos. Atualmente, AWS Batch apoia trabalhos SageMaker de treinamento como empregos de serviços. Diferentemente dos trabalhos em contêineres, nos quais AWS Batch gerencia a execução subjacente do contêiner, os trabalhos de serviço AWS Batch permitem fornecer recursos de agendamento e enfileiramento de trabalhos, enquanto o AWS serviço de destino (como a SageMaker IA) lida com a execução real do trabalho.

AWS Batch for SageMaker Training jobs permite que cientistas de dados enviem trabalhos de treinamento com prioridades para filas configuráveis, garantindo que as cargas de trabalho sejam executadas sem intervenção assim que os recursos estiverem disponíveis. Esse recurso trata de desafios comuns, como coordenação de recursos, prevenção de gastos excessivos acidentais, atendimento às restrições orçamentárias, otimização de custos com instâncias reservadas e eliminação da necessidade de coordenação manual entre os membros da equipe.

Os trabalhos de serviço diferem dos trabalhos em contêineres de várias maneiras principais:

- Envio de trabalhos: os trabalhos de serviço devem ser enviados usando a [SubmitServiceJobAPI](#). Os trabalhos de serviço não podem ser enviados pelo console AWS Batch .
- Execução do trabalho: AWS Batch agenda e enfileira os trabalhos de serviço, mas o AWS serviço de destino executa a carga de trabalho real.
- Identificadores de recursos: uso de trabalhos de serviço ARNs que contêm “trabalho de serviço” em vez de “trabalho” para diferenciá-los dos trabalhos em contêineres.

Para começar com trabalhos AWS Batch de serviço para SageMaker treinamento, consulte [the section called “Começando a usar AWS Batch a SageMaker IA”](#).

Tópicos

- [Cargas úteis de trabalho de serviço em AWS Batch](#)
- [Envie um trabalho de serviço em AWS Batch](#)
- [Mapeando AWS Batch o status do trabalho do serviço para o status de SageMaker IA](#)
- [Estratégias de nova tentativa de emprego no serviço em AWS Batch](#)
- [Monitore trabalhos de serviço em uma AWS Batch fila](#)
- [Encerrar trabalhos de serviço](#)

Cargas úteis de trabalho de serviço em AWS Batch

Ao enviar trabalhos de serviço usando [SubmitServiceJob](#), você fornece dois parâmetros principais que definem o trabalho: `serviceJobType` e `serviceRequestPayload`.

- O `serviceJobType` especifica qual AWS serviço executará o trabalho. Para trabalhos SageMaker de treinamento, esse valor é `SAGEMAKER_TRAINING`.
- `serviceRequestPayload` é uma string codificada em JSON que contém a solicitação completa que normalmente seria enviada diretamente para o serviço de destino. Para trabalhos de SageMaker treinamento, essa carga contém os mesmos parâmetros que você usaria com a [CreateTrainingJob](#) API de SageMaker IA.

Para ver uma lista completa de todos os parâmetros disponíveis e suas descrições, consulte a referência da [CreateTrainingJob](#) API SageMaker IA. Todos os parâmetros suportados pelo `CreateTrainingJob` podem ser incluídos na carga útil do seu trabalho de serviço.

Para ver exemplos de mais configurações de tarefas de treinamento [APIs](#), consulte [CLI SDKs](#) e o Guia [SageMaker do Desenvolvedor de IA](#).

Recomendamos usar o PySDK para a criação de trabalhos de serviço porque o PySDK tem classes auxiliares e utilitários. Para ver um exemplo de uso do PySDK, consulte exemplos de [SageMaker IA](#) em GitHub.

Exemplo de carga útil de trabalho de serviço

O exemplo a seguir mostra uma carga útil de um trabalho de serviço simples para um trabalho de SageMaker treinamento que executa um script de treinamento “hello world”:

Essa carga útil seria passada como uma string JSON para o parâmetro `serviceRequestPayload` ao chamar o `SubmitServiceJob`.

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::123456789012:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
```

```
    "hello world"
  ],
},
"ResourceConfig": {
  "InstanceType": "ml.c5.xlarge",
  "InstanceCount": 1,
  "VolumeSizeInGB": 1
},
"OutputDataConfig": {
  "S3OutputPath": "s3://your-output-bucket/output"
},
"StoppingCondition": {
  "MaxRuntimeInSeconds": 30
}
}
```

Envie um trabalho de serviço em AWS Batch

Para enviar trabalhos de serviço para AWS Batch, você usa a [SubmitServiceJob](#) API. Você pode enviar trabalhos usando o SDK AWS CLI ou.

Se ainda não tiver um perfil de execução, você deverá criar um antes de enviar seu trabalho de serviço. Para criar a função de execução de SageMaker IA, consulte [Como usar funções de execução de SageMaker IA](#) no [guia do desenvolvedor de SageMaker IA](#).

Fluxo de trabalho de envio de trabalhos de serviço

Quando você envia um trabalho de serviço, AWS Batch segue este fluxo de trabalho:

1. AWS Batch recebe sua [SubmitServiceJob](#) solicitação e valida os parâmetros AWS Batch específicos. O `serviceRequestPayload` passa sem validação.
2. O trabalho entra no estado SUBMITTED e é colocado na fila de trabalhos especificada
3. AWS Batch avalia se há capacidade disponível no ambiente de serviço para RUNNABLE trabalhos na frente da fila
4. Se a capacidade estiver disponível, o trabalho será transferido para a IA SCHEDULED e o trabalho será passado para a SageMaker IA
5. Quando a capacidade for adquirida e a SageMaker IA fizer o download dos dados do trabalho de serviço, o trabalho de serviço iniciará a inicialização e o trabalho será alterado para STARTING.
6. Quando a SageMaker IA começa a executar o trabalho, seu status é alterado para RUNNING.

7. Enquanto a SageMaker IA executa o trabalho, AWS Batch monitora seu progresso e mapeia os estados do serviço para os estados do AWS Batch trabalho. Para detalhes sobre como os estados dos trabalhos de serviço são mapeados, consulte [???](#)
8. Quando o trabalho de serviço é concluído, ele passa para SUCCEEDED e qualquer saída está pronta para o download.

Pré-requisitos

Antes de enviar um trabalho de serviço, verifique se você tem:

- Ambiente de serviço: um ambiente de serviço que define limites de capacidade. Para obter mais informações, consulte [Criar um ambiente de serviço no AWS Batch](#).
- SageMaker fila de trabalhos — Uma fila de SageMaker trabalhos para fornecer agendamento de trabalhos. Para obter mais informações, consulte [Criar uma fila de trabalhos do SageMaker Training no AWS Batch](#).
- Permissões do IAM: permissões para criar e gerenciar filas de trabalhos AWS Batch e ambientes de serviço. Para obter mais informações, consulte [AWS Batch Políticas, funções e permissões do IAM](#).

Envie um trabalho de serviço com a AWS CLI

Veja a seguir como enviar um trabalho de serviço usando a AWS CLI:

```
aws batch submit-service-job \
  --job-name "my-sagemaker-training-job" \
  --job-queue "my-sagemaker-job-queue" \
  --service-job-type "SAGEMAKER_TRAINING" \
  --service-request-payload '{"TrainingJobName\\": "\\sagemaker-training-job-example\\", "\\AlgorithmSpecification\\": {"TrainingImage\\": "\\123456789012.dkr.ecr.us-east-1.amazonaws.com/pytorch-inference:1.8.0-cpu-py3\\", "\\TrainingInputMode\\": "\\File\\", "\\ContainerEntrypoint\\": [\\"sleep\\", \\"1\\"]}, "\\RoleArn\\": "\\arn:aws:iam::123456789012:role/SageMakerExecutionRole\\", "\\OutputDataConfig\\": {"S3OutputPath\\": "\\s3://example-bucket/model-output/\\", "\\ResourceConfig\\": {"InstanceType\\": "\\m1.m5.large\\", "\\InstanceCount\\": 1, "\\VolumeSizeInGB\\": 1}}}' \
  --client-token "unique-token-12345"
```

Para obter mais informações sobre o parâmetro `serviceRequestPayload`, consulte [the section called “Cargas úteis de trabalho de serviço”](#).

Mapeando AWS Batch o status do trabalho do serviço para o status de SageMaker IA

Quando você envia trabalhos para uma SageMaker fila de trabalhos usando [SubmitServiceJob](#), AWS Batch gerencia o ciclo de vida do trabalho e mapeia os estados do trabalho para [estados de AWS Batch trabalho](#) de SageMaker treinamento equivalentes. Os trabalhos de serviço, como trabalhos de SageMaker treinamento, seguem um ciclo de vida de estado diferente dos trabalhos tradicionais em contêineres. Embora os trabalhos de serviço compartilhem a maioria dos estados com os trabalhos em contêineres, eles introduzem o estado SCHEDULED e apresentam diferentes comportamentos de repetição, principalmente para lidar com erros de capacidade insuficientes do serviço de destino.

A tabela a seguir mostra o estado do AWS Batch trabalho e o SageMaker Status/ SecondaryStatus correspondente:

Status do lote	SageMaker Status primário da IA	SageMaker Status secundário de IA	Description
SUBMITTED	N/D	N/D	Trabalho enviado para a fila, aguardando a avaliação do agendador.
RUNNABLE	N/D	N/D	Trabalho na fila e pronto para agendamento. Os trabalhos nesse estado são iniciados assim que há recursos suficientes no ambiente de serviço. Os trabalhos podem permanecer nesse estado indefinidamente quando não há recursos suficientes.
SCHEDULED	InProgress	Pending	Trabalho de serviço enviado com sucesso à SageMaker AI
STARTING	InProgress	Downloading	SageMaker Trabalho de treinamento baixando dados e imagens. A capacidade da tarefa de treinamento foi adquirida e a inicialização do trabalho começa.

Status do lote	SageMaker Status primário da IA	SageMaker Status secundário de IA	Description
RUNNING	InProgress	Training	SageMaker Algoritmo de execução de tarefas de treinamento
RUNNING	InProgress	Uploading	SageMaker Trabalho de treinamento carregando artefatos de saída após a conclusão do treinamento
SUCCEEDED	Completed	Completed	SageMaker Trabalho de treinamento concluído com sucesso. O envio dos artefatos de saída foi concluído.
FAILED	Failed	Failed	SageMaker O trabalho de treinamento encontrou um erro irreversível.
FAILED	Stopped	Stopped	SageMaker O trabalho de treinamento foi interrompido manualmente <code>StopTrainingJob</code> .

Estratégias de nova tentativa de emprego no serviço em AWS Batch

As estratégias de repetição de tarefas de serviço AWS Batch permitem repetir automaticamente tarefas de serviço com falha sob condições específicas.

Os trabalhos de serviço podem exigir várias tentativas por vários motivos:

- Problemas temporários de serviço: erros internos de serviço, controle de utilização ou interrupções temporárias podem fazer com que os trabalhos falhem durante o envio ou a execução.
- Falhas na inicialização do treinamento: problemas durante a inicialização do trabalho, como problemas de extração de imagens ou erros de inicialização, podem ser resolvidos na nova tentativa.

Ao configurar estratégias de nova tentativa apropriadas, você pode melhorar as taxas de sucesso no trabalho e reduzir a necessidade de intervenção manual, especialmente para workloads de treinamento de longa duração.

Note

Os trabalhos de serviço repetem automaticamente certos tipos de falhas, como erros de capacidade insuficiente, sem consumir suas tentativas de repetição configuradas. Sua estratégia de repetição lida principalmente com outros tipos de falhas, como erros de algoritmo ou problemas de serviço.

Configurar estratégias de repetição

As estratégias de repetição de tarefas de serviço são configuradas usando [ServiceJobRetryStrategy](#), que suporta contagens simples de repetições e lógica de repetição condicional.

Configuração de repetição

A estratégia de repetição mais simples especifica o número de tentativas que devem ser feitas se um trabalho de serviço falhar:

```
{
  "retryStrategy": {
    "attempts": 3
  }
}
```

Essa configuração permite que o trabalho de serviço seja repetido até 3 vezes em caso de falha.

Important

O valor `attempts` representa o número total de vezes que o trabalho pode ser colocado no estado `RUNNABLE`, incluindo a tentativa inicial. Um valor de 3 significa que haverá uma tentativa inicial do trabalho e, em seguida, ele será repetido até 2 vezes adicionais se falhar.

Repita a configuração com evaluateOnExit

Você pode usar o parâmetro `evaluateOnExit` para especificar as condições sob as quais os trabalhos devem ser repetidos ou podem falhar. Isso é útil quando diferentes tipos de falhas exigem tratamento diferente.

A matriz `evaluateOnExit` pode conter até 5 estratégias de repetição, cada uma especificando uma ação (RETRY ou EXIT) e condições com base nos motivos do status:

```
{
  "retryStrategy": {
    "attempts": 5,
    "evaluateOnExit": [
      {
        "action": "RETRY",
        "onStatusReason": "Received status from SageMaker: InternalServerError*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "Received status from SageMaker: ValidationException*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "*"
      }
    ]
  }
}
```

Esta configuração:

- Tenta novamente trabalhos que falham devido a erros internos do servidor de SageMaker IA
- Falha imediatamente trabalhos que encontram exceções de validação (erros do cliente que não serão resolvidos por meio de uma nova tentativa)
- Inclui uma regra abrangente para sair de qualquer outro tipo de falha

Correspondência de padrão do motivo de status

O parâmetro `onStatusReason` oferece suporte à correspondência de padrões com até 512 caracteres. Os padrões podem usar curingas (*) e corresponder aos motivos de status retornados pela SageMaker IA.

Para trabalhos de serviço, as mensagens de status da SageMaker IA são prefixadas com “Status recebido de SageMaker:” para diferenciá-las das mensagens AWS Batch geradas. Os padrões comuns incluem:

- Received status from SageMaker: InternalServerError* - Correspondência com erros internos de serviço
- Received status from SageMaker: ValidationException* - Correspondência com erros de validação do cliente
- Received status from SageMaker: ResourceLimitExceeded* - Correspondência com erros de limite de recursos
- *CapacityError* - Correspondência com falhas relacionadas à capacidade

Tip

Use a correspondência de padrões específicos para lidar com diferentes tipos de erro de forma adequada. Por exemplo, repetir os erros internos do servidor, mas falhar imediatamente os erros de validação que indicam problemas com os parâmetros do trabalho.

Monitore trabalhos de serviço em uma AWS Batch fila

Você pode monitorar o status dos trabalhos em sua fila de trabalhos de SageMaker treinamento usando `list-service-jobs` e `get-job-queue-snapshot`

Veja os trabalhos em execução na sua fila:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status RUNNING
```

Veja os trabalhos em espera na fila:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status RUNNABLE
```

Exibir trabalhos que foram enviados, SageMaker mas ainda não estão em execução:

```
aws batch list-service-jobs \  
  --job-queue my-sm-training-fifo-jq \  
  --job-status SCHEDULED
```

Obtenha um instantâneo dos trabalhos na frente da sua fila:

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Esse comando mostra a ordem dos próximos trabalhos de serviço em sua fila.

Obter informações detalhadas sobre o trabalho de serviço

Use a operação [DescribeServiceJob](#) para obter informações abrangentes sobre um trabalho de serviço específico, incluindo seu status atual, identificadores de recursos de serviço e informações detalhadas sobre tentativas.

Visualizar informações detalhadas sobre um trabalho específico:

```
aws batch describe-service-job \  
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d
```

Esse comando retorna informações abrangentes sobre o trabalho, incluindo:

- ARN do trabalho e status atual
- Identificadores de recursos de serviço (como ARN do trabalho SageMaker de treinamento)
- Prioridade de agendamento e configuração de novas tentativas
- Carga útil da solicitação de serviço contendo os parâmetros originais do serviço
- Informações detalhadas da tentativa com horários de início e término
- Mensagens de status do serviço de destino

Monitore trabalhos SageMaker de treinamento

Ao monitorar os trabalhos de SageMaker treinamento AWS Batch, você pode acessar as informações do AWS Batch trabalho e os detalhes subjacentes do trabalho de SageMaker treinamento.

O identificador do recurso de serviço nos detalhes do trabalho contém o ARN do trabalho de SageMaker treinamento:

```
{
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/my-training-job"
    }
  }
}
```

Você pode usar esse ARN para obter detalhes adicionais diretamente de: SageMaker

```
aws sagemaker describe-training-job \
  --training-job-name my-training-job
```

Monitore o progresso do trabalho verificando AWS Batch o status e o status do trabalho de SageMaker treinamento. O status do AWS Batch trabalho mostra o ciclo de vida geral do trabalho, enquanto o status do trabalho de SageMaker treinamento fornece detalhes específicos do serviço sobre o processo de treinamento.

Encerrar trabalhos de serviço

Use a operação [TerminateServiceJob](#) para interromper um trabalho de serviço em execução.

Encerre um trabalho de serviço específico:

```
aws batch terminate-service-job \
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d \
  --reason "Job terminated by user request"
```

Quando você encerra um trabalho de serviço, AWS Batch interrompe o trabalho e notifica o serviço de destino. Para trabalhos SageMaker de treinamento, isso também interromperá o trabalho de treinamento em SageMaker IA.

Estados do trabalho

Quando você envia um trabalho para uma fila de AWS Batch trabalhos, o trabalho entra no SUBMITTED estado. Em seguida, ele passa pelos seguintes estados, até ter êxito (sai com código 0) ou falhar (saindo com um código diferente de zero). Os trabalhos do AWS Batch podem ter os seguintes estados:

SUBMITTED

Um trabalho que foi enviado para a fila e ainda não foi avaliado pelo programador. O programador avalia o trabalho para determinar se ele tem dependências pendentes na conclusão bem-sucedida de todos os outros trabalhos. Se houver dependências, o trabalho será movido para PENDING. Se não houver dependências, o trabalho será movido para RUNNABLE.

PENDING

Um trabalho que reside na fila e ainda não pode ser executado devido a dependência de outro trabalho ou recurso. Após as dependências serem atendidas, o trabalho é movido para RUNNABLE.

Note

Os pais dos trabalhos de matriz são atualizados para PENDING quando qualquer trabalho secundário é atualizado RUNNABLE e permanecem no PENDING status enquanto os trabalhos secundários estão em execução. Para visualizar esses trabalhos, filtre por PENDING status até que todos os trabalhos secundários atinjam um estado terminal.

RUNNABLE

Um trabalho que reside na fila, não tem dependências pendentes e, portanto, está pronto para ser programado para um host. Os trabalhos nesse estado são iniciados assim que recursos suficientes estão disponíveis em um dos ambientes de computação mapeados para a fila do trabalho. No entanto, os trabalhos podem permanecer nesse estado indefinidamente quando recursos suficientes estiverem indisponíveis.

Note

Se os trabalhos não progredirem para STARTING, consulte [Trabalhos presos no status RUNNABLE](#) na seção de solução de problemas.

STARTING

Esses trabalhos foram programados para um host e as operações de inicialização de contêiner relevantes estão em andamento. Após a imagem de contêiner ser obtida e o contêiner estar em execução, o trabalho ocorre a transição do trabalho para RUNNING.

A duração da extração da imagem, a duração da conclusão de `initContainer` do Amazon EKS e a duração da resolução de `containerDependency` do Amazon ECS ocorrem no estado `STARTING`. O tempo necessário para extrair uma imagem do seu trabalho é equivalente ao tempo em que o trabalho estará no estado `STARTING`.

Por exemplo, se demorar três minutos para extrair a imagem do trabalho, ele ficará no estado `STARTING` por três minutos. Se `initContainers` demorar dez minutos no total para ser concluído, seu trabalho do Amazon EKS ficará em `STARTING` por dez minutos. Se você tiver conjuntos de `containerDependencies` do Amazon ECS no seu trabalho do Amazon ECS, esse trabalho ficará em `STARTING` até que todas as dependências do contêiner (seu runtime) sejam resolvidas. `STARTING` não está incluído nos tempos limite; a duração começa em `RUNNING`. Para obter mais informações, consulte [Estados de trabalhos](#).

RUNNING

O trabalho é executado como um contêiner, em uma instância de contêiner do Amazon ECS em um ambiente de computação. Quando o contêiner do trabalho é encerrado, o código de saída do processo determina se o trabalho foi bem-sucedido ou não. Um código de saída `0` indica êxito, e qualquer código de saída diferente de zero, falha. Se o trabalho associado a uma falha na tentativa tiver tentativas restantes em sua configuração de estratégia de repetição opcional, o trabalho será movido para `RUNNABLE` novamente. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).

Note

Os registros de `RUNNING` trabalhos estão disponíveis em CloudWatch Registros. O grupo de logs é `/aws/batch/job`, e o formato do nome do fluxo de logs, o seguinte: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Esse formato pode mudar no futuro.

Depois que um trabalho atinge o `RUNNING` status, você pode recuperar programaticamente o nome do fluxo de registros com a operação da [DescribeJobs](#) API. Para obter mais informações, consulte [Exibir dados de registro enviados para CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs. Por padrão, esses logs nunca expiram. No entanto, é possível modificar o período de retenção. Para obter mais informações, consulte [Alterar a retenção de dados do log em CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs.

SUCCEEDED

O trabalho foi concluído com êxito com um código de saída 0. O estado do trabalho para SUCCEEDED trabalhos persiste AWS Batch por pelo menos 7 dias.

Note

Os registros de SUCCEEDED trabalhos estão disponíveis em CloudWatch Registros. O grupo de logs é `/aws/batch/job`, e o formato do nome do fluxo de logs, o seguinte: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Esse formato pode mudar no futuro.

Depois que um trabalho atinge o RUNNING status, você pode recuperar programaticamente o nome do fluxo de registros com a operação da [DescribeJobs](#) API. Para obter mais informações, consulte [Exibir dados de registro enviados para CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs. Por padrão, esses logs nunca expiram. No entanto, é possível modificar o período de retenção. Para obter mais informações, consulte [Alterar a retenção de dados do log em CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs.

FAILED

Ocorreu falha na tarefa em todas as tentativas. O estado da tarefa para tarefas FAILED persiste no AWS Batch por pelo menos 7 dias.

Note

Os registros de FAILED trabalhos estão disponíveis em CloudWatch Registros. O grupo de logs é `/aws/batch/job`, e o formato do nome do fluxo de logs, o seguinte: *first200CharsOfJobDefinitionName/default/ecs_task_id*. Esse formato pode mudar no futuro.

Depois que um trabalho atinge o RUNNING status, você pode recuperar programaticamente seu fluxo de registros com a operação da [DescribeJobs](#) API. Para obter mais informações, consulte [Exibir dados de registro enviados para CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs. Por padrão, esses logs nunca expiram. No entanto, é possível modificar o período de retenção. Para obter mais informações, consulte [Alterar a retenção de dados do log em CloudWatch registros](#) no Guia do usuário do Amazon CloudWatch Logs.

AWS Batch variáveis do ambiente de trabalho

AWS Batch define variáveis de ambiente específicas em trabalhos de contêiner. Essas variáveis de ambiente fornecem introspecção para os contêineres dentro dos trabalhos. Você pode utilizar os valores dessas variáveis na lógica de seus aplicativos. Todas as variáveis AWS Batch definidas começam com o `AWS_BATCH_` prefixo. Esse é um prefixo de variável de ambiente protegido. Você não pode usar esse prefixo para suas próprias variáveis, em definições de trabalhos ou substituições.

As seguintes variáveis de ambiente estão disponíveis em contêineres de trabalho:

`AWS_BATCH_CE_NAME`

Essa variável é definida como o nome do ambiente de computação no qual o trabalho é colocado.

`AWS_BATCH_JOB_ARRAY_INDEX`

Essa variável só é definida em trabalhos de matriz filhos. O índice de trabalho de matriz começa com 0, e cada trabalho filho recebe um número de índice exclusivo. Por exemplo, um trabalho de matriz com 10 filhos tem valores de índice de 0 a 9. Você pode utilizar esse valor de índice para controlar como os filhos do trabalho de matriz serão diferenciados. Para obter mais informações, consulte [Usar o índice de trabalho de matriz para controle de diferenciação de trabalhos](#).

`AWS_BATCH_JOB_ARRAY_SIZE`

Essa variável é definida para o tamanho do trabalho da matriz pai. O tamanho do trabalho da matriz pai é passado para o trabalho de matriz filho nessa variável.

`AWS_BATCH_JOB_ATTEMPT`

Essa variável é definida como o número de tentativas de trabalho. A primeira tentativa recebe o número 1. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).

`AWS_BATCH_JOB_ID`

Essa variável é definida como a ID do AWS Batch trabalho.

`AWS_BATCH_JOB_KUBERNETES_NODE_UID`

Essa variável é definida como o UID do Kubernetes do objeto de nó que está no cluster do Kubernetes onde o pod é executado. Essa variável é definida somente para trabalhos executados nos recursos do Amazon EKS. Para obter mais informações, consulte [UIDs](#) na documentação Kubernetes.

AWS_BATCH_JOB_MAIN_NODE_INDEX

Essa variável é definida apenas em tarefas em paralelo de vários nós. Essa variável é definida como o número índice do nó principal do trabalho. O código do seu aplicativo pode comparar o `AWS_BATCH_JOB_MAIN_NODE_INDEX` ao `AWS_BATCH_JOB_NODE_INDEX` em um nó individual para determinar se ele é o nó principal.

AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS

Essa variável é definida apenas em nós filhos de trabalhos em paralelo de vários nós. Essa variável não está presente no nó principal, mas está definida como o IPv4 endereço privado do nó principal do trabalho. O código da aplicação do nó filho pode utilizar esse endereço para comunicar-se com o nó principal.

AWS_BATCH_JOB_NODE_INDEX

Essa variável é definida apenas em tarefas em paralelo de vários nós. Essa variável é definida como o número do índice de nós do nó. O índice do nó começa em 0, e cada nó filho recebe um número índice exclusivo. Por exemplo, uma tarefa em paralelo de vários nós com 10 filhos possui valores índice de 0-9.

AWS_BATCH_JOB_NUM_NODES

Essa variável é definida apenas em tarefas em paralelo de vários nós. Essa variável é definida como o número de nós solicitado para sua tarefa paralela de vários nós.

AWS_BATCH_JQ_NAME

Essa variável é definida como o nome da fila de trabalhos para a qual o trabalho foi enviado.

Repetições de trabalho automatizadas

Você pode aplicar uma estratégia de repetição aos seus trabalhos e a definições de trabalho que permitam que trabalhos com falha sofram novas tentativas automaticamente. Os possíveis cenários de falha incluem os seguintes:

- Qualquer código de saída diferente de zero de um trabalho de contêiner
- Falha de instância ou encerramento do Amazon EC2
- Erro ou interrupção do AWS serviço interno

Quando um trabalho é enviado a uma fila de trabalhos e colocado no estado RUNNING, isso é considerado uma tentativa. Por padrão, cada trabalho tem uma tentativa de movimentação para o estado de trabalho SUCCEEDED ou FAILED. No entanto, tanto a definição de trabalho quanto o fluxo de trabalho do envio de trabalho podem ser utilizados para especificar uma estratégia de repetição entre 1 e 10 tentativas. Se [evaluateOnExit](#) for especificado, ele pode conter até 5 estratégias de repetição. Se [evaluateOnExit](#) for especificado, mas nenhuma das estratégias de repetição corresponder, o trabalho será repetido. Para trabalhos que não correspondam à saída, adicione uma entrada final que saia por qualquer motivo. Por exemplo, esse objeto `evaluateOnExit` tem duas entradas com ações de RETRY e uma entrada final com uma ação de EXIT.

```
"evaluateOnExit": [  
  {  
    "action": "RETRY",  
    "onReason": "AGENT"  
  },  
  {  
    "action": "RETRY",  
    "onStatusReason": "Task failed to start"  
  },  
  {  
    "action": "EXIT",  
    "onReason": "*"   
  }  
]
```

No runtime, a variável de ambiente `AWS_BATCH_JOB_ATTEMPT` é definida como o número de tentativas de trabalho correspondentes do contêiner. A primeira tentativa é numerada 1, e as tentativas subsequentes, em ordem ascendente (por exemplo, 2, 3, 4).

Por exemplo, suponha que uma tentativa de trabalho falhe por algum motivo e o número de tentativas especificado no campo de configuração de nova tentativa seja maior do que o número `AWS_BATCH_JOB_ATTEMPT`. Em seguida, o trabalho será devolvido ao estado RUNNABLE. Para obter mais informações, consulte [Estados do trabalho](#).

Note

Os trabalhos cancelados ou encerrados não são repetidos. Além disso, os trabalhos que falham devido a uma definição de trabalho inválida não são repetidos.

Para obter mais informações, consulte [Estratégia de repetição](#), [Criar uma definição de tarefa de nó único](#), [Tutorial: enviar um trabalho](#) e [Códigos de Erro de Tarefas Interrompidas](#).

Dependências do trabalho

Ao enviar um AWS Batch trabalho, você pode especificar o trabalho do IDs qual o trabalho depende. Ao fazer isso, o AWS Batch agendador garante que seu trabalho seja executado somente após a conclusão bem-sucedida das dependências especificadas. Depois de concluído com êxito, o trabalho dependente transiciona de PENDING para RUNNABLE e, em seguida, para STARTING e RUNNING. Se uma das dependências do trabalho falhar, o trabalho dependente transicionará automaticamente de PENDING para FAILED.

Por exemplo, o trabalho A pode expressar uma dependência de até 20 outros trabalhos, que devem obter êxito antes que ele possa ser executado. Você pode, então, enviar trabalhos adicionais que dependam do trabalho A e de até 19 outros trabalhos.

Para trabalhos de matriz, você pode especificar uma dependência do tipo SEQUENTIAL sem especificar uma ID do trabalho, de forma que cada trabalho de matriz filho seja concluído sequencialmente, a começar pelo índice 0. Você também pode especificar uma dependência do tipo N_TO_N com uma ID do trabalho. Deste modo, cada índice filho dessa tarefa precisa aguardar, para que o índice filho correspondente de cada dependência seja concluído antes de começar. Para obter mais informações, consulte [Trabalhos de matriz](#).

Para enviar um AWS Batch trabalho com dependências, consulte [Tutorial: enviar um trabalho](#).

O [Agendamento com reconhecimento de recurso](#) permite que você agende trabalhos com base nos recursos consumíveis necessários para executar seus trabalhos. Você especifica os recursos consumíveis necessários para a execução de um trabalho e o Batch leva essas dependências de recursos em consideração ao agendar um trabalho. Você pode reduzir a subutilização dos recursos computacionais alocando somente os trabalhos que têm todos os recursos necessários disponíveis. O agendamento com reconhecimento de recursos está disponível para políticas de agendamento FIFO e de compartilhamento justo e pode ser usado com todas as plataformas de computação compatíveis com o Batch, incluindo EKS, ECS e Fargate. Ele pode ser usado com trabalhos de matriz, trabalhos de paralelos de vários nós (MNP) e trabalhos de Batch regulares.

Tempos limite de trabalho

Você pode configurar uma duração de tempo limite para os seus trabalhos para que, caso uma execução de trabalho seja maior, AWS Batch possa encerrar o trabalho. Por exemplo, caso você

tenha um trabalho que sabe que deve demorar apenas 15 minutos para ser concluído. Às vezes, o aplicativo fica preso em um loop e executa para sempre. Nesse caso, você pode definir um tempo limite de 30 minutos para encerrar a tarefa presa.

 Important

Por padrão, AWS Batch não tem um tempo limite de trabalho. Se você não definir um tempo limite do trabalho, o mesmo será executado até a saída do contêiner.

Especifique um parâmetro `attemptDurationSeconds`, que deve ser pelo menos 60 segundos, seja na definição de trabalho ou no envio do trabalho. Quando esse número de segundos tiver passado após o registro de `startedAt` data e hora da tentativa de trabalho, o trabalho AWS Batch será encerrado. No recurso de computação, o contêiner de tarefa recebe um sinal `SIGTERM` para que seu aplicativo possa encerrar normalmente. Se o contêiner ainda estiver em execução após 30 segundos, um sinal `SIGKILL` será enviado para encerrar o contêiner.

Os encerramentos por tempo limite são processados com base no melhor esforço. Você não deve esperar pelo encerramento por tempo limite no momento exato em que a tentativa de trabalho expirar (pode demorar alguns segundos adicionais). Se o aplicativo obrigar a execução do tempo limite exato, você deve implementar essa lógica no aplicativo. Caso tenha uma grande quantidade de tarefas atingindo o tempo limite concomitantemente, os encerramentos por tempo limite se comportarão como uma fila por ordem de chegada, na qual os trabalhos são encerrados em lotes.

 Note

Não há valor máximo de tempo limite para um AWS Batch trabalho.

Se um trabalho for encerrado devido a uma duração de tempo limite excedido, ele não será executado novamente. Se ocorrer uma falha na tentativa de trabalho por si só, ela poderá ser iniciada novamente, caso repetições estejam habilitadas. Além disso, a contagem do tempo limite é reiniciada para a nova tentativa.

 Important

Não se pode esperar que trabalhos executados com recursos Fargate sejam executados por um período maior que 14 dias. Se a duração do tempo limite exceder 14 dias, os recursos Fargate podem não estar mais disponíveis, e o trabalho será encerrado.

Em trabalhos de matriz, os trabalhos filho têm a mesma configuração de tempo limite do trabalho pai.

Para obter informações sobre como enviar um AWS Batch trabalho com uma configuração de tempo limite, consulte. [Tutorial: enviar um trabalho](#)

Trabalhos do Amazon EKS

Um emprego é a menor unidade de trabalho em AWS Batch. Um AWS Batch trabalho no Amazon EKS tem um one-to-one mapeamento para um Kubernetes pod. Uma definição de AWS Batch trabalho é um modelo para um AWS Batch trabalho. Ao enviar um AWS Batch trabalho, você faz referência a uma definição de trabalho, direciona uma fila de trabalhos e fornece um nome para um trabalho. Na definição de trabalho de um AWS Batch trabalho no Amazon EKS, o parâmetro [EKSProperties](#) define o conjunto de parâmetros que um trabalho no AWS Batch Amazon EKS suporta. Em uma [SubmitJob](#) solicitação, o [eksPropertiesOverride](#) parâmetro permite a substituição de alguns parâmetros comuns. Dessa forma, você pode usar modelos de definições de trabalhos para vários trabalhos. Quando um trabalho é enviado para seu cluster Amazon EKS, AWS Batch transforma o trabalho em um podspec (). Kind: Pod O podspec usa alguns AWS Batch parâmetros adicionais para garantir que os trabalhos sejam escalados e programados corretamente. AWS Batch combina rótulos e contaminações para garantir que os trabalhos sejam executados somente em nós AWS Batch gerenciados e que outros pods não sejam executados nesses nós.

 Important

- Se o `hostNetwork` parâmetro não for definido explicitamente em uma definição de trabalho do Amazon EKS, o modo de rede do pod assumirá como AWS Batch padrão o modo host. Mais especificamente, as seguintes configurações serão aplicadas: `hostNetwork=true` e `dnsPolicy=ClusterFirstWithHostNet`.
- AWS Batch limpa os pods de trabalho logo após um pod concluir seu trabalho. Para ver os logs do aplicativo do pod, configure um serviço de log para seu cluster. Para obter mais

informações, consulte [Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS](#).

Tópicos

- [Tutorial: mapear um trabalho em execução para um pod e um nó](#)
- [Tutorial: mapear um pod em execução de volta ao seu trabalho](#)

Tutorial: mapear um trabalho em execução para um pod e um nó

As `podProperties` de um trabalho em execução têm parâmetros `podName` e `nodeName` definidos para a tentativa de trabalho atual. Use a operação [DescribeJobs](#) da API para visualizar esses parâmetros.

O seguinte é um exemplo de saída.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
  "jobs": [
    {
      "status": "RUNNING",
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/MyJobOnEks_SleepWithRequestsOnly:1",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
      "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
      "eksProperties": {
        "podProperties": {
          "nodeName": "ip-192-168-55-175.ec2.internal",
          "containers": [
            {
              "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
              "resources": {
                "requests": {
                  "cpu": "1",
                  "memory": "1024Mi"
                }
              }
            }
          ]
        }
      }
    }
  ]
}
```

```

    ],
    "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
  }
}
]
}

```

Para um trabalho com novas tentativas ativadas, o podName final nodeName de cada tentativa concluída está no parâmetro de eksAttempts lista da operação da [DescribeJobs](#) API. O podName e nodeName da tentativa de execução atual estarão no objeto podProperties.

Tutorial: mapear um pod em execução de volta ao seu trabalho

Um pod tem rótulos que indicam jobId a extremidade uuid do ambiente computacional ao qual ele pertence. AWS Batch injeta variáveis de ambiente para que o tempo de execução do trabalho possa referenciar as informações do trabalho. Para obter mais informações, consulte [AWS Batch variáveis do ambiente de trabalho](#). Você pode visualizar isso executando o seguinte comando. A saída é a seguinte:

```

$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-namespace
Name:          aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:     my-aws-batch-namespace
Priority:       0
Node:          ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:    Wed, 26 Oct 2022 00:30:48 +0000
Labels:        batch.amazonaws.com/compute-environment-uuid=5c19160b-d450-31c9-8454-86cf5b30548f
               batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
               batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:        Running
IP:            192.168.45.88
IPs:
  IP: 192.168.45.88
Containers:
  default:
    Image:          public.ecr.aws/amazonlinux/amazonlinux:2
    ...
    Environment:
      AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647

```



```
AWS_BATCH_JOB_ID:          f980f2cf-6309-4c77-a2b2-d83fbbba0e9f0
AWS_BATCH_JOB_NAME:        My-Eks-JQ1
AWS_BATCH_JOB_ATTEMPT:     1
AWS_BATCH_CE_NAME:        My-Eks-CE1
```

...

Recursos que o AWS Batch Amazon EKS Jobs oferece suporte

Esses são os recursos AWS Batch específicos que também são comuns aos Kubernetes trabalhos executados no Amazon EKS:

- [Dependências do trabalho](#)
- [Trabalhos de matriz](#)
- [Tempos limite de trabalho](#)
- [Repetições de trabalho automatizadas](#)
- [Usar o agendamento de compartilhamento justo para ajudar a agendar trabalhos](#)

Kubernetes **Secrets** e **ServiceAccounts**

AWS Batch suporta referências Kubernetes Secrets e ServiceAccounts. Você pode configurar pods para usar perfis do IAM para contas de serviço do Amazon EKS. Para obter mais informações, consulte [Como Configurar Pods para Utilizar uma Conta de Serviço Kubernetes](#) no [Manual do Usuário Amazon EKS](#).

Documentação relacionada

- [Considerações sobre memória e vCPU para AWS Batch no Amazon EKS](#)
- [Executar trabalhos de GPU](#)
- [Trabalhos presos no status **RUNNABLE**](#)

Trabalhos em paralelo de vários nós

Você pode usar trabalhos paralelos de vários nós para executar trabalhos únicos, que englobem várias instâncias do Amazon EC2. Com trabalhos paralelo de vários nós do AWS Batch (também conhecidos como programação de grupos), você pode executar aplicações de computação de alta performance em grande escala e treinamento em modelo de GPU distribuído sem a necessidade de

iniciar, configurar e gerenciar diretamente os recursos do Amazon EC2. Uma tarefa paralela de AWS Batch vários nós é compatível com qualquer estrutura que ofereça suporte à comunicação entre nós baseada em IP. Os exemplos incluem Apache MXNet TensorFlow, Caffe2 ou Message Passing Interface (MPI).

Trabalhos paralelos de vários nós de vários nós são enviados como uma única tarefa. No entanto, sua definição de trabalho (ou substituições de nó de envio de trabalho) especifica o número de nós a serem criados para o trabalho e quais grupos de nós criar. Cada trabalho paralelo de vários nós contém um nó principal que é executado primeiro. Depois que o nó principal estiver ativo, os nós filhos serão executados e iniciados. O trabalho será concluído somente se o nó principal sair. Nesse caso, todos nós secundários serão terminados. Para obter mais informações, consulte [Grupos de nós](#).

Os nós de trabalho paralelos de vários nós são de locatário único. Isso significa que um único contêiner de trabalho é executado em cada instância do Amazon EC2.

O status final do trabalho (SUCCEEDED ou FAILED) é determinado pelo status final do trabalho do nó principal. Para obter o status de um trabalho paralelo de vários nós, descreva a tarefa usando a ID da tarefa retornada quando você enviou a tarefa. Se precisar dos detalhes dos nós filhos, você deverá descrever cada nó filho individualmente. Você pode endereçar os nós usando a notação `#N` (começando com 0). Por exemplo, para acessar os detalhes do segundo nó de um trabalho, descreva `aws_batch_job_id#1` usando a operação da AWS Batch [DescribeJobs](#) API. As informações de `started`, `stoppedAt`, `statusReason` e `exit` para um trabalho paralelo de vários nós são preenchidas a partir do nó principal.

Se você especificar novas repetições de trabalho, uma falha no nó principal fará com que outra tentativa ocorra. Falhas no nó filho não causam a ocorrência de mais tentativas. Cada nova tentativa de um tarefa em paralelo de vários nós atualiza a tentativa correspondente de seus nós filhos associados.

Para executar trabalhos paralelos de vários nós AWS Batch, o código do aplicativo deve conter as estruturas e bibliotecas necessárias para a comunicação distribuída.

Tópicos

- [Variáveis de ambiente](#)
- [Grupos de nós](#)
- [Ciclo de vida do trabalho para trabalhos MNP](#)
- [Considerações sobre o ambiente computacional para MNP com AWS Batch](#)

Variáveis de ambiente

Em tempo de execução, cada nó é configurado com as variáveis de ambiente padrão que todos os AWS Batch trabalhos recebem. Além disso, os nós são configurados com as seguintes variáveis de ambiente específicas de trabalhos paralelos de vários nós:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Essa variável é definida como o número índice do nó principal do trabalho. O código do seu aplicativo pode comparar o `AWS_BATCH_JOB_MAIN_NODE_INDEX` ao `AWS_BATCH_JOB_NODE_INDEX` em um nó individual para determinar se ele é o nó principal.

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Essa variável é definida apenas em nós filhos de trabalhos em paralelo de vários nós. Essa variável não está presente no nó principal. Essa variável é definida como o IPv4 endereço privado do nó principal do trabalho. O código da aplicação do nó filho pode utilizar esse endereço para comunicar-se com o nó principal.

`AWS_BATCH_JOB_NODE_INDEX`

Essa variável é definida como o número do índice de nós do nó. O índice do nó começa em 0, e cada nó filho recebe um número índice exclusivo. Por exemplo, uma tarefa em paralelo de vários nós com 10 filhos possui valores índice de 0-9.

`AWS_BATCH_JOB_NUM_NODES`

Essa variável é definida como o número de nós solicitados por você para seu trabalho paralelo de vários nós.

Grupos de nós

Um grupo de nós é um grupo idêntico de nós de trabalhos que compartilham as mesmas propriedades do contêiner. Você pode usar AWS Batch para especificar até cinco grupos de nós distintos para cada trabalho.

Cada grupo pode ter suas próprias imagens de contêiner, comandos, variáveis de ambiente e assim por diante. Por exemplo, você pode enviar um trabalho que exija uma instância única `c5.xlarge` para o nó principal e cinco nós filhos da instância `c5.xlarge`. Cada um desses grupos de nós distintos pode especificar diferentes imagens de contêiner ou comandos a serem executados para cada trabalho.

Como alternativa, todos os nós em seu trabalho podem usar um único grupo de nós. Além disso, o código do aplicação pode diferenciar as perfis do nó, como o nó principal e o nó filho. Ele faz isso comparando a variável de ambiente `AWS_BATCH_JOB_MAIN_NODE_INDEX` com seu próprio valor para `AWS_BATCH_JOB_NODE_INDEX`. Você pode ter até 1.000 nós em um único trabalho. Esse é o limite padrão para instâncias em um cluster do Amazon ECS. Você pode [solicitar que esse limite seja aumentado](#).

 Note

Atualmente, todos os grupos de nós em um trabalho paralelo de vários nós devem usar o mesmo tipo de instância.

Ciclo de vida do trabalho para trabalhos MNP

Quando você envia um trabalho paralelo de vários nós, o trabalho entra no status `SUBMITTED`. Em seguida, o trabalho aguarda a conclusão de todas as dependências do trabalho. O trabalho também passa para o status `RUNNABLE`. Por fim, AWS Batch provisiona a capacidade da instância necessária para executar seu trabalho e executa essas instâncias.

Cada tarefa em paralelo de vários nós contém um nó principal. O nó principal é uma subtarefa única que AWS Batch monitora para determinar o resultado do trabalho de vários nós enviado. O nó principal é iniciado primeiro e passa para o status de `STARTING`. O valor do tempo limite especificado no campo de parâmetro `attemptDurationSeconds` se aplica a todo o trabalho, não aos nós.

Quando o nó principal atinge o status de `RUNNING` depois que o contêiner do nó estiver em execução, os nós filhos são iniciados e também são movidos para o status de `STARTING`. Os nós filhos são exibidos em uma ordem aleatória. Não há garantias sobre o tempo ou a ordem de execução do nó filho. Para garantir que todos os nós das tarefas estejam no status de `RUNNING` após a execução do contêiner do nó, o código do seu aplicativo pode consultar a API do AWS Batch para obter informações do nó principal e do nó filho. Como alternativa, o código da aplicação pode esperar até que todos os nós estejam online, antes de iniciar qualquer tarefa de processamento distribuído. O endereço IP privado do nó principal está disponível como a variável de ambiente `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` em cada nó filho. Seu código de aplicação pode usar essas informações para coordenar e comunicar dados entre cada tarefa.

À medida que os nós individuais saem, eles são movidos para `SUCCEEDED` ou `FAILED`, a depender do código de saída. Se o nó principal for encerrado, a tarefa será considerada terminada e todos

os nós filhos serão parados. Se um nó filho morrer, AWS Batch não executará nenhuma ação nos outros nós do trabalho. Se você não quiser que seu trabalho continue com um número de nós reduzido, você deve incluir isso no código do aplicativo. Isso termina ou cancela o trabalho.

Considerações sobre o ambiente computacional para MNP com AWS Batch

Há várias coisas a serem consideradas ao criar ambientes de computação para executar trabalhos paralelos de vários nós com o AWS Batch.

- Trabalhos paralelos de vários nós não são compatíveis com ambientes de computação UNMANAGED.
- Se você deseja enviar trabalhos paralelos de vários nós para um ambiente de computação, crie um grupo com posicionamento em cluster em uma Zona de Disponibilidade única e associe-a aos seus recursos de computação. Isso mantém seus trabalhos paralelos de vários nós em um agrupamento lógico de instâncias próximas com alto potencial de fluxo de rede. Para obter mais informações, consulte [Grupos de posicionamento](#), no Guia do usuário do Amazon EC2.
- Trabalhos paralelos de vários nós não são compatíveis com ambientes de computação que usam instâncias spot.
- AWS Batch trabalhos paralelos de vários nós usam o modo de awsvpc rede do Amazon ECS, que fornece aos contêineres de trabalhos paralelos de vários nós as mesmas propriedades de rede das instâncias do Amazon EC2. Cada contêiner de trabalho paralelo de vários nós obtém sua própria interface de rede elástica, um endereço IP privado primário e um nome de host DNS interno. A interface de rede é criada na mesma sub-rede VPC que seu recurso de computação do host.
- Seu ambiente de computação não pode ter mais de cinco grupos de segurança associados a ele. As interfaces de rede elástica criadas e anexadas a uma tarefa do MNP usarão os grupos de segurança especificados em seu ambiente computacional. Se você não especificar um grupo de segurança, o grupo de segurança padrão da VPC será usado.
- O modo de rede awsvpc não fornece as interfaces de rede elásticas para trabalhos paralelos de vários nós com endereços IP públicos. Para acessar a internet, seus recursos de computação devem ser iniciados em uma sub-rede privada configurada para usar um gateway NAT. Para obter mais informações, consulte [Gateways NAT](#) no Guia do usuário da Amazon VPC. A comunicação entre nós deve usar o endereço IP privado ou o nome do host DNS para o nó. Trabalhos paralelos de vários nós executados em recursos de computação em sub-redes públicas não têm acesso à rede de saída. Para criar uma VPC com sub-redes privadas e um gateway NAT, consulte [Criar uma nuvem privada virtual](#).

- As interfaces de rede elástica criadas e anexadas aos seus recursos de computação não podem ser desassociadas manualmente nem modificadas pela sua conta. Isso visa prevenir a exclusão acidental de uma interface de rede elástica associada a uma tarefa em execução. Para liberar as interfaces de rede elásticas para uma tarefa, encerre a tarefa.
- Seu ambiente computacional deve ter o máximo suficiente CPUs para suportar seu trabalho paralelo de vários nós.
- Sua cota de instância do Amazon EC2 inclui o número de instâncias exigidas para executar seu trabalho. Por exemplo, suponha que sua tarefa exija 30 instâncias, mas sua conta só pode executar 20 instâncias em uma região. Nesse caso, seu trabalho ficará preso no status `RUNNABLE`.
- Se você especificar um tipo de instância para um grupo de nós em um trabalhos paralelos de vários nós, seu ambiente de computação deverá executar esse tipo de instância.

Trabalhos paralelos de vários nós no Amazon EKS

Você pode usar o AWS Batch Amazon Elastic Kubernetes Service para executar trabalhos paralelos de vários nós (MNP) (também conhecidos como agendamento de grupos) em seus clusters gerenciados. Kubernetes Essa opção é comumente usada para trabalhos grandes, de acoplamento firme e de alto desempenho que não podem ser executados em uma única instância do Amazon Elastic Compute Cloud. Para obter mais informações, consulte [Trabalhos em paralelo de vários nós](#).

É possível usar esse atributo para executar aplicações de computação de alto desempenho específicos do Kubernetes gerenciado pelo Amazon EKS, treinamento de grandes modelos de linguagem e outros trabalhos de Inteligência Artificial (IA)/machine learning (ML).

Tópicos

- [Execução de trabalhos de MNP](#)
- [Criar uma definição de trabalho de GPU do Amazon EKS](#)
- [Enviar um trabalho MNP do Amazon EKS](#)
- [Substituir uma definição de trabalho MNP do Amazon EKS](#)

Execução de trabalhos de MNP

AWS Batch suporta trabalhos de MNP no Amazon Elastic Container Service e no Amazon EKS usando o Amazon EC2. Veja a seguir mais detalhes sobre os parâmetros de instância e contêiner do atributo.

Cotas de instância para MNP no Amazon EKS

- Até 1.000 instâncias podem ser usadas para um único trabalho MNP.
- Até 5.000 instâncias podem participar de um único cluster do Amazon EKS.
- Até 5 ambientes de computação podem ser agrupados e anexados a uma fila de trabalhos.

Por exemplo, é possível dimensionar até 5 ambientes de computação em cluster em uma fila de trabalhos e 1.000 instâncias em cada ambiente de computação.

Além dos parâmetros de instância, é importante observar que você não pode usar o Fargate para trabalhos MNP por meio de nenhum dos serviços.

É possível usar apenas um tipo de instância em cada trabalho MNP. É possível alterar o tipo de instância ao atualizar o ambiente de computação ou ao definir um novo ambiente de computação. Também é possível especificar o tipo de instância e fornecer requisitos de vCPU e memória ao criar a definição de trabalho.

Cotas de contêineres para MNP no Amazon EKS

- Um trabalho paralelo de vários nós aceita um pod por nó.
- Até 10 contêineres (ou 10 contêineres de inicialização. Para obter mais informações, consulte [Contêineres de inicialização](#) na documentação do Kubernetes) em cada pod.
- Até 5 intervalos de nós em cada trabalho MNP.
- Até 10 imagens de contêineres distintos em cada intervalo de nós.

Por exemplo, é possível executar até máximo de 10.000 contêineres em um único trabalho MNP com 5 intervalos de nós e um total de 50 imagens exclusivas.

Execução de trabalhos MNP em uma Amazon VPC privado e em um cluster do Amazon EKS

Os trabalhos MNP podem ser executados em qualquer cluster do Amazon EKS, quer ele tenha Internet pública ou não. Ao usar um cluster do Amazon EKS com apenas acesso à rede privada, certifique-se de que ele AWS Batch possa acessar o plano de controle do Amazon EKS e o servidor de Kubernetes API gerenciado. É possível conceder o acesso necessário por meio dos endpoints da Amazon Virtual Private Cloud. Para obter mais informações, consulte [Configurar um serviço de endpoint](#).

Os pods de cluster do Amazon EKS não podem baixar uma imagem de uma fonte pública, pois a VPC privada não tem acesso à Internet. O cluster do Amazon EKS deve extrair imagens de um registro de contêiner que esteja na Amazon VPC. É possível criar um [Amazon Elastic Container Registry \(Amazon ECR\)](#) na sua Amazon VPC e copiar imagens de contêiner para ele para acesso aos seus nós.

Também é possível criar uma regra de cache pull through com o Amazon ECR. Depois que uma regra de cache pull through é criada para um registro público externo, você pode simplesmente extrair uma imagem desse registro público externo usando o URI do registro privado do Amazon ECR. Em seguida, o Amazon ECR cria um repositório e armazena a imagem no cache. Quando uma imagem em cache é extraída usando o URI de registro privado do Amazon ECR, o Amazon ECR verifica o registro remoto para ver se há uma nova versão da imagem e atualizará seu registro privado até uma vez a cada 24 horas. Para obter mais informações, consulte [Criar uma regra de cache de pullthrough no Amazon ECR](#).

Notificação de erro

Se suas tarefas de MNP estiverem bloqueadas, você poderá receber notificações por meio do Console de gerenciamento da AWS e da Amazon EventBridge. Por exemplo, se um trabalho MNP estiver preso no topo da fila, você poderá receber uma notificação sobre o problema, juntamente com informações sobre o que o causou, para que você possa tomar medidas imediatas para desbloquear a fila de trabalhos. Opcionalmente, é possível encerrar automaticamente o trabalho MNP se nenhuma ação for tomada em um determinado período de tempo, que pode ser definido no modelo de fila de trabalho. Para obter mais informações, consulte [Eventos bloqueados da fila de trabalhos](#).

Criar uma definição de trabalho de GPU do Amazon EKS

Para definir e executar trabalhos MNP no Amazon EKS, há novos parâmetros nas operações de API [RegisterJobDefinition](#) e [SubmitJob](#).

- Use [eksProperties](#) na seção [nodeProperties](#) para configurar sua definição de trabalho MNP.
- Use [eksPropertiesOverride](#) na seção [nodePropertyOverrides](#) para substituir os parâmetros configurados na definição do trabalho ao enviar um trabalho MNP.

Essas ações podem ser definidas por meio de operações de API e do Console de gerenciamento da AWS.

Referência: registrar a carga útil da solicitação de definição de trabalho MNP do Amazon EKS

O exemplo a seguir ilustra como é possível registrar uma definição de trabalho MNP do Amazon EKS com dois nós.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext": {
                  "runAsUser": 1000,
                  "runAsGroup": 3000,
                  "privileged": true,
                  "readOnlyRootFilesystem": true,
                  "runAsNonRoot": true
                }
              }
            ]
          },
          "initContainers": [
            {
              "name": "init-ekscontainer",
```

```
"image": "public.ecr.aws/amazonlinux/amazonlinux:2",  
  "command": [  
    "echo",  
    "helloWorld"  
  ],  
  "resources": {  
    "limits": {  
      "cpu": "1",  
      "memory": "1024Mi"  
    }  
  }  
},  
],  
"metadata": {  
  "labels": {  
    "environment" : "test"  
  }  
}  
}  
}  
]  
}
```

Para registrar a definição de tarefa usando o AWS CLI, copie a definição em um arquivo local chamado `MyEksMnpJobDefinition.json` e execute o comando a seguir.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Você receberá a seguinte resposta JSON.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/MyEksMnpJobDefinition:1",
  "revision": 1
}
```

Enviar um trabalho MNP do Amazon EKS

Para enviar um trabalho usando a definição de trabalho registrada, digite o seguinte comando. Substitua o valor de `<EKS_JOB_QUEUE_NAME>` pelo nome ou ARN de uma fila de trabalho pré-existente associada a um ambiente de computação do Amazon EKS.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \  
  --job-definition MyEksMnpJobDefinition \  
  --job-name myFirstEksMnpJob
```

Você receberá a seguinte resposta JSON.

```
{  
  "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",  
  "jobName": "myFirstEksMnpJob",  
  "jobId": "<JOB_ID>"  
}
```

Você pode verificar o status do trabalho usando o jobId retornado com o seguinte comando.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Substituir uma definição de trabalho MNP do Amazon EKS

Opcionalmente, é possível substituir os detalhes da definição do trabalho (como alterar o tamanho do trabalho MNP ou os detalhes do trabalho secundário). A seguir, um exemplo de carga útil de solicitação JSON para enviar um trabalho MNP de cinco nós e alterações no comando `test-eks-container-1` do contêiner.

```
{  
  "numNodes": 5,  
  "nodePropertyOverrides": [  
    {  
      "targetNodes": "0:",  
      "eksPropertiesOverride": {  
        "podProperties": {  
          "containers": [  
            {  
              "name": "test-eks-container-1",  
              "command": [  

```

```
        "sleep",  
        "150"  
    ]  
  }  
]  
}  
}  
]  
}  
]  
}
```

Para enviar um trabalho com essas substituições, salve o exemplo em um arquivo local, `eks-mnp-job-nodeoverride.json`, e use o AWS CLI para enviar o trabalho com as substituições.

Trabalhos de matriz

Um trabalho de matriz é um trabalho que compartilha parâmetros comuns, como a definição do trabalho CPUs, v e memória. Ele é executado como coleções de tópicos de trabalhos básicos porém separados, que podem ser distribuídos entre vários hosts e executados simultaneamente. Os trabalhos de matriz são a maneira mais eficiente de executar trabalhos extremamente em paralelo, como simulações de Monte Carlo, varreduras paramétricas ou trabalhos grandes de renderização.

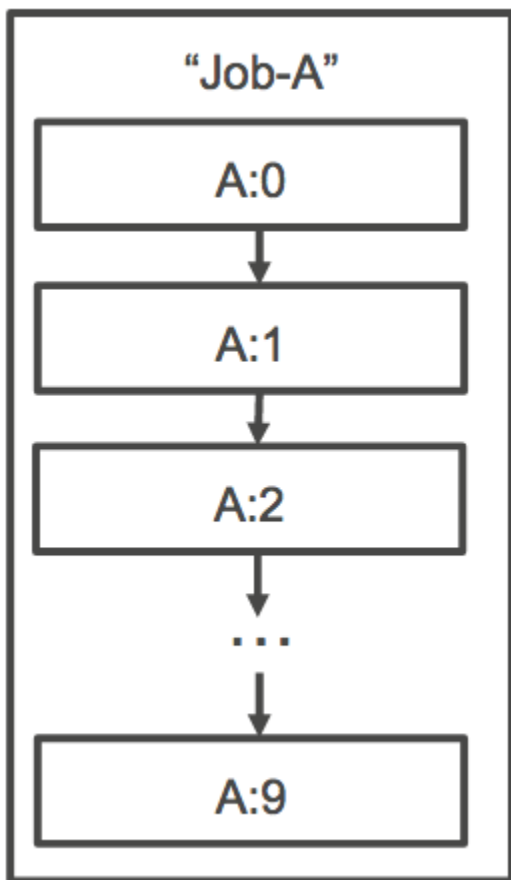
AWS Batch os trabalhos de matriz são enviados da mesma forma que os trabalhos normais. No entanto, você especifica o tamanho da matriz (entre 2 e 10.000) para definir quantos trabalhos filho devem ser executados no campo da matriz. Se você enviar um trabalho com tamanho da matriz 1.000, haverá uma única execução de trabalho gerando 1.000 trabalhos filho. O trabalho de matriz é uma referência ou ponteiro para gerenciar todos os trabalhos filho. Você pode enviar grandes workloads com uma única consulta. O tempo limite especificado no campo do parâmetro `attemptDurationSeconds` é aplicado em cada trabalho filho. O trabalho pai da matriz não tem um tempo limite.

Quando você envia um trabalho de matriz, o trabalho de matriz principal recebe uma ID de AWS Batch trabalho normal. Cada trabalho filho tem a mesma ID base. No entanto, o índice da matriz do trabalho filho é anexado até o fim da ID pai, como `example_job_ID:0` para o primeiro trabalho filho da matriz.

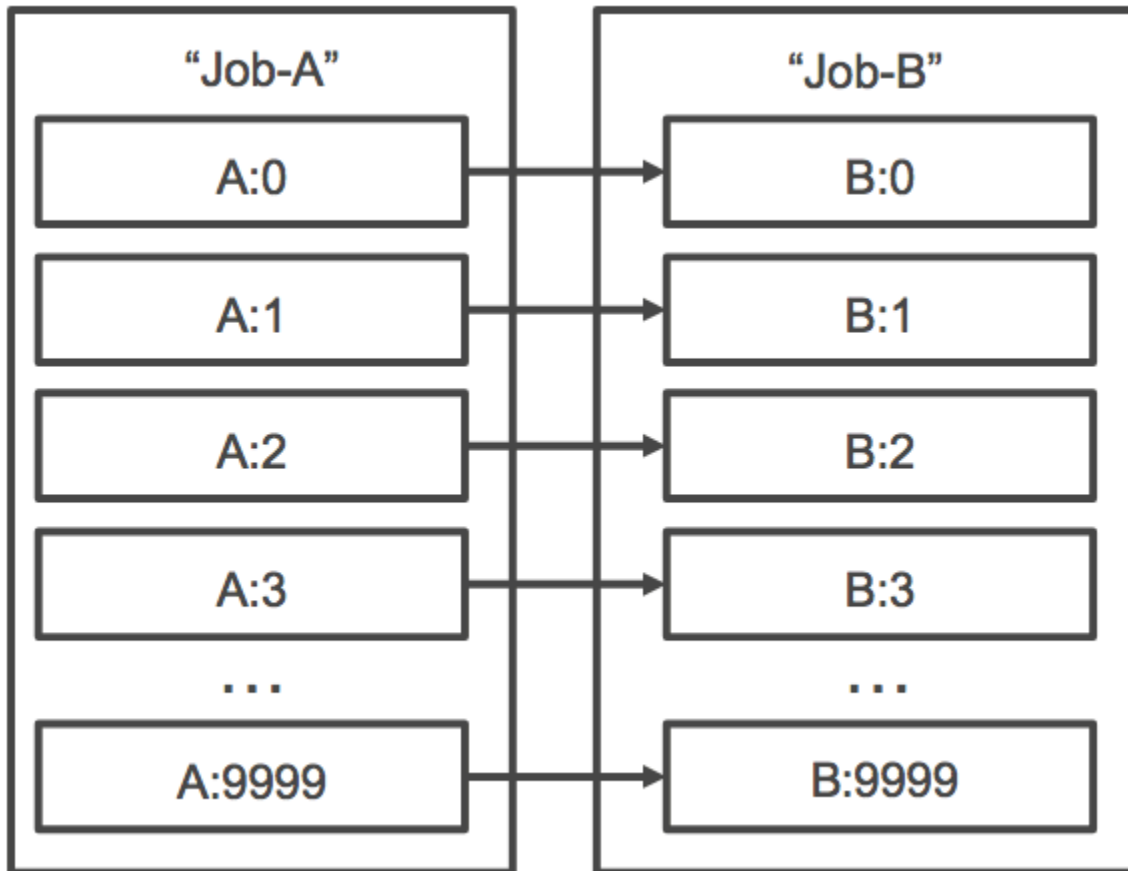
O trabalho de matriz pai pode inserir um status SUBMITTED, PENDING, FAILED ou SUCCEEDED. Um trabalho de matriz pai é atualizado para PENDING quando qualquer trabalho filho é atualizado para RUNNABLE. Para obter mais informações sobre dependências do trabalho, consulte [Dependências do trabalho](#).

Durante o runtime, a variável de ambiente `AWS_BATCH_JOB_ARRAY_INDEX` é definida como o número do índice da matriz do trabalho correspondente do contêiner. O primeiro índice do trabalho de matriz é numerado com 0 e as tentativas subsequentes são numeradas em ordem crescente (por exemplo 1, 2 e 3). Você pode utilizar esse valor de índice para controlar como os filhos do trabalho de matriz serão diferenciados. Para obter mais informações, consulte [Usar o índice de trabalho de matriz para controle de diferenciação de trabalhos](#).

Para dependências do trabalhos de matriz, você pode especificar um tipo para cada dependência, como `SEQUENTIAL` ou `N_TO_N`. Você pode especificar uma dependência do tipo `SEQUENTIAL` (sem especificar ID de trabalho), de forma que cada trabalho de matriz filho seja concluído sequencialmente, começando com o índice 0. Por exemplo, se você enviar um trabalho de matriz com o tamanho da matriz 100 e especificar uma dependência do tipo `SEQUENTIAL`, 100 trabalhos filho serão gerados sequencialmente. O primeiro trabalho filho precisará ter êxito antes que o próximo trabalho filho comece. A figura abaixo mostra o trabalho A, um trabalho de matriz com um tamanho da matriz somando 10. Cada trabalho no índice filho do trabalho A depende do trabalho filho anterior. O trabalho A:1 não pode iniciar até que o trabalho A:0 termine.



Você também pode especificar uma dependência do tipo N_TO_N com ID do trabalho para trabalhos de matriz. Deste modo, cada índice filho dessa tarefa precisa aguardar, para que o índice filho correspondente de cada dependência seja concluído antes de começar. A figura a seguir mostra o trabalho A e o trabalho B, dois trabalhos de matriz de tamanho 10.000 cada. Cada trabalho no índice filho do trabalho B depende do índice correspondente no trabalho A. O trabalho B:1 não pode iniciar até que o trabalho A:1 termine.



Se você cancelar ou encerrar um trabalho de matriz pai, todos os trabalhos filho serão cancelados ou encerrados com ele. Você pode cancelar ou encerrar trabalhos filho individuais (movendo-os para um status FAILED) sem afetar os outros trabalhos filho. No entanto, se um trabalho de matriz filho falhar (sozinho, por ter sido cancelado ou por encerramento manual), o trabalho pai também falhará. Nesse cenário, o trabalho pai faz a transição para FAILED quando todos os trabalhos filhos são concluídos.

Para obter mais informações sobre como pesquisar e filtrar trabalhos de matriz, consulte [Pesquisar por trabalhos em uma fila de trabalho](#).

Tópicos

- [Exemplo de um fluxo de trabalho de matriz](#)
- [Usar o índice de trabalho de matriz para controle de diferenciação de trabalhos](#)

Exemplo de um fluxo de trabalho de matriz

Um fluxo de trabalho comum para AWS Batch os clientes é executar um trabalho de configuração pré-requisito, executar uma série de comandos em um grande número de tarefas de entrada e, em seguida, concluir com um trabalho que agrega resultados e grava dados resumidos no Amazon S3, DynamoDB, Amazon Redshift ou Aurora.

Por exemplo:

- JobA: um trabalho de não matriz padrão que executa uma rápida listagem e validação de metadados de objetos em um bucket do Amazon S3, BucketA. A sintaxe [SubmitJob](#)JSON é a seguinte.

```
{
  "jobName": "JobA",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobA-list-and-validate:1"
}
```

- JobB: um trabalho de matriz com 10 mil cópias que depende do JobA, que, por sua vez, executa comandos com uso intensivo de CPU para cada objeto em BucketA e carrega resultados para BucketB. A sintaxe [SubmitJob](#)JSON é a seguinte.

```
{
  "jobName": "JobB",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobB-CPU-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "4096"
      },
      {
        "type": "VCPU",
        "value": "32"
      }
    ]
  }
}
```

```

    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobA_job_ID"
    }
  ]
}

```

- JobC: outro trabalho de matriz com 10 mil cópias, que depende do JobB com um modelo de dependência de N_T0_N, que executa comandos com uso intensivo de memória em relação a cada item em BucketB, grava metadados no DynamoDB e carrega saída resultante em BucketC. A sintaxe [SubmitJobJSON](#) é a seguinte.

```

{
  "jobName": "JobC",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobC-Memory-Intensive-Processing:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10000
  },
  "dependsOn": [
    {
      "jobId": "JobB_job_ID",
      "type": "N_T0_N"
    }
  ]
}

```



```
}
```

- JobD: um trabalho de matriz que executa 10 etapas de validação, no qual cada uma precisa consultar o DynamoDB e pode interagir com qualquer um dos buckets do Amazon S3 mencionados acima. Cada uma das etapas no JobD executa o mesmo comando. No entanto, o comportamento é diferente baseado no valor da variável de ambiente `AWS_BATCH_JOB_ARRAY_INDEX` dentro do contêiner do trabalho. Essas etapas de validação são executadas sequencialmente (por exemplo, JobD:0 e, em seguida, JobD:1). A sintaxe [SubmitJob](#)JSON é a seguinte.

```
{
  "jobName": "JobD",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobD-Sequential-Validation:1",
  "containerOverrides": {
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "32768"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
  "arrayProperties": {
    "size": 10
  },
  "dependsOn": [
    {
      "jobId": "JobC_job_ID"
    },
    {
      "type": "SEQUENTIAL"
    }
  ]
}
```

- JobE: um trabalho de não matriz final, que executa algumas operações de limpeza simples e envia uma notificação SNS da Amazon, com uma mensagem avisando que o pipeline foi concluído e vinculado a URL de saída. A sintaxe [SubmitJobJSON](#) é a seguinte.

```
{
  "jobName": "JobE",
  "jobQueue": "ProdQueue",
  "jobDefinition": "JobE-Cleanup-and-Notification:1",
  "parameters": {
    "SourceBucket": "s3://amzn-s3-demo-source-bucket",
    "Recipient": "pipeline-notifications@mycompany.com"
  },
  "dependsOn": [
    {
      "jobId": "JobD_job_ID"
    }
  ]
}
```

Usar o índice de trabalho de matriz para controle de diferenciação de trabalhos

Este tutorial descreve instruções de uso da variável de ambiente `AWS_BATCH_JOB_ARRAY_INDEX` para diferenciar os trabalhos filho. Cada trabalho filho é atribuído a essa variável. O exemplo utiliza o número de índice do trabalho filho para ler uma linha específica em um arquivo. Em seguida, ele substitui o parâmetro associado a esse número de linha por um comando dentro do contêiner do trabalho. O resultado é que você pode ter vários AWS Batch trabalhos que executam a mesma imagem do Docker e argumentos de comando. No entanto, os resultados serão diferentes porque o índice de trabalho da matriz é usado como um modificador.

Neste tutorial, um arquivo de texto é criado com todas as cores do arco-íris, cada uma em sua própria linha. Em seguida, você cria um script de ponto de entrada para um contêiner do Docker que converte o índice em um valor, que pode ser usado para um número de linha no campo do arquivo de cores. O índice inicia em zero, mas os números da linha iniciam em um. Crie um Dockerfile que copie os arquivos de cores e índice na imagem de contêiner, e defina o `ENTRYPOINT` da imagem como o script de ponto de entrada. O Dockerfile e os recursos são incorporados em uma imagem do Docker, que é enviada para o Amazon ECR. Em seguida, você registra uma definição de tarefa que

usa sua nova imagem de contêiner, envia uma tarefa de AWS Batch matriz com essa definição de tarefa e visualiza os resultados.

Tópicos

- [Pré-requisitos](#)
- [Criar uma imagem de contêiner](#)
- [Enviar sua imagem para o Amazon ECR](#)
- [Criar e registrar uma definição de trabalho](#)
- [Enviar um trabalho AWS Batch de matriz](#)
- [Visualizar seus logs de trabalhos de matriz](#)

Pré-requisitos

Este fluxo de trabalho de tutorial tem os seguintes pré-requisitos:

- Um ambiente AWS Batch computacional. Para obter mais informações, consulte [Crie um ambiente de computação](#).
- Uma fila AWS Batch de trabalhos e um ambiente computacional associado. Para obter mais informações, consulte [Crie uma fila de trabalhos](#).
- O AWS CLI instalado em seu sistema local. Para obter mais informações, consulte [> Instalar a AWS Command Line Interface](#) no Guia do usuário da AWS Command Line Interface .
- Docker instalado em seu sistema local. Para obter mais informações, consulte [Sobre o Docker CE](#) no campo de documentação do Docker.

Criar uma imagem de contêiner

Você pode usar o `AWS_BATCH_JOB_ARRAY_INDEX` em uma definição de trabalho no parâmetro de comando. No entanto, recomendamos criar uma imagem de contêiner que use a variável em um script de ponto de entrada. Esta seção descreve instruções para criação dessa imagem de contêiner.

Para criar sua imagem de contêiner do Docker

1. Crie novo diretório, a ser usado como seu workspace de imagem do Docker, e navegue até ele.
2. Crie um arquivo de nome `colors.txt` em seu diretório do workspace e cole o conteúdo a seguir.

```
red
orange
yellow
green
blue
indigo
violet
```

3. Crie um arquivo de nome `print-color.sh` em seu diretório do workspace e cole o conteúdo a seguir.

 Note

A variável `LINE` é definida como `AWS_BATCH_JOB_ARRAY_INDEX + 1`, pois o índice de matriz inicia em 0, mas os números de linha iniciam em com 1. A variável `COLOR` é definida como a cor no `colors.txt` associado ao número de linha.

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Crie um arquivo de nome `Dockerfile` em seu diretório do workspace e cole nele o conteúdo a seguir. Esse `Dockerfile` copia os arquivos anteriores em seu contêiner e define o script de ponto de entrada a ser executado quando o contêiner for iniciado.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Compile sua imagem do Docker.

```
$ docker build -t print-color .
```

6. Teste seu contêiner com o script a seguir. Esse script define a variável `AWS_BATCH_JOB_ARRAY_INDEX` como 0 localmente e, em seguida, a acrescenta, para simular o que um trabalho de matriz com sete filhos faz.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

A saída vem a seguir.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Enviar sua imagem para o Amazon ECR

Agora que você criou e testou o contêiner do Docker, envie-o para um repositório de imagens. Este exemplo usa o Amazon ECR, mas você pode usar outro registro, como DockerHub.

1. Crie um repositório de imagens do Amazon ECR para armazenar sua imagem de contêiner. Este exemplo usa somente AWS CLI o. mas você também pode usar Console de gerenciamento da AWS o. Para obter mais informações, consulte [Criando um Repositório](#) no campo do Manual do Usuário do Amazon Elastic Container Registry.

```
$ aws ecr create-repository --repository-name print-color
```

2. Marque sua imagem `print-color` com o URI do repositório do Amazon ECR retornado na etapa anterior.

```
$ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

3. Inicie a sessão no seu registro do Amazon ECR. Para obter mais informações, consulte [Autenticação de Registro](#) no campo do Manual do Usuário do Amazon Elastic Container Registry.

```
$ aws ecr get-login-password \
  --region region | docker login \
  --username AWS \
  --password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

4. Envie sua imagem para o Amazon ECR.

```
$ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
```

Criar e registrar uma definição de trabalho

Agora que sua imagem do Docker está em um registro de imagem, você pode especificá-la em uma definição de AWS Batch tarefa. Assim, você pode usá-la posteriormente para executar um trabalho de matriz. Este exemplo usa apenas o AWS CLI. No entanto, você também pode usar o Console de gerenciamento da AWS. Para obter mais informações, consulte [Criar uma definição de tarefa de nó único](#).

Para criar uma definição de trabalho

1. Crie um arquivo de nome `print-color-job-def.json` em seu diretório do workspace e cole o conteúdo a seguir. Substitua o URI do repositório de imagem pelo seu próprio URI da imagem.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
      {
        "type": "MEMORY",
        "value": "250"
      },
      {
        "type": "VCPU",
        "value": "1"
      }
    ]
  }
}
```

```
]
}
}
```

2. Registre a definição do trabalho com AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-def.json
```

Enviar um trabalho AWS Batch de matriz

Depois de registrar sua definição de tarefa, você pode enviar uma tarefa de AWS Batch matriz que usa sua nova imagem de contêiner.

Para enviar um trabalho AWS Batch de matriz

1. Crie um arquivo de nome `print-color-job.json` em seu diretório do workspace e cole o conteúdo a seguir.

Note

Este exemplo usa a fila de trabalhos mencionada na seção [the section called “Pré-requisitos”](#).

```
{
  "jobName": "print-color",
  "jobQueue": "existing-job-queue",
  "arrayProperties": {
    "size": 7
  },
  "jobDefinition": "print-color"
}
```

2. Envie o trabalho para sua fila de AWS Batch trabalhos. Anote a ID do trabalho retornado na saída.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Descreva o status do trabalho e espere até que o trabalho seja movido para SUCCEEDED.

Visualizar seus logs de trabalhos de matriz

Depois que seu trabalho atingir o SUCCEEDED status, você poderá visualizar os CloudWatch registros no contêiner do trabalho.

Para ver os registros do seu trabalho em CloudWatch Registros

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. No painel de navegação à esquerda, escolha Trabalhos.
3. Para Fila de Trabalhos, selecione uma fila.
4. Na seção Status, escolha bem-sucedido.
5. Para exibir todas os trabalhos filho de seu trabalho matriz, selecione a ID do trabalho que foi retornado na seção anterior.
6. Para consultar os logs do contêiner do trabalho, selecione um dos trabalhos filho e escolha Visualizar Logs.

Filter events	
Time (UTC +00:00)	Message
2018-07-13	
No older events found at the moment. Retry.	
▶ 20:16:20	My favorite color of the rainbow is red.
No newer events found at the moment. Retry.	

7. Visualize os outros logs de trabalho filho. Cada trabalho retorna uma cor diferente do arco-íris.

Executar trabalhos de GPU

Os trabalhos de GPU ajudam você a executar trabalhos que usam os de uma instância. GPUs

Os seguintes tipos de instância do Amazon EC2 baseados em GPU são compatíveis. Para obter mais informações, consulte [Instâncias Amazon EC2 G3](#), [Instâncias Amazon EC2 G4](#), [Instâncias Amazon EC2 G5](#), [Instâncias Amazon EC2 G6](#), [Instâncias Amazon EC2 P2](#), [Instâncias Amazon EC2 P3](#), [Instâncias Amazon EC2 P4d](#), [Instâncias Amazon EC2 P5](#), [Instâncias Amazon EC2 P6](#), [Instâncias Amazon EC2 Trn1](#), [Instâncias Amazon EC2 Trn2](#), [Instâncias Amazon EC2 Inf1](#), [Instâncias Amazon EC2 Inf2](#), [Instâncias Amazon EC2 DI1](#) e [Instâncias Amazon EC2 DI2](#).

Tipo de instância	GPUs	Memória da GPU	v CPUs	Memória	Largura de banda de rede
g3s.xlarge	1	8 GiB	4	30,5 GiB	10 Gbps
g3.4xlarge	1	8 GiB	16	122 GiB	Até 10 Gbps
g3.8xlarge	2	16 GiB	32	244 GiB	10 Gbps
g3.16xlarge	4	32 GiB	64	488 GiB	25 Gbps
g4dn.xlarge	1	16 GiB	4	16 GiB	Até 25 Gbps
g4dn.2xlarge	1	16 GiB	8	32 GiB	Até 25 Gbps
g4dn.4xlarge	1	16 GiB	16	64 GiB	Até 25 Gbps
g4dn.8xlarge	1	16 GiB	32	128 GiB	50 Gbps
g4dn.12xlarge	4	64 GiB	48	192 GiB	50 Gbps
g4dn.16xlarge	1	16 GiB	64	256 GiB	50 Gbps
g5.xlarge	1	24 GiB	4	16 GiB	Até 10 Gbps
g5.2xlarge	1	24 GiB	8	32 GiB	Até 10 Gbps
g5.4xlarge	1	24 GiB	16	64 GiB	Até 25 Gbps
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g5.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
g5g.xlarge	1	16 GiB	4	8 GiB	Até 10 Gbps
g5g.2xlarge	1	16 GiB	8	16 GiB	Até 10 Gbps

Tipo de instância	GPUs	Memória da GPU	v CPUs	Memória	Largura de banda de rede
g5g.4xlarge	1	16 GiB	16	32 GiB	Até 10 Gbps
g5g.8xlarge	1	16 GiB	32	64 GiB	12 Gbps
g5g.16xlarge	2	32 GiB	64	128 GiB	25 Gbps
g5g.metal	2	32 GiB	64	128 GiB	25 Gbps
g6.xlarge	1	24 GiB	4	16 GiB	Até 10 Gbps
g6.2xlarge	1	24 GiB	8	32 GiB	Até 10 Gbps
g6.4xlarge	1	24 GiB	16	64 GiB	Até 25 Gbps
g6.8xlarge	1	24 GiB	32	128 GiB	25 Gbps
g6.16xlarge	1	24 GiB	64	256 GiB	25 Gbps
g6.12xlarge	4	96 GiB	48	192 GiB	40 Gbps
g6.24xlarge	4	96 GiB	96	384 GiB	50 Gbps
g6.48xlarge	8	192 GiB	192	768 GiB	100 Gbps
g6e.xlarge	1	48 GiB	4	32 GiB	Até 20 Gbps
g6e.2xlarge	1	48 GiB	8	64 GiB	Até 20 Gbps
g6e.4xlarge	1	48 GiB	16	128 GiB	20 Gbps
g6e.8xlarge	1	48 GiB	32	256 GiB	25 Gbps
g6e.16xlarge	1	48 GiB	64	512 GiB	35 Gbps
g6e.12xlarge	4	192 GiB	48	384 GiB	100 Gbps
g6e.24xlarge	4	192 GiB	96	768 GiB	200 Gbps
g6e.48xlarge	8	384 GiB	192	1536 GiB	400 Gbps

Tipo de instância	GPUs	Memória da GPU	v CPUs	Memória	Largura de banda de rede
gr6.4xlarge	1	24 GiB	16	128 GiB	Até 25 Gbps
gr6.8xlarge	1	24 GiB	32	256 GiB	25 Gbps
p2.xlarge	1	12 GiB	4	61 GiB	Alto
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gbps
p2.16xlarge	16	192 GiB	64	732 GiB	20 Gbps
p3.2xlarge	1	16 GiB	8	61 GiB	Até 10 Gbps
p3.8xlarge	4	64 GiB	32	244 GiB	10 Gbps
p3.16xlarge	8	128 GiB	64	488 GiB	25 Gbps
p3dn.24xlarge	8	256 GiB	96	768 GiB	100 Gbps
p4d.24xlarge	8	320 GiB	96	1152 GiB	400 Gbps
p4de.24xlarge	8	640 GiB	96	1152 GiB	400 Gbps
p5.48xlarge	8	640 GiB	192	2 TiB	3200 Gbps
p5e.48xlarge	8	1128 GiB	192	2 TiB	3200 Gbps
p5en.48xlarge	8	1128 GiB	192	2 TiB	3200 Gbps
p6-b200.48xlarge	8	1440 GiB	192	2 TiB	100 Gbps
trn1.2xlarge	1	32 GiB	8	32 GiB	Até 12,5 Gbps
trn1.32xlarge	16	512 GiB	128	512 GiB	800 Gbps
trn1n.32xlarge	16	512 GiB	128	512 GiB	1600 Gbps
trn2.48xlarge	16	1,5 TiB	192	2 TiB	3,2 Tbps
inf1.xlarge	1	8 GiB	4	8 GiB	Até 25 Gbps

Tipo de instância	GPUs	Memória da GPU	v CPUs	Memória	Largura de banda de rede
inf1.2xlarge	1	8 GiB	8	16 GiB	Até 25 Gbps
inf1.6xlarge	4	32 GiB	24	48 GiB	25 Gbps
inf1.24xlarge	16	128 GiB	96	192 GiB	100 Gbps
inf2.xlarge	1	32 GiB	4	16 GiB	Até 15 Gbps
inf2.8xlarge	1	32 GiB	32	128 GiB	Até 25 Gbps
inf2.24xlarge	6	192 GiB	96	384 GiB	50 Gbps
inf2.48xlarge	12	384 GiB	192	768 GiB	100 Gbps
dl1.24xlarge	8	256 GiB	96	768 GiB	400 Gbps
dl2q.24xlarge	8	128 GiB	96	768 GiB	100 Gbps

Note

Para trabalhos de GPU, AWS Batch só oferece suporte a tipos de instância que tenham GPUs NVIDIA. Por exemplo, a família [G4ad](#) não é compatível com o agendamento de GPU. Você ainda pode usar [G4ad](#) on AWS Batch definindo somente os requisitos de vcpu e memória na definição do trabalho e, em seguida, acessando as GPUs hospedeiras diretamente por meio da personalização em um [modelo de lançamento do Amazon EC2, dados do usuário com](#) uma AMI otimizada para computação do Amazon ECS ou Amazon EKS, ou uma AMI personalizada para usar a AMD. GPUs

Os tipos de instância que usam uma ARM64 arquitetura são compatíveis com trabalhos de GPU em dados personalizados AMIs fornecidos ao usuário AWS Batch ou no Amazon EC2 para GPUs acessá-los por meio de código e configurações personalizados. Por exemplo, a família de instâncias [G5g](#).

O parâmetro [ResourceRequirements](#) para a definição do trabalho especifica o número de tarefas a GPUs serem fixadas no contêiner. Esse número de GPUs não está disponível para nenhum outro

trabalho executado nessa instância durante esse trabalho. Todos os tipos de instâncias em um ambiente computacional que executem trabalhos de GPU devem ser das famílias de instâncias p6, p3, p4, p5, g3, g3s, g4, g5 ou g6. Se isso não for feito, um trabalho de GPU poderá ficar preso no status `RUNNABLE`.

Os trabalhos que não usam o GPUs podem ser executados em instâncias de GPU. No entanto, eles podem custar mais para serem executados nas instâncias de GPU do que em instâncias semelhantes sem GPU. A depender da vCPU específica, da memória e do tempo necessário, essas tarefas que não são de GPU podem bloquear a execução de tarefas de GPU.

Tópicos

- [Criar um cluster do Kubernetes baseado em GPU no Amazon EKS](#)
- [Criar uma definição de trabalho de GPU do Amazon EKS](#)
- [Executar um trabalho de GPU em seu Cluster do Amazon EKS](#)

Criar um cluster do Kubernetes baseado em GPU no Amazon EKS

Antes de criar um cluster do Kubernetes baseado em GPU no Amazon EKS, você deve ter concluído as etapas em [Conceitos básicos do AWS Batch no Amazon EKS](#). Além disso, considere também o seguinte:

- AWS Batch oferece suporte a tipos de instância com a NVIDIA GPUs.
- Por padrão, AWS Batch seleciona a AMI acelerada do Amazon EKS com a Kubernetes versão que corresponde à versão do plano de controle do cluster do Amazon EKS.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
```

```

    "maxvCpus": 1024,
    "instanceTypes": [
        "p3dn.24xlarge",
        "p4d.24xlarge"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF

$ aws batch create-compute-environment --cli-input-json file:///./batch-eks-gpu-ce.json

```

AWS Batch não gerencia o plug-in do dispositivo NVIDIA GPU em seu nome. Você deve instalar esse plug-in em seu cluster Amazon EKS e permitir que ele atinja os AWS Batch nós. Para obter mais informações, consulte [Habilitando o GPU Support em Kubernetes](#) on. GitHub

Para configurar o plug-in do NVIDIA dispositivo (DaemonSet) para direcionar os AWS Batch nós, execute os comandos a seguir.

```

# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml

```

Não recomendamos que você misture workloads baseadas em computação (CPU e memória) com workloads baseados em GPU nos mesmos pares de ambiente de computação e fila de trabalhos. Isso ocorre porque os trabalhos de computação podem usar a capacidade da GPU.

Para anexar filas de trabalhos, execute os comandos a seguir.

```

$ cat <<EOF > ./batch-eks-gpu-jq.json

```

```
{
  "jobQueueName": "My-Eks-GPU-JQ1",
  "priority": 10,
  "computeEnvironmentOrder": [
    {
      "order": 1,
      "computeEnvironment": "My-Eks-GPU-CE1"
    }
  ]
}
EOF
```

```
$ aws batch create-job-queue --cli-input-json file:///./batch-eks-gpu-jq.json
```

Criar uma definição de trabalho de GPU do Amazon EKS

Somente o `nvidia.com/gpu` é compatível no momento, e o valor do recurso definido deve ser um número inteiro. Não é possível usar frações da GPU. Para obter mais informações, consulte [Programação GPUs](#) na Kubernetes documentação.

Para registrar uma definição de trabalho de GPU para o Amazon EKS, execute os seguintes comandos.

```
$ cat <<EOF > ./batch-eks-gpu-jd.json
{
  "jobDefinitionName": "MyGPUJobOnEks_Smi",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
          "command": ["nvidia-smi"],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi",
              "nvidia.com/gpu": "1"
            }
          }
        }
      ]
    }
  }
}
```

```

    }
  }
}
EOF

```

```
$ aws batch register-job-definition --cli-input-json file:///./batch-eks-gpu-jd.json
```

Executar um trabalho de GPU em seu Cluster do Amazon EKS

O recurso de GPU não é compactável. AWS Batch cria uma especificação de pod para trabalhos de GPU em que o valor da solicitação é igual ao valor dos limites. Isso é um requisito do Kubernetes.

Para enviar um trabalho de GPU, use os comandos a seguir.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJob0nEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
{podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Exibir AWS Batch trabalhos em uma fila de trabalhos

Você pode visualizar e filtrar seus trabalhos no AWS Batch. Esse atributo oferece uma opção para visualizar uma fila de trabalho existente e filtrar seus trabalhos por uma das três opções.

A pesquisa e o filtro são capazes de recuperar trabalhos que não estão em um estado terminal (SUCCEEDED ou FAILED). Quando o estado de um trabalho estiver SUCCEEDED ou FAILED você deverá ser capaz de recuperá-lo por até sete dias. Você ainda pode ver os EventBridge registros de um trabalho CloudWatch ou da Amazon.

Use esse procedimento para listar todos os trabalhos em uma fila de trabalhos no AWS Batch console. Opcionalmente, use o campo Filtrar resultados para restringir os resultados com base nos critérios que você especificar.

1. Navegue até o [console do AWS Batch](#).

2. No painel de navegação, escolha Tarefas.
3. Expanda a lista suspensa Fila de trabalhos e escolha a fila de trabalhos na qual você deseja pesquisar.

 Note

É possível pesquisar trabalhos em apenas uma fila de trabalhos por vez.

4. No campo Filtrar resultados, insira palavras-chave para incluir nos resultados. Você pode usar esse campo para filtrar por nome do trabalho, status ou ID do trabalho. Dependendo da propriedade, pode haver operadores adicionais, como igual (=) ou contém (:), que você deve definir.

 Note

SageMaker As filas de trabalhos de treinamento suportam somente a filtragem por nome do trabalho e ID do trabalho

5. Selecione a opção Pesquisar.

AWS Batch Procure empregos em uma fila de empregos

Você pode pesquisar e filtrar suas vagas AWS Batch usando a Pesquisa de vagas. Esse atributo fornece uma opção para pesquisar em uma fila de trabalhos existente e filtrar seus trabalhos.

A pesquisa e o filtro são capazes de recuperar trabalhos que não estão em um estado terminal (SUCCEEDED ou FAILED). Quando o estado de um trabalho estiver SUCCEEDED ou FAILED você deverá ser capaz de recuperá-lo por até sete dias. Você ainda pode ver os EventBridge registros de um trabalho CloudWatch ou da Amazon.

Para pesquisar usando vários critérios simultaneamente, use o atributo Pesquisa avançada. Por exemplo, é possível incluir qualquer um ou todos os filtros a seguir: Status, Intervalo de datas e Critérios adicionais (como nome do trabalho, definição do trabalho ou ID do trabalho).

Pesquisar AWS Batch empregos (AWS Console)

Use esse procedimento para pesquisar os trabalhos em uma fila de trabalhos no AWS Batch console.

1. Navegue até o [console do AWS Batch](#).
2. No painel de navegação, escolha Tarefas.
3. Ative a Pesquisa avançada.
4. Expanda a lista suspensa Fila de trabalhos e escolha a fila de trabalhos na qual você deseja pesquisar.

 Note

É possível pesquisar trabalhos em apenas uma fila de trabalhos por vez.

5. Para Opções de pesquisa:
 - a. Na lista suspensa Status, você pode escolher um ou mais status para filtrar. Para obter mais informações, consulte [Estados do trabalho](#) e [Status de serviço do trabalho](#).

 Note

Os pais dos trabalhos de matriz são atualizados para PENDING quando qualquer trabalho secundário é atualizado RUNNABLE e permanecem no PENDING status enquanto os trabalhos secundários estão em execução. Para visualizar esses trabalhos, filtre por PENDING status até que todos os trabalhos secundários atinjam um estado terminal.

- b. Escolha o Intervalo de datas para filtrar os resultados com base em um intervalo de data e hora.
 - Escolha o modo Relativo para pesquisar trabalhos com uma data de criação dentro de um intervalo de tempo contado para trás a partir da data e hora atuais.
 - Escolha o Modo absoluto para pesquisar trabalhos com uma data de criação em um intervalo de data e hora especificado por você.
 - c. No campo Critérios adicionais, insira palavras-chave a serem incluídas nos resultados da pesquisa. Por exemplo, você pode usar esse campo para pesquisar por nome do trabalho, definição do trabalho, ID do trabalho ou identificador de compartilhamento. Dependendo da propriedade, pode haver operadores adicionais, como igual (=) ou contém (:), que você deve definir.

 Note

SageMaker As filas de trabalhos de treinamento suportam somente a filtragem por nome do trabalho e ID do trabalho

 Note

Ao filtrar por identificador de compartilhamento, você também pode especificar o status do trabalho. Essa é uma exceção à restrição em que outros filtros excluem a filtragem do status do trabalho.

6. Selecione a opção Pesquisar.

Pesquisar e filtrar AWS Batch vagas (AWS CLI)

Use esse procedimento para listar todos os trabalhos em uma fila de trabalhos com o AWS CLI. Opcionalmente, use o parâmetro `-filters` para restringir os resultados com base nos critérios que você especificar.

Search job queue (AWS CLI)

Você pode usar o comando [list-jobs](#) para pesquisar e filtrar uma fila de trabalhos.

Por exemplo, você pode pesquisar uma fila de trabalhos com base no nome do trabalho:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=JOB_NAME,values="my-job"
```

Filtrar trabalhos por identificador de compartilhamento:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

Ao filtrar por identificador de compartilhamento, você pode incluir o status do trabalho:

```
aws batch list-jobs \  
  --job-queue my-job-queue \  
  --job-status RUNNING \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

No comando anterior, faça as seguintes alterações:

- *my-job-queue* Substitua pelo nome da sua fila de trabalhos.
- *my-job* Substitua pelo nome do seu trabalho.
- *my-share* Substitua pelo identificador de compartilhamento pelo qual você deseja filtrar.

Search service job queue (AWS CLI)

Você pode usar o [list-service-jobs](#) comando para pesquisar e filtrar uma fila de trabalhos de serviço.

Por exemplo, você pode pesquisar uma fila de trabalhos de serviço com base no nome do trabalho:

```
aws batch list-service-jobs \  
  --job-queue my-sm-queue \  
  --filters name=JOB_NAME,values="my-sm-job"
```

Filtrar trabalhos de serviço por identificador de compartilhamento:

```
aws batch list-service-jobs \  
  --job-queue my-sm-queue \  
  --filters name=SHARE_IDENTIFIER,values="my-share"
```

No comando anterior, faça as seguintes alterações:

- *my-sm-queue* Substitua pelo nome da sua fila de trabalhos de serviço.
- *my-sm-job* Substitua pelo nome do seu trabalho de serviço.
- *my-share* Substitua pelo identificador de compartilhamento pelo qual você deseja filtrar.

Modos de rede para AWS Batch trabalhos

A tabela a seguir descreve os modos de rede e o uso típico dos tipos de AWS Batch trabalho. Consulte os links na coluna “Tipo de trabalho” para obter mais detalhes sobre considerações e comportamentos.

Tipo de tarefa	Modos de rede compatíveis	Uso típico
Trabalho simples ECS-EC2	host	Usado para workloads em lote mais escaláveis e embaraçosamente paralelos que exigem apenas a saída para o vpc definido no ambiente de computação.
Trabalho paralelo de vários nós ECS-EC2	awsvpc	Usado para workloads distribuídos de vários hosts (nós) fortemente acoplados, modelados como um único trabalho com comunicações coordenadas entre os nós de tarefa.
Trabalho simples ECS-Fargate	awsvpc	Verdadeiramente em tecnologia sem servidor para workloads em lotes embaraçosamente paralelos. Normalmente o menor TCO e o maior modelo de trabalho de isolamento de contêineres.
Trabalho simples EKS-EC2	host e pod	Usado para workloads em lotes altamente escaláveis e embaraçosamente paralelos que exigem apenas a saída para a vpc definida no

Tipo de tarefa	Modos de rede compatíveis	Uso típico
		ambiente de computação. O padrão é rede de host.
Trabalho paralelo de vários nós EKS-EC2	host e pod	Usado para workloads distribuídos de vários hosts (nós) fortemente acoplados, modelados como um único trabalho com comunicações coordenadas entre os nós de pod. O padrão é rede de host.

Exibir registros de AWS Batch trabalhos em CloudWatch Registros

Você pode [configurar seus AWS Batch trabalhos](#) para enviar informações de registro para o Amazon CloudWatch Logs. Assim, você pode visualizar logs diferentes de trabalhos em um local conveniente. Para obter mais informações, consulte [Usando CloudWatch registros com AWS Batch](#).

Você também pode usar os registros de trabalhos no AWS Batch console para monitorar ou solucionar problemas de um AWS Batch trabalho.

1. Abra o [AWS Batch Console](#).
2. Escolha Trabalhos. Para obter informações mais detalhadas sobre como classificar e filtrar trabalhos na fila de trabalhos, consulte [Exibir AWS Batch trabalhos em uma fila de trabalhos](#) e [Pesquisar por trabalhos em uma fila de trabalho](#)
3. Em Fila de Trabalhos, escolha a fila de trabalhos desejada.

Tip

Se houver vários trabalhos na fila de trabalhos, você poderá ativar Pesquisa e Filtragem para encontrar um trabalho mais rapidamente. Para obter mais informações, consulte [AWS Batch Procure empregos em uma fila de empregos](#).

4. Em Status, escolha os status do trabalho que você deseja exibir.
5. Escolha o trabalho que você deseja e a página Detalhes será aberta.

6. Na página Detalhes, role para baixo até Nome do fluxo de logs e escolha o link. O link abre a página Amazon CloudWatch Logs do trabalho.
7. (Opcional) Se esta for a primeira vez que você visualiza os logs, poderá ser solicitada uma autorização.

Para Autorização necessária **OK**, insira e escolha Autorizar para aceitar CloudWatch cobranças da Amazon.

 Note

Para revogar sua autorização para CloudWatch cobranças:

1. No painel de navegação esquerdo, escolha Permissões.
2. Em Logs de Trabalho, escolha Editar.
3. Desmarque a caixa de CloudWatch seleção Autorizar o uso do Batch.
4. Escolha Salvar alterações.

Revise as informações do AWS Batch trabalho

Você pode analisar as informações sobre o trabalho do AWS Batch , como status, definição de trabalho e informações do contêiner.

1. Abra o [AWS Batch Console](#).
2. Escolha Trabalhos.
3. Em Fila de Trabalhos, escolha a fila de trabalhos desejada.

 Tip

Se houver vários trabalhos na fila de trabalhos, ative Pesquisa e filtragem para encontrar um trabalho mais rapidamente. Para obter mais informações, consulte [AWS Batch Procure empregos em uma fila de empregos](#).

4. Escolha o trabalho desejado.

 Note

Você também pode usar o AWS Command Line Interface (AWS CLI) para ver detalhes sobre um AWS Batch trabalho. Para obter mais informações, consulte [describe-jobs](#) na [Referência de Comandos AWS CLI](#).

Segurança em AWS Batch

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de data centers e arquiteturas de rede criados para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [Modelo de Responsabilidade Compartilhada](#) descreve isso como segurança da nuvem e segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços no Nuvem AWS. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam AWS Batch, consulte [AWS Serviços no escopo do programa de conformidade AWS](#) .
- **Segurança na nuvem**: sua responsabilidade é determinada pelo serviço da AWS que você usa. Você também é responsável por outros fatores, inclusive a sensibilidade de seus dados, os requisitos da sua empresa, leis e regulamentos aplicáveis.

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar AWS Batch. Os tópicos a seguir mostram como configurar para atender AWS Batch aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus AWS Batch recursos.

Tópicos

- [Identity and Access Management para AWS Batch](#)
- [AWS Batch Políticas, funções e permissões do IAM](#)
- [AWS Batch Função de execução do IAM](#)
- [Criar uma nuvem privada virtual](#)
- [Use um endpoint de interface para acessar AWS Batch](#)
- [Validação de conformidade para AWS Batch](#)
- [Segurança da infraestrutura em AWS Batch](#)
- [Prevenção do problema "confused deputy" entre serviços](#)
- [Registrando chamadas de AWS Batch API com AWS CloudTrail](#)

- [Solucionar problemas do IAM AWS Batch](#)

Identity and Access Management para AWS Batch

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) a usar AWS Batch os recursos. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

- [Público](#)
- [Autenticação com identidades](#)
- [Gerenciar o acesso usando políticas](#)
- [Como AWS Batch funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para AWS Batch](#)
- [AWS políticas gerenciadas para AWS Batch](#)

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte [Solucionar problemas do IAM AWS Batch](#)).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte [Como AWS Batch funciona com o IAM](#))
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte [Exemplos de políticas baseadas em identidade para AWS Batch](#))

Autenticação com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte [Como fazer login em sua Conta da AWS](#) no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte [AWS Signature Version 4 para solicitações de API](#) no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte [O que é o IAM Identity Center?](#) no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um [usuário do IAM](#) é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte [Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias](#) no Guia do usuário do IAM.

Um [grupo do IAM](#) especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte [Casos de uso de usuários do IAM](#) no Guia do usuário do IAM.

Perfis do IAM

Uma [perfil do IAM](#) é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função [mudando de um usuário para uma função do IAM \(console\)](#) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte [Métodos para assumir um perfil](#) no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. Entre os exemplos estão políticas de confiança de perfil do IAM e políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. É necessário [especificar uma entidade principal](#) em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte [Políticas de controle de serviço](#) no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte [Políticas de controle de recursos \(RCPs\)](#) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, consulte [Políticas de sessão](#) no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

Como AWS Batch funciona com o IAM

Antes de usar o IAM para gerenciar o acesso AWS Batch, saiba com quais recursos do IAM estão disponíveis para uso AWS Batch.

Recursos do IAM que você pode usar com AWS Batch

Recurso do IAM	AWS Batch apoio
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC (tags em políticas)	Sim
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Sim
Perfis vinculados a serviço	Sim

Para ter uma visão de alto nível de como AWS Batch e outros AWS serviços funcionam com a maioria dos recursos do IAM, consulte [AWS os serviços que funcionam com o IAM](#) no Guia do usuário do IAM.

Políticas baseadas em identidade para AWS Batch

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas

políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente](#) no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Saiba mais sobre todos os elementos que podem ser usados em uma política JSON consultando [Referência de Elementos de Política JSON do IAM](#) no Guia do Usuário do IAM.

Exemplos de políticas baseadas em identidade para o AWS Batch

Para ver exemplos de políticas AWS Batch baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS Batch](#)

Ações políticas para AWS Batch

Compatível com ações de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Action` de uma política JSON descreve as ações que podem ser usadas para permitir ou negar acesso em uma política. Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de AWS Batch ações, consulte [Ações definidas por AWS Batch](#) na Referência de autorização de serviço.

As ações de política AWS Batch usam o seguinte prefixo antes da ação:

```
batch
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
    "batch:action1",  
    "batch: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": "batch:Describe"
```

Para ver exemplos de políticas AWS Batch baseadas em identidade, consulte [Exemplos de políticas baseadas em identidade para AWS Batch](#)

Recursos políticos para AWS Batch

Compatível com recursos de políticas: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu [nome do recurso da Amazon \(ARN\)](#). Para ações que não oferecem compatibilidade com permissões em nível de recurso, use um curinga (*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de AWS Batch recursos e seus ARNs, consulte [Recursos definidos por AWS Batch](#) na Referência de autorização de serviço. Para saber com quais ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pelo AWS Batch](#).

Chaves de condição de políticas para AWS Batch

Compatível com chaves de condição de política específicas de serviço: sim

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

O elemento `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de AWS Batch condição, consulte [Chaves de condição AWS Batch](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas por AWS Batch](#).

Controle de acesso baseado em atributos (ABAC) com AWS Batch

Compatível com ABAC (tags em políticas): sim

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte [Definir permissões com autorização do ABAC](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso por atributo \(ABAC\)](#) no Guia do usuário do IAM.

Use credenciais temporárias com AWS Batch

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte [Credenciais de segurança temporárias no IAM](#) e [Serviços da Serviços da AWS que funcionam com o IAM](#) no Guia do usuário do IAM.

Permissões de entidade principal entre serviços para o AWS Batch

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Sessões de acesso direto](#).

Funções de serviço para AWS Batch

Compatível com perfis de serviço: sim

O perfil de serviço é um [perfil do IAM](#) que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do Usuário do IAM.

Warning

Alterar as permissões de uma função de serviço pode interromper AWS Batch a funcionalidade. Edite perfis de serviço somente quando o AWS Batch fornecer orientação para fazê-lo.

Funções vinculadas a serviços para AWS Batch

Compatibilidade com perfis vinculados a serviços: sim

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços da AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a serviço desse serviço.

Exemplos de políticas baseadas em identidade para AWS Batch

Por padrão, usuários e perfis não têm permissão para criar ou modificar recursos do AWS Batch . Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte [Criar políticas do IAM \(console\)](#) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos por AWS Batch, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição AWS Batch na Referência de Autorização de Serviço](#).

Tópicos

- [Práticas recomendadas de política](#)
- [Usando o AWS Batch console](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)

Práticas recomendadas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir AWS Batch recursos 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 com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso. Para saber mais, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para funções de trabalho](#) no Guia do usuário do IAM.
- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como CloudFormation. Para saber mais, consulte [Elementos da política JSON do IAM: condição](#) no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas

sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do Usuário do IAM.

- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte [Configuração de acesso à API protegido por MFA](#) no Guia do Usuário do IAM.

Para saber mais sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

Usando o AWS Batch console

Para acessar o AWS Batch console, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os AWS Batch recursos em seu Conta da AWS. Caso crie uma política baseada em identidade mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam à operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o AWS Batch console, anexe também a política AWS Batch ConsoleAccess ou a política ReadOnly AWS gerenciada às entidades. Para obter informações, consulte [Adicionar permissões a um usuário](#) no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
    }
  ]
}

```

AWS políticas gerenciadas para AWS Batch

Você pode usar políticas AWS gerenciadas para simplificar o gerenciamento do acesso à identidade da sua equipe e dos recursos provisionados AWS . AWS as políticas gerenciadas abrangem uma variedade de casos de uso comuns, estão disponíveis por padrão em sua AWS conta e são mantidas e atualizadas em seu nome. Você não pode alterar as permissões nas políticas AWS gerenciadas. Se o requisito for maior flexibilidade, você também pode optar por criar políticas gerenciadas pelo cliente do IAM. Dessa forma, você pode fornecer a sua equipe recursos provisionados apenas com as permissões exatas de que ela precisa.

Para obter mais informações sobre políticas AWS gerenciadas, consulte [políticas AWS gerenciadas](#) no Guia do usuário do IAM.

AWS os serviços mantêm e atualizam as políticas AWS gerenciadas em seu nome. Periodicamente, AWS os serviços adicionam permissões adicionais a uma política AWS gerenciada. AWS as políticas gerenciadas provavelmente são atualizadas quando o lançamento ou operação de um novo recurso é disponibilizado. Essas atualizações afetam automaticamente todas as identidades (usuários, grupos e perfis) em que a política está anexada. No entanto, eles não removem as permissões nem interrompem as permissões existentes.

Além disso, AWS oferece suporte a políticas gerenciadas para funções de trabalho que abrangem vários serviços. Por exemplo, a política `ReadOnlyAccess` AWS gerenciada fornece acesso somente de leitura a todos os AWS serviços e recursos. Quando um serviço lança um novo recurso, AWS adiciona permissões somente de leitura para novas operações e recursos. Para obter uma lista e descrições das políticas de perfis de trabalho, consulte [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.

AWS política gerenciada: `BatchServiceRolePolicy`

A política `BatchServiceRolePolicy` gerenciada do IAM é usada pela função [AWSBatchServiceRoleForBatch](#) vinculada ao serviço. Isso permite AWS Batch realizar ações em seu nome. Não é possível anexar essa política a suas entidades do IAM. Para obter mais informações, consulte [Usar perfis vinculados a serviço do AWS Batch](#).

Essa política permite AWS Batch concluir as seguintes ações em recursos específicos:

- `autoscaling`— Permite AWS Batch criar e gerenciar recursos do Amazon EC2 Auto Scaling. AWS Batch cria e gerencia grupos do Amazon EC2 Auto Scaling para a maioria dos ambientes computacionais.
- `ec2`— Permite AWS Batch controlar o ciclo de vida das instâncias do Amazon EC2, bem como criar e gerenciar modelos e tags de lançamento. AWS Batch cria e gerencia solicitações do EC2 Spot Fleet para alguns ambientes computacionais EC2 Spot.
- `ecs`- Permite AWS Batch criar e gerenciar clusters, definições de tarefas e tarefas do Amazon ECS para execução de trabalhos.
- `eks`- Permite AWS Batch descrever o recurso de cluster Amazon EKS para validações.

- **iam**— Permite AWS Batch validar e passar funções fornecidas pelo proprietário para o Amazon EC2, o Amazon EC2 Auto Scaling e o Amazon ECS.
- **logs**— Permite AWS Batch criar e gerenciar grupos de registros e fluxos de registros para AWS Batch trabalhos.

Para ver o JSON da política, consulte o [Guia BatchServiceRolePolicyde referência de políticas AWS gerenciadas](#).

AWS política gerenciada: AWSBatchServiceRolePolicyForSageMaker

[AWSServiceRoleForAWSBatchWithSagemaker](#) permite AWS Batch realizar ações em seu nome. Não é possível anexar essa política a suas entidades do IAM. Para obter mais informações, consulte [Usar perfis vinculados a serviço do AWS Batch](#).

Essa política permite AWS Batch concluir as seguintes ações em recursos específicos:

- **sagemaker**— Permite AWS Batch gerenciar trabalhos de treinamento de SageMaker IA e outros recursos de SageMaker IA.
- **iam:PassRole**— Permite AWS Batch passar funções de execução definidas pelo cliente para a SageMaker IA para execução de tarefas. A restrição de recursos permite passar funções para serviços de SageMaker IA.

Para ver o JSON da política, consulte o [Guia AWSBatchServiceRolePolicyForSageMakerde referência de políticas AWS gerenciadas](#).

AWS política gerenciada: AWSBatchServiceRolepolítica

A política de permissões de função nomeada AWSBatchServiceRole AWS Batch permite concluir as seguintes ações em recursos específicos:

A política AWSBatchServiceRolegerenciada do IAM geralmente é usada por uma função chamada AWSBatchServiceRolee inclui as seguintes permissões. Ao seguir o aviso de segurança padrão de concessão de privilégios mínimos, a política gerenciada AWSBatchServiceRole pode ser usada como um guia. Se qualquer permissão concedida na política gerenciada não for necessária para o seu caso de uso, crie uma política personalizada e adicione apenas as permissões necessárias. Essa política e função AWS Batch gerenciadas podem ser usadas com a maioria dos tipos de ambiente computacional, mas o uso de funções vinculadas a serviços é preferível para uma experiência gerenciada menos propensa a erros, com melhor escopo e melhor definição.

- **autoscaling**— Permite AWS Batch criar e gerenciar recursos do Amazon EC2 Auto Scaling. AWS Batch cria e gerencia grupos do Amazon EC2 Auto Scaling para a maioria dos ambientes computacionais.
- **ec2**— Permite AWS Batch gerenciar o ciclo de vida das instâncias do Amazon EC2, bem como criar e gerenciar modelos e tags de lançamento. AWS Batch cria e gerencia solicitações do EC2 Spot Fleet para alguns ambientes computacionais EC2 Spot.
- **ecs**— Permite AWS Batch criar e gerenciar clusters, definições de tarefas e tarefas do Amazon ECS para execução de trabalhos.
- **iam**— Permite AWS Batch validar e passar funções fornecidas pelo proprietário para o Amazon EC2, o Amazon EC2 Auto Scaling e o Amazon ECS.
- **logs**— Permite AWS Batch criar e gerenciar grupos de registros e fluxos de registros para AWS Batch trabalhos.

Para ver o JSON da política, consulte o [Guia AWSBatchServiceRolede referência de políticas AWS gerenciadas](#).

AWS política gerenciada: AWSBatchFullAccess

A AWSBatchFullAccesspolítica concede AWS Batch às ações acesso total aos AWS Batch recursos. Também concede acesso para descrever e listar ações para serviços Amazon EC2, Amazon ECS, Amazon EKS e CloudWatch IAM. Isso é para que as identidades do IAM, sejam usuários ou funções, possam visualizar os recursos AWS Batch gerenciados que foram criados em seu nome. Por fim, essa política também permite que perfis selecionados do IAM sejam passadas para esses serviços.

Você pode anexar AWSBatchFullAccessàs suas entidades do IAM. AWS Batch também anexa essa política a uma função de serviço que permite AWS Batch realizar ações em seu nome.

Para ver o JSON da política, consulte o [Guia AWSBatchFullAccessde referência de políticas AWS gerenciadas](#).

AWS Batch atualizações nas políticas AWS gerenciadas

Veja detalhes sobre as atualizações das políticas AWS gerenciadas AWS Batch desde que esse serviço começou a rastrear essas alterações. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página Histórico do AWS Batch documento.

Alteração	Descrição	Data
AWSBatchServiceRolePolicyForSageMaker política adicionada	Foi adicionada uma nova política AWS gerenciada para a função <code>AWSBatchServiceRolePolicyForSageMaker</code> vinculada ao serviço que permite gerenciar AWS Batch a SageMaker IA em seu nome.	31 de julho de 2025
BatchServiceRolePolicy política atualizada	Atualização para adicionar suporte à descrição do histórico de solicitações da Frota Spot e de atividades do Amazon EC2 Auto Scaling .	5 de dezembro de 2023
AWSBatchServiceRole política adicionada	Atualizado para adicionar declarações IDs, conceder AWS Batch permissões para <code>ec2:DescribeSpotFleetRequestHistory</code> <code>autoscaling:DescribeScalingActivities</code> e.	5 de dezembro de 2023
BatchServiceRolePolicy política atualizada	Atualizada para adicionar suporte à descrição de clusters do Amazon EKS.	20 de outubro de 2022
AWSBatchFullAccess política atualizada	Atualizada para adicionar suporte a listagem e descrição de clusters do Amazon EKS.	20 de outubro de 2022
BatchServiceRolePolicy política atualizada	Atualizada para suportar grupos de reserva de capacidade do Amazon EC2 gerenciados por AWS Resource Groups. Para obter mais informações, consulte Trabalhar com grupos de reserva de capacidade , no Guia do usuário do Amazon EC2.	18 de maio de 2022

Alteração	Descrição	Data
BatchServiceRolePolicy e AWSBatchServiceRole políticas atualizadas	Atualizado para adicionar suporte para descrever o status das instâncias AWS Batch gerenciadas no Amazon EC2 para que as instâncias não íntegras sejam substituídas.	6 de dezembro de 2021
BatchServiceRolePolicy política atualizada	Atualizada para adicionar suporte para grupos de posicionamento, reserva de capacidade, GPU elástica e recursos do Elastic Inference no Amazon EC2.	26 de março de 2021
BatchServiceRolePolicy política adicionada	Com a política BatchServiceRolePolicy gerenciada para a função AWSServiceRoleForBatch vinculada ao serviço, você pode usar uma função vinculada ao serviço gerenciada por AWS Batch. Com essa política, você não precisa manter sua própria função para uso em ambientes de computação.	10 de março de 2021
AWSBatchFullAccess - adicionar permissão para adicionar uma função vinculada ao serviço	Adicione permissões do IAM para permitir que a função AWSServiceRoleForBatch vinculada ao serviço seja adicionada à conta.	10 de março de 2021
AWS Batch começou a rastrear alterações	AWS Batch começou a rastrear as mudanças em suas políticas AWS gerenciadas.	10 de março de 2021

AWS Batch Políticas, funções e permissões do IAM

Por padrão, os usuários não têm permissão para criar ou modificar AWS Batch recursos ou realizar tarefas usando a AWS Batch API, o AWS Batch console ou AWS CLI. Para permitir que usuários realizem essas ações, crie políticas do IAM que concedam permissão aos usuários para recursos e

operações API específicas. Em seguida, anexe as políticas aos usuários ou grupos que exijam tais permissões.

Quando você anexa uma política a um usuário ou grupo de usuários, a política concede ou nega as permissões para tarefas específicas em recursos específicos. Para mais informações, consulte [Permissões e Políticas](#) no Guia de Usuário do IAM. Para mais informações sobre como gerenciar e criar políticas personalizadas do IAM, consulte [Gerenciamento de Políticas do IAM](#).

AWS Batch faz chamadas para outras pessoas Serviços da AWS em seu nome. Como resultado, AWS Batch deve se autenticar usando suas credenciais. Mais especificamente, AWS Batch autentica criando uma função e uma política do IAM que fornecem essas permissões. Em seguida, associa o perfil aos seus ambientes de computação quando criados. Para obter mais informações [Perfil de instância do Amazon ECS](#), consulte [Funções do IAM, Uso de funções vinculadas ao serviço](#) e [Criação de uma função para delegar permissões a um AWS serviço](#) no Guia do usuário do IAM.

Tópicos

- [Estrutura da política do IAM](#)
- [Recurso: exemplos de políticas para AWS Batch](#)
- [Recurso: política AWS Batch gerenciada](#)

Estrutura da política do IAM

Os tópicos a seguir explicam a estrutura de uma política do IAM.

Tópicos

- [Sintaxe da política](#)
- [Ações de API para AWS Batch](#)
- [Nomes de recursos da Amazon para AWS Batch](#)
- [Confirmar se os usuários têm as permissões necessárias](#)

Sintaxe da política

A política do IAM é um documento JSON que consiste em uma ou mais declarações. Cada instrução é estruturada da maneira a seguir.

```
{
```

```
"Statement": [{
  "Effect": "effect",
  "Action": "action",
  "Resource": "arn",
  "Condition": {
    "condition": {
      "key": "value"
    }
  }
}]
}
```

Há quatro elementos principais que compõem uma declaração:

- Efeito: o efeito pode ser Allow ou Deny. Por padrão, os usuários não têm permissão para usar recursos e ações de API. Então, todas as solicitações são negadas. Um permitir explícito substitui o padrão. Uma negar explícito substitui todas as permissões.
- Ação: a ação é a ação de API específica para a qual você está concedendo ou negando permissão. Para obter instruções sobre como especificar a ação, consulte [Ações de API para AWS Batch](#).
- Recurso: o recurso afetado pela ação. Com algumas ações de API do AWS Batch é possível incluir recursos específicos na política que podem ser criados ou modificados pela ação. Para especificar um recurso na declaração, use o respectivo nome de recurso da Amazon (ARN). Para obter mais informações, consulte [Permissões em nível de recurso suportadas para ações de API AWS Batch](#) e [Nomes de recursos da Amazon para AWS Batch](#). Se a operação da AWS Batch API atualmente não oferecer suporte a permissões em nível de recurso, inclua um caractere curinga (*) para especificar que todos os recursos podem ser afetados pela ação.
- Condição: condições são opcionais. Elas podem ser usadas para controlar quando a política está em vigor.

Para obter mais informações sobre exemplos de declarações de política do IAM para AWS Batch, consulte [Recurso: exemplos de políticas para AWS Batch](#).

Ações de API para AWS Batch

Em uma declaração de política do IAM, é possível especificar qualquer ação de API de qualquer serviço que dê suporte ao IAM. Para AWS Batch, use o seguinte prefixo com o nome da ação da API: batch: (por exemplo, batch:SubmitJob ebatch:CreateComputeEnvironment).

Para especificar várias ações em uma única declaração, separe cada ação com uma vírgula.

```
"Action": ["batch:action1", "batch:action2"]
```

Você também pode especificar várias ações incluindo um curinga (*). Por exemplo, é possível especificar todas as ações com um nome começando com a palavra "Describe".

```
"Action": "batch:Describe*"
```

Para especificar todas as ações AWS Batch da API, inclua um caractere curinga (*).

```
"Action": "batch:*"
```

Para ver uma lista de AWS Batch ações, consulte [Ações](#) na Referência AWS Batch da API.

Nomes de recursos da Amazon para AWS Batch

Cada declaração de política do IAM se aplica aos recursos que você especifica usando seus Amazon Resource Names (ARNs).

Um nome do recurso da Amazon (ARN) tem a seguinte sintaxe geral:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

O serviço (por exemplo, batch).

region

O Região da AWS para o recurso (por exemplo, us-east-2).

account

O Conta da AWS ID, sem hífen (por exemplo, 123456789012).

resourceType

O tipo de recurso (por exemplo, compute-environment).

resourcePath

Um caminho que identifica o recurso. É possível usar um curinga (*) nos caminhos.

AWS Batch Atualmente, as operações de API oferecem suporte a permissões em nível de recurso em várias operações de API. Para obter mais informações, consulte [Permissões em nível de recurso suportadas para ações de API AWS Batch](#). Para especificar todos os recursos, ou se uma ação de API específica não for compatível ARNs, inclua um caractere curinga (*) no Resource elemento.

```
"Resource": "*"
```

Confirmar se os usuários têm as permissões necessárias

Antes de colocar uma política do IAM em vigor, verifique se ela concede aos usuários as permissões para usar as ações e recursos de API específicos de que eles precisam.

Para fazer isso, primeiro, crie um usuário do para fins de teste e anexe a política do IAM ao usuário de teste. Em seguida, faça uma solicitação como o usuário de teste. Você pode fazer solicitações de teste no console ou com a AWS CLI.

Note

Você também pode testar as políticas usando o [Simulador de políticas do IAM](#). Para obter mais informações sobre o simulador de políticas, consulte [Como trabalhar com o simulador de políticas do IAM](#) no Guia do usuário do IAM.

Caso a política não conceda ao usuário as permissões que você esperava ou caso ela seja muito permissiva, você pode ajustar a política conforme necessário. Teste novamente até obter os resultados desejados.

Important

Pode levar alguns minutos para que as alterações de política sejam propagadas até entrarem em vigor. Por isso, recomendamos que você aguarde pelo menos cinco minutos antes de testar as atualizações da política.

Caso uma verificação de autorização falhe, a solicitação retorna uma mensagem codificada com informações de diagnóstico. É possível decodificar a mensagem usando a ação `DecodeAuthorizationMessage`. Para obter mais informações, consulte [DecodeAuthorizationMessage](#) na Referência AWS Security Token Service da API e [decode-authorization-message](#) na Referência de AWS CLI Comandos.

Recurso: exemplos de políticas para AWS Batch

Você pode criar políticas do IAM específicas para restringir chamadas e recursos a que os usuários em sua conta têm acesso. Em seguida, você pode anexar tais políticas aos usuários.

Ao anexar uma política a um usuário ou grupo de usuários, ela concede ou nega aos usuários permissão para realizar tarefas específicas em recursos específicos. Para mais informações, consulte [Permissões e Políticas](#) no Guia de Usuário do IAM. Para instruções sobre como gerenciar e criar políticas do IAM personalizadas, consulte [Gerenciando Políticas do IAM](#).

Os exemplos a seguir mostram declarações de política que você pode usar para controlar as permissões que os usuários têm para o AWS Batch.

Exemplos

- [Recurso: Acesso somente para leitura para AWS Batch](#)
- [Recurso: Restringir ao usuário POSIX, imagem do Docker, nível de privilégio e perfil no envio do trabalho](#)
- [Recurso: Restringir ao prefixo de definição de trabalho no envio de trabalho](#)
- [Recurso: restringir a uma fila de trabalho](#)
- [Ação Negar quando todas as condições correspondem a strings](#)
- [Recurso: Negar ação quando qualquer chave de condição corresponder às strings](#)
- [Recurso: usar a chave de condição batch:ShareIdentifier](#)
- [Gerencie recursos de SageMaker IA com AWS Batch](#)
- [Recurso: restrinja o envio de trabalhos por meio de tags de recursos na definição de trabalhos e na fila de trabalhos](#)

Recurso: Acesso somente para leitura para AWS Batch

A política a seguir concede aos usuários permissões para usar todas as ações AWS Batch da API com um nome que começa com `Describe List` e.

Os usuários não podem executar nenhuma ação nesses recursos ou criar novos recursos, a menos que outra declaração conceda permissão para eles fazerem isso. Por padrão, eles não têm permissão para usar ações de API.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:Describe*",
        "batch:List*",
        "batch:Get*"
      ],
      "Resource": "*"
    }
  ]
}
```

Recurso: Restringir ao usuário POSIX, imagem do Docker, nível de privilégio e perfil no envio do trabalho

A política a seguir permite que um usuário POSIX gerencie seu próprio conjunto de definições de tarefas restritos.

Use a primeira e a segunda declarações para registrar e cancelar o registro de qualquer nome de definição de tarefa cujo nome esteja prefixado com. *JobDefA_*

A primeira afirmação também usa chaves de contexto condicional para restringir o usuário de POSIX, status de privilegiado e valores de imagem de contêiner dentro das `containerProperties` de uma definição de tarefa. Para obter mais informações, consulte [RegisterJobDefinition](#) na Referência de APIs do AWS Batch . Neste exemplo, as definições de tarefas só podem ser registradas quando o usuário POSIX está definido como `nobody`. O sinalizador privilegiado está definido como `false`. Por exemplo, a imagem é definida como `myImage` em um repositório do Amazon ECR.

 Important

O docker resolve o parâmetro `user` para o `uid` desse usuário de dentro da imagem do contêiner. Na maioria dos casos, isso é encontrado no arquivo `/etc/passwd` dentro da imagem do contêiner. Essa resolução de nomes pode ser evitada usando valores diretos de `uid` tanto na definição de tarefa quanto nas políticas do IAM associadas. Tanto as operações

de API do AWS Batch quanto as chaves condicionais do `batch:User` do IAM suportam valores numéricos.

Use a terceira afirmação para restringir a apenas uma função específica a uma definição de trabalho.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:User": [
            "nobody"
          ],
          "batch:Image": [
            "999999999999.dkr.ecr.us-east-2.amazonaws.com/myImage"
          ]
        },
        "Bool": {
          "batch:Privileged": "false"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "batch:DeregisterJobDefinition"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
      ]
    }
  ],
}
```

```

    {
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::999999999999:role/MyBatchJobRole"
      ]
    }
  ]
}

```

Recurso: Restringir ao prefixo de definição de trabalho no envio de trabalho

Use a política a seguir para enviar trabalhos para qualquer fila de trabalhos com qualquer nome de definição de trabalho que comece com *JobDefA*.

Important

Para o escopo de acesso em nível de recurso para envio de tarefa, você deve fornecer os tipos de recursos de definição de tarefa e fila de tarefas.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:111122223333:job-definition/JobDefA_*",
        "arn:aws:batch:us-east-2:111122223333:job-queue/*"
      ]
    }
  ]
}

```

Recurso: restringir a uma fila de trabalho

Use a política a seguir para enviar trabalhos para uma fila de trabalhos específica nomeada `queue1` com qualquer nome de definição de trabalho.

Important

Para o escopo de acesso em nível de recurso para envio de tarefa, você deve fornecer os tipos de recursos de definição de tarefa e fila de tarefas.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:888888888888:job-definition/*",
        "arn:aws:batch:us-east-2:888888888888:job-queue/queue1"
      ]
    }
  ]
}
```

Ação Negar quando todas as condições correspondem a strings

A política a seguir nega acesso à operação da [RegisterJobDefinition](#) API quando a chave de condição `batch:Image` (ID da imagem do contêiner) é `"string1"` e a chave de condição `batch:LogDriver` (driver de registro do contêiner) é `"string2"`. AWS Batch avalia as chaves de condição em cada contêiner. Quando um trabalho abrange vários contêineres, como um trabalho paralelo de vários nós, é possível que os contêineres tenham configurações diferentes. Se várias chaves de condição forem avaliadas em uma declaração, elas serão combinadas usando a lógica AND. Portanto, se alguma das várias chaves de condição não corresponder a um contêiner, o efeito

Deny não será aplicado a esse contêiner. Em vez disso, um contêiner diferente no mesmo trabalho pode ser negado.

Para obter a lista de chaves de condição para AWS Batch, consulte [Chaves de condição para AWS Batch](#) na Referência de autorização de serviço. Com exceção de `batch:ShareIdentifier`, todas as chaves de condição do batch podem ser usadas dessa forma. A chave de condição `batch:ShareIdentifier` é definida para um trabalho, não para uma definição de trabalho.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```

Recurso: Negar ação quando qualquer chave de condição corresponder às strings

A política a seguir nega acesso à operação da [RegisterJobDefinition](#) API quando a chave de condição `batch:Image` (ID da imagem do contêiner) é "*string1*" ou a chave de condição

`batch:LogDriver` (driver de registro do contêiner) é `""string2`. Quando um trabalho abrange vários contêineres, como um trabalho paralelo de vários nós, é possível que os contêineres tenham configurações diferentes. Se várias chaves de condição forem avaliadas em uma declaração, elas serão combinadas usando a lógica AND. Portanto, se alguma das várias chaves de condição não corresponder a um contêiner, o efeito Deny não será aplicado a esse contêiner. Em vez disso, um contêiner diferente no mesmo trabalho pode ser negado.

Para obter a lista de chaves de condição para AWS Batch, consulte [Chaves de condição para AWS Batch](#) na Referência de autorização de serviço. Com exceção de `batch:ShareIdentifier`, todas as chaves de condição do batch podem ser usadas dessa forma. (A chave de condição `batch:ShareIdentifier` é definida para um trabalho, não para uma definição de trabalho.)

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:Image": [
            "string1"
          ]
        }
      }
    }
  ]
}
```

```

    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:LogDriver": [
            "string2"
          ]
        }
      }
    }
  ]
}

```

Recurso: usar a chave de condição **batch:ShareIdentifier**

Use a política a seguir para enviar trabalhos que usam a definição de trabalho jobDefA para a fila de trabalhos jobqueue1 com o identificador de compartilhamento lowCpu.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:555555555555:job-definition/JobDefA",
        "arn:aws:batch:us-east-2:555555555555:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {

```

```

        "batch:ShareIdentifier": [
            "lowCpu"
        ]
    }
}
]
}

```

Gerencie recursos de SageMaker IA com AWS Batch

Essa política permite AWS Batch gerenciar recursos de SageMaker IA.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": "arn:aws:iam::*:role/*AWSServiceRoleForAWSBatchWithSagemaker",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "sagemaker-queuing.batch.amazonaws.com"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",

```

```

        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": [
                    "sagemaker.amazonaws.com"
                ]
            }
        }
    }
}

```

Recurso: restrinja o envio de trabalhos por meio de tags de recursos na definição de trabalhos e na fila de trabalhos

Use a política a seguir para enviar trabalhos somente quando a fila de trabalhos tiver a tag `Environment=dev` e a definição do trabalho tiver a tag `Project=calc`. Essa política demonstra como usar tags de recursos para controlar o acesso aos AWS Batch recursos durante o envio de trabalhos.

Important

Ao enviar trabalhos com políticas que avaliam tags de recursos de definição de trabalho, você deve enviar trabalhos usando o formato de revisão de definição de trabalho (`job-definition:revision`). Se você enviar trabalhos sem especificar uma revisão, as tags de definição de trabalho não serão avaliadas, potencialmente ignorando os controles de acesso pretendidos. O `:*` padrão no ARN do recurso exige que os envios incluam uma revisão, garantindo que as políticas de tags sejam sempre aplicadas de forma eficaz.

Essa política usa duas declarações separadas porque aplica condições de tag diferentes a tipos de recursos diferentes. Para o escopo de acesso em nível de recurso para envio de tarefa, você deve fornecer os tipos de recursos de definição de tarefa e fila de tarefas.

```

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

```



```

    "Action": "batch:SubmitJob",
    "Resource": "arn:aws:batch:*:*:job-queue/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Environment": "dev"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": "batch:SubmitJob",
    "Resource": "arn:aws:batch:*:*:job-definition/*:*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/Project": "calc"
      }
    }
  }
]
}

```

Recurso: política AWS Batch gerenciada

AWS Batch fornece uma política gerenciada que você pode anexar aos usuários. Essa política fornece permissão para usar AWS Batch recursos e operações de API. Você pode aplicar essa política diretamente ou usá-la como ponto de partida para criar suas próprias políticas. Para mais informações sobre cada uma das operações API mencionadas nessas políticas, consulte [Ações](#) em Referência API AWS Batch .

AWSBatchFullAccess

Essa política permite acesso total do administrador AWS Batch a.

Para ver o JSON da política, consulte o [Guia AWSBatchFullAccessde referência de políticas AWS gerenciadas](#).

AWS Batch Função de execução do IAM

A função de execução concede ao contêiner e aos AWS Fargate agentes do Amazon ECS permissão para fazer chamadas de AWS API em seu nome.

Note

O papel de execução é compatível pela versão do atendente 1.16.0 do atendente de contêiner do Amazon ECS e superior.

É necessário informar o perfil de execução do IAM dependendo dos requisitos da sua tarefa. Você pode ter vários papéis de execução para diferentes objetivos e associações de serviço em sua conta.

Note

Para obter mais informações sobre a função da instância do Amazon ECS, consulte [Perfil de instância do Amazon ECS](#). Para obter informações sobre perfis de serviço, consulte [Como AWS Batch funciona com o IAM](#).

O Amazon ECS fornece a política gerenciada AmazonECSTaskExecutionRolePolicy. Esta política contém as permissões obrigatórias para os casos de uso comuns descritos acima. Talvez seja necessário adicionar políticas em linha ao seu perfil de execução de trabalhos para os casos de uso especiais resumidos abaixo.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Permissões em nível de recurso suportadas para ações de API AWS Batch

O termo permissões em nível de recurso se refere à capacidade de especificar os recursos nos quais os usuários podem realizar ações. AWS Batch tem suporte parcial para permissões em nível de recurso. Para algumas AWS Batch ações, você pode controlar quando os usuários podem usar essas ações com base nas condições que devem ser atendidas. Você também pode controlar com base nos recursos específicos que os usuários podem usar. Por exemplo, você pode conceder permissões aos usuários para enviar tarefas, mas somente para uma determinada fila de tarefas e apenas com uma determinada definição de tarefa.

Para obter detalhes sobre ações e tipos de recursos definidos pelo AWS Batch, incluindo o formato do ARNs para cada um dos tipos de recursos, consulte [Actions, Resources, and Condition Keys for AWS Batch](#) the Service Authorization Reference.

Tutorial: criar o perfil de execução do IAM

Se a sua conta ainda não tiver um perfil de execução do IAM, use as etapas a seguir para criar o perfil.

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Perfis.
3. Escolha Criar Perfil.
4. Em Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).
5. Em Serviço ou Caso de uso, escolha Elastic Container Service. Então escolha Tarefa de serviço do Elastic Container novamente.
6. Escolha Próximo.
7. Para políticas de permissões, pesquise por Amazon ECSTask ExecutionRolePolicy.
8. Escolha a caixa de seleção à esquerda da ECSTask ExecutionRolePolicy política da Amazon e, em seguida, escolha Avançar.
9. Em Nome do perfil, insira `ecsTaskExecutionRole` e então, escolha Criar perfil.

Tutorial: verificar a perfil de execução do IAM

Use o procedimento a seguir para verificar se a sua conta já tem o perfil de execução do IAM e anexar a política gerenciada do IAM, caso necessário.

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Perfis.
3. Pesquise `ecsTaskExecutionRole` na lista de perfis. Se você não conseguir encontrar a função, consulte [Tutorial: criar o perfil de execução do IAM](#). Se a função existir, escolha-a para visualizar as políticas anexadas.
4. Na guia Permissões, verifique se a política `ECSTask ExecutionRolePolicy` gerenciada da Amazon está vinculada à função. Se a política estiver anexada, isso significa que seu perfil de execução está configurado corretamente. Caso contrário, siga as etapas secundárias abaixo para anexar a política.
 - a. Escolha Adicionar Permissões e depois escolha Anexar Políticas.
 - b. Pesquise pela Amazon `ECSTask ExecutionRolePolicy`.
 - c. Marque a caixa à esquerda da `ECSTask ExecutionRolePolicy` política da Amazon e escolha Anexar políticas.
5. Escolha Relações de Confiança.
6. Verifique se a relação de confiança contém a seguinte política. Se a relação de confiança corresponder à política abaixo, a função será configurada corretamente. Se a relação de confiança não corresponder, escolha Editar Política de Confiança, insira o seguinte e escolha Atualizar Política.

JSON

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

```
}  
]  
}
```

Usar perfis vinculados a serviço do AWS Batch

AWS Batch usa funções [vinculadas ao serviço AWS Identity and Access Management \(IAM\)](#). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS Batch As funções vinculadas ao serviço são predefinidas AWS Batch e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

AWS Batch usa duas funções vinculadas a serviços diferentes:

- [AWSServiceRoleForBatch](#)- Para AWS Batch operações, incluindo ambientes computacionais.
- [AWSServiceRoleForAWSBatchWithSageMaker](#)- Para gerenciamento de cargas de trabalho e filas de SageMaker IA.

Tópicos

- [Usando funções para AWS Batch](#)
- [Usando funções para AWS Batch com SageMaker IA](#)

Usando funções para AWS Batch

AWS Batch usa funções [vinculadas ao serviço AWS Identity and Access Management \(IAM\)](#). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS Batch As funções vinculadas ao serviço são predefinidas AWS Batch e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração AWS Batch porque você não precisa adicionar manualmente as permissões necessárias. AWS Batch define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só AWS Batch pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

 Note

Siga um destes procedimentos para especificar uma função de serviço para um ambiente AWS Batch computacional.

- Use uma string vazia para o perfil de serviço. Isso permite AWS Batch criar a função de serviço.
- Especifique o perfil de serviço no seguinte formato:
`arn:aws:iam::account_number:role/aws-service-role/
batch.amazonaws.com/AWSServiceRoleForBatch.`

Para obter mais informações, consulte [Nome do perfil ou ARN incorreto](#) o Guia AWS Batch do usuário.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus AWS Batch recursos porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS Serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões de função vinculadas ao serviço para AWS Batch

AWS Batch usa a função vinculada ao serviço chamada `AWSServiceRoleForBatch`— Permite AWS Batch criar e gerenciar AWS recursos em seu nome.

A função `AWSService RoleForBatch` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `batch.amazonaws.com`

A política de permissões de função nomeada [BatchServiceRolePolicy](#) AWS Batch permite concluir as seguintes ações nos recursos especificados:

- **autoscaling**— Permite AWS Batch criar e gerenciar recursos do Amazon EC2 Auto Scaling. AWS Batch cria e gerencia grupos do Amazon EC2 Auto Scaling para a maioria dos ambientes computacionais.
- **ec2**— Permite AWS Batch controlar o ciclo de vida das instâncias do Amazon EC2, bem como criar e gerenciar modelos e tags de lançamento. AWS Batch cria e gerencia solicitações do EC2 Spot Fleet para alguns ambientes computacionais EC2 Spot.
- **ecs**- Permite AWS Batch criar e gerenciar clusters, definições de tarefas e tarefas do Amazon ECS para execução de trabalhos.
- **eks**- Permite AWS Batch descrever o recurso de cluster Amazon EKS para validações.
- **iam**- Permite AWS Batch validar e passar funções fornecidas pelo proprietário para o Amazon EC2, o Amazon EC2 Auto Scaling e o Amazon ECS.
- **logs**— Permite AWS Batch criar e gerenciar grupos de registros e fluxos de registros para AWS Batch trabalhos.

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para ter mais informações, consulte [Permissões de perfil vinculado a serviços](#) no Guia do usuário do IAM.

Criar um perfil vinculado ao serviço para o AWS Batch

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você cria um ambiente computacional na Console de gerenciamento da AWS, na ou na AWS API AWS CLI, AWS Batch cria a função vinculada ao serviço para você.

 Important

Esse perfil vinculado ao serviço pode aparecer em sua conta se você concluiu uma ação em outro serviço que usa os atributos compatíveis com esse perfil. Se você estava usando o AWS Batch serviço antes de 10 de março de 2021, quando ele começou a oferecer suporte a funções vinculadas ao serviço, AWS Batch criou a AWSService RoleForBatch função em sua conta. Para saber mais, consulte [Uma nova função apareceu no meu Conta da AWS](#).

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um ambiente computacional, AWS Batch cria a função vinculada ao serviço para você novamente.

Editar um perfil vinculado ao serviço para o AWS Batch

AWS Batch não permite que você edite a função `AWSService RoleForBatch` vinculada ao serviço. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para saber mais, consulte [Editar um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

Para permitir que uma entidade do IAM edite a descrição da função `AWSService RoleForBatch` vinculada ao serviço

Adicione a seguinte instrução a política de permissões. Isso permite que uma entidade do IAM edite a descrição de uma função vinculada ao serviço.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:UpdateRoleDescription"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Excluir um perfil vinculado ao serviço para o AWS Batch

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não terá uma entidade não utilizada que não seja ativamente monitorada ou mantida. No entanto, você deve limpar seu perfil vinculado ao serviço para excluí-la manualmente.

Para permitir que uma entidade do IAM exclua a função `AWSService RoleForBatch` vinculada ao serviço

Adicione a seguinte instrução a política de permissões. Isso permite que a entidade do IAM delete a função vinculada ao serviço.

```
{
  "Effect": "Allow",
  "Action": [
    "iam:DeleteServiceLinkedRole",
  ]
}
```



```
    "iam:GetServiceLinkedRoleDeletionStatus"
  ],
  "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
  "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, você deve primeiro confirmar se a função não tem sessões ativas e excluir todos os ambientes AWS Batch computacionais que usam a função em todas as AWS regiões em uma única partição.

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 Funções e, em seguida, o AWSService RoleForBatch nome (não a caixa de seleção).
3. Na página Resumo, escolha Consultor de Acesso e analise as atividades recentes para a função vinculada ao serviço.

Note

Se você não sabe se AWS Batch está usando a AWSService RoleForBatch função, tente excluir a função. Se o serviço estiver usando a função, haverá falha ao excluir a função. Você pode visualizar as regiões nas quais a função estiver sendo usada. Se a função estiver sendo usada, você deve aguardar a sessão final antes de excluir a função. Você não pode revogar a sessão de uma função vinculada ao serviço.

Para remover AWS Batch recursos usados pela função vinculada ao AWSService RoleForBatch serviço

Você deve excluir todos os ambientes AWS Batch computacionais que usam a AWSService RoleForBatch função em todas as AWS regiões antes de excluir a AWSService RoleForBatch função.

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione a Região a ser usada.

3. No painel de navegação, escolha Ambientes de computação.
4. Selecione o ambiente de computação.
5. Escolha Desabilitar. Espere até que o Estado mude para DESATIVADO.
6. Selecione o ambiente de computação.
7. Escolha Deletar. Confirme que você deseja excluir o ambiente de computação escolhendo Excluir o ambiente computacional.
8. Repita as etapas 1 a 7 para todos os ambientes de computação usando a função vinculada ao serviço em todas as regiões.

Excluir uma função vinculada ao serviço no IAM (Console)

Você pode 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 do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Em seguida, marque a caixa de seleção ao lado AWSServiceRoleForBatch, não o nome ou a linha em si.
3. Escolha Excluir perfil.
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço mostrando quando cada uma das perfis selecionadas acessou pela última vez um AWS service (Serviço da AWS). Isso ajuda você a confirmar se a função está ativo no momento. Se quiser prosseguir, escolha Sim, Excluir para enviar a função vinculada ao serviço para exclusão.
5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão do perfil vinculado ao serviço do IAM é assíncrona, depois de enviar a função para exclusão, a exclusão da tarefa pode obter êxito ser reprovada.
 - 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 obtiver êxito, você poderá escolher Visualizar Detalhes ou Visualizar Recursos a partir das notificações para saber por que a exclusão falhou. Se a exclusão falhou porque a função está usando os recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

 Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles, ou pode não retornar relatórios de nenhum dos recursos.

- Se a tarefa falhar e a notificação não incluir uma lista de recursos, o serviço pode não retornar essas informações. Para saber como limpar os recursos para esse serviço, consulte [Serviços da AWS que Funcionam com o IAM](#). Descubra o serviço na tabela e escolha o link Sim para visualizar a documentação da função vinculada ao serviço.

Excluir uma função vinculada ao serviço no IAM (AWS CLI)

Você pode usar os comandos do IAM do 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 pode ser excluída se estiver sendo usada ou tiver 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 status da tarefa de exclusão. Insira o seguinte comando para enviar uma solicitação de exclusão de função vinculada ao serviço:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

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 retornará o motivo de falha para que você possa acionar a solução de problemas. Se a exclusão falhar porque a função estiver usando os

recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

 Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles. Ou talvez não relate qualquer recurso. Saiba como limpar os recursos de um serviço que não reporta qualquer recurso consultando [Serviços da AWS compatíveis com o IAM](#). Descubra o seu serviço na tabela e escolha o link Sim para ver a documentação da função vinculada ao serviço.

Excluir uma função vinculada ao serviço no IAM (API da AWS)

Você pode usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de um roll vinculada ao serviço, chame [DeleteServiceLinkedRole](#). Na solicitação, especifique o nome da AWSService RoleForBatch função.

Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o DeletionTaskId da resposta para verificar o status 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 retornará o motivo de falha para que você possa acionar a solução de problemas. Se a exclusão falhar porque a função estiver usando os recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

 Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles, ou pode não retornar relatórios de nenhum dos recursos. Saiba como limpar os recursos de um serviço que não relata qualquer recurso consultando [Serviços da AWS que funcionam com o IAM](#). Descubra o seu serviço na tabela e escolha o link Sim para ver a documentação da função vinculada ao serviço.

Regiões suportadas por perfis vinculados a serviço do AWS Batch

AWS Batch suporta o uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Endpoints do AWS Batch](#).

Usando funções para AWS Batch com SageMaker IA

AWS Batch usa funções [vinculadas ao serviço AWS Identity and Access Management](#) (IAM). Uma função vinculada ao serviço é um tipo exclusivo de função do IAM vinculada diretamente a. AWS Batch As funções vinculadas ao serviço são predefinidas AWS Batch e incluem todas as permissões que o serviço exige para chamar outros AWS serviços em seu nome.

Uma função vinculada ao serviço facilita a configuração AWS Batch porque você não precisa adicionar manualmente as permissões necessárias. AWS Batch define as permissões de suas funções vinculadas ao serviço e, a menos que seja definido de outra forma, só AWS Batch pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões, que não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus AWS Batch recursos porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços que oferecem suporte a funções vinculadas a serviços, consulte [AWS Serviços que funcionam com IAM](#) e procure os serviços que têm Sim na coluna

Funções vinculadas ao serviço. Escolha um Sim com um link para visualizar a documentação do perfil vinculado a esse serviço.

Permissões de função vinculadas ao serviço para AWS Batch

AWS Batch usa a função vinculada ao serviço chamada `AWSServiceRoleForAWSBatchWithSagemaker`— Permite AWS Batch enfileirar e gerenciar trabalhos de SageMaker treinamento em seu nome.

A função `AWSService RoleFor AWSBatch WithSagemaker` vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `sagemaker-queuing.batch.amazonaws.com`

A política de permissões de função AWS Batch permite concluir as seguintes ações nos recursos especificados:

- `sagemaker`— Permite AWS Batch gerenciar trabalhos de SageMaker treinamento, transformar trabalhos e outros recursos de SageMaker IA.
- `iam:PassRole`— Permite AWS Batch passar funções de execução definidas pelo cliente para a SageMaker IA para execução de tarefas. A restrição de recursos permite passar funções para serviços de SageMaker IA.

Você deve configurar permissões para permitir que seus usuários, grupos ou perfis criem, editem ou excluam um perfil vinculado ao serviço. Para ter mais informações, consulte [Permissões de perfil vinculado a serviços](#) no Guia do usuário do IAM.

Criar um perfil vinculado ao serviço para o AWS Batch

Não é necessário criar manualmente uma função vinculada ao serviço. Quando você cria um ambiente de serviço usando `CreateServiceEnvironment` na Console de gerenciamento da AWS, na ou na AWS API AWS CLI, AWS Batch cria a função vinculada ao serviço para você.

Se excluir esse perfil vinculado ao serviço e precisar criá-lo novamente, será possível usar esse mesmo processo para recriar o perfil em sua conta. Quando você cria um ambiente de serviço usando `CreateServiceEnvironment`, AWS Batch cria a função vinculada ao serviço para você novamente.

Para ver o JSON da política, consulte o [Guia AWSBatchServiceRolePolicyForSageMakerde referência de políticas AWS gerenciadas](#).

Editar um perfil vinculado ao serviço para o AWS Batch

AWS Batch não permite que você edite a função AWSService RoleFor AWSBatch WithSagemaker vinculada ao serviço. Depois que criar um perfil vinculado ao serviço, você não poderá alterar o nome do perfil, pois várias entidades podem fazer referência a ele. No entanto, será possível editar a descrição do perfil usando o IAM. Para obter mais informações, consulte [Editar uma função vinculada a serviço](#) no Guia do usuário do IAM.

Excluir um perfil vinculado ao serviço para o AWS Batch

Se você não precisar mais usar um atributo ou serviço que requer um perfil vinculado ao serviço, é recomendável excluí-lo. Dessa forma, você não terá uma entidade não utilizada que não seja ativamente monitorada ou mantida. No entanto, você deve limpar seu perfil vinculado ao serviço para excluí-la manualmente.

Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir uma função vinculada ao serviço, você deve primeiro confirmar se a função não tem sessões ativas e excluir todos os ambientes de serviço que usam a função em todas as AWS regiões em uma única partição.

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 Funções e, em seguida, o AWSService RoleFor AWSBatch WithSagemaker nome (não a caixa de seleção).
3. Na página Resumo, escolha Consultor de Acesso e analise as atividades recentes para a função vinculada ao serviço.

Note

Se você não sabe se AWS Batch está usando a AWSService RoleFor AWSBatch WithSagemaker função, você pode tentar excluir a função. Se o serviço estiver usando a função, haverá falha ao excluir a função. Você pode visualizar as regiões nas quais a função estiver sendo usada. Se a função estiver sendo usada, você deve aguardar a

sessão final antes de excluir a função. Você não pode revogar a sessão de uma função vinculada ao serviço.

Para remover AWS Batch recursos usados pela função vinculada ao AWSService RoleFor AWSBatch WithSagemaker serviço

Você deve dissociar todas as filas de trabalhos de todos os ambientes de serviço e, em seguida, excluir todos os ambientes de serviço que usam a AWSService RoleFor AWSBatch WithSagemaker função em todas as AWS regiões antes de excluir a função. AWSService RoleFor AWSBatch WithSagemaker

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione a Região a ser usada.
3. No painel de navegação, escolha Ambientes e, então, Ambientes de serviço.
4. Selecione todos os Ambientes de serviço.
5. Escolha Desabilitar. Espere até que o Estado mude para DESATIVADO.
6. Selecione o ambiente de serviço.
7. Escolha Excluir. Confirme que você deseja excluir o ambiente de serviço escolhendo Excluir o ambiente de serviço.
8. Repita as etapas 1 a 7 para todos os ambientes de serviço usando o perfil vinculado ao serviço em todas as regiões.

Excluir uma função vinculada ao serviço no IAM (Console)

Você pode 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 do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação do console do IAM, escolha Perfis. Em seguida, marque a caixa de seleção ao lado AWSService RoleFor AWSBatchWithSagemaker, não o nome ou a linha em si.
3. Escolha Excluir perfil.
4. Na caixa de diálogo de confirmação, revise os dados do último acesso ao serviço mostrando quando cada uma das perfis selecionadas acessou pela última vez um AWS service (Serviço

da AWS). Isso ajuda você a confirmar se a função está ativo no momento. Se quiser prosseguir, escolha Sim, Excluir para enviar a função vinculada ao serviço para exclusão.

5. Monitore as notificações do console do IAM para progresso da exclusão da função vinculada ao serviço. Como a exclusão do perfil vinculado ao serviço do IAM é assíncrona, depois de enviar a função para exclusão, a exclusão da tarefa pode obter êxito ser reprovada.
 - 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 obtiver êxito, você poderá escolher Visualizar Detalhes ou Visualizar Recursos a partir das notificações para saber por que a exclusão falhou. Se a exclusão falhou porque a função está usando os recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

 Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles, ou pode não retornar relatórios de nenhum dos recursos.

- Se a tarefa falhar e a notificação não incluir uma lista de recursos, o serviço pode não retornar essas informações. Para saber como limpar os recursos para esse serviço, consulte [Serviços da AWS que Funcionam com o IAM](#). Descubra o serviço na tabela e escolha o link Sim para visualizar a documentação da função vinculada ao serviço.

Excluir uma função vinculada ao serviço no IAM (AWS CLI)

Você pode usar os comandos do IAM do 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 pode ser excluída se estiver sendo usada ou tiver 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 status da tarefa de exclusão. Insira o seguinte comando para enviar uma solicitação de exclusão de função vinculada ao serviço:

```
$ aws iam delete-service-linked-role --role-name  
AWSServiceRoleForAWSBatchWithSagemaker
```

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 retornará o motivo de falha para que você possa acionar a solução de problemas. Se a exclusão falhar porque a função estiver usando os recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles. Ou talvez não relate qualquer recurso. Saiba como limpar os recursos de um serviço que não reporta qualquer recurso consultando [Serviços da AWS compatíveis com o IAM](#). Descubra o seu serviço na tabela e escolha o link Sim para ver a documentação da função vinculada ao serviço.

Excluir uma função vinculada ao serviço no IAM (API da AWS)

Você pode usar a API do IAM para excluir uma função vinculada ao serviço.

Para excluir uma função vinculada ao serviço (API)

1. Para enviar uma solicitação de exclusão de um roll vinculada ao serviço, chame [DeleteServiceLinkedRole](#). Na solicitação, especifique o nome da AWSService RoleFor AWSBatch WithSagemaker função.

Como uma função vinculada ao serviço não podem ser excluída se estiver sendo usada ou tiver recursos associados, você deverá enviar uma solicitação de exclusão. Essa solicitação poderá ser negada se essas condições não forem atendidas. Você deve capturar o `DeletionTaskId` da resposta para verificar o status 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 retornará o motivo de falha para que você possa acionar a solução de problemas. Se a exclusão falhar porque a função estiver usando os recursos do serviço, a notificação incluirá uma lista de recursos, caso o serviço retorne essas informações. Você poderá então [limpar os recursos](#) e enviar novamente a exclusão.

Note

Você pode repetir esse processo várias vezes, a depender das informações que o serviço retornar. 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 exclusão falhará e o serviço emitirá relatório sobre o recurso restante. Um serviço pode retornar todos os recursos, alguns deles, ou pode não retornar relatórios de nenhum dos recursos. Saiba como limpar os recursos de um serviço que não relata qualquer recurso consultando [Serviços da AWS que funcionam com o IAM](#). Descubra o seu serviço na tabela e escolha o link Sim para ver a documentação da função vinculada ao serviço.

Regiões suportadas por perfis vinculados a serviço do AWS Batch

AWS Batch suporta o uso de funções vinculadas ao serviço em todas as regiões em que o serviço está disponível. Para obter mais informações, consulte [Endpoints do AWS Batch](#).

Perfil de instância do Amazon ECS

AWS Batch os ambientes computacionais são preenchidos com instâncias de contêiner do Amazon ECS. Eles executam o atendente de contêiner do Amazon ECS localmente. O agente de contêineres do Amazon ECS faz chamadas para várias operações de AWS API em seu nome. Portanto, as instâncias de contêiner que executam o atendente exigem uma política e um perfil do IAM para

esses serviços para reconhecerem que o atendente pertence a você. Você deve criar um perfil do IAM e um perfil de instância para as instâncias de contêiner poderem usar quando iniciadas. Do contrário, você não poderá criar um ambiente de computação e iniciar instâncias de contêiner nele. Esse requisito se aplica a instâncias de contêiner iniciadas com ou sem o AMI otimizado Amazon ECS, fornecido pela Amazon. Para obter mais informações, consulte [Perfil de instância do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Tópicos

- [Verificar o perfil de instância do Amazon ECS de sua conta](#)

Verificar o perfil de instância do Amazon ECS de sua conta

A função e o perfil de instância Amazon ECS são criados automaticamente para você durante a primeira execução do console. No entanto, é possível seguir essas etapas para verificar se a sua conta já possui a função e perfil de instância do Amazon ECS. As etapas a seguir também abordam como anexar a política do IAM gerenciada.

Tutorial: verificar **ecsInstanceRole** no campo de console do IAM

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, escolha Perfis.
3. Procure na lista de perfis por `ecsInstanceRole`. Caso o perfil não exista, use os seguintes passos para criá-lo.
 - a. Escolha Criar Perfil.
 - b. Em Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).
 - c. Em Ocasões de Uso Comuns, escolha EC2.
 - d. Escolha Próximo.
 - e. Para políticas de permissões, pesquise Amazon EC2 ContainerServicefor EC2 Role.
 - f. Escolha a caixa de seleção ao lado de Amazon EC2 ContainerServicefor EC2 Role e, em seguida, escolha Avançar.
 - g. Em Nome do Perfil, digite `ecsInstanceRole` e selecione Criar Perfil.

 Note

Se você usar o Console de gerenciamento da AWS para criar uma função para o Amazon EC2, o console cria um perfil de instância com o mesmo nome da função.

Como alternativa, você pode usar o AWS CLI para criar a função `ecsInstanceRole` do IAM. O exemplo a seguir cria uma função do IAM com uma política de confiança e uma política AWS gerenciada.

Tutorial: criar um perfil do IAM e um de instância (AWS CLI)

1. Crie a seguinte política de confiança e salve-a em um arquivo de texto chamado `ecsInstanceRole-role-trust-policy.json`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com" },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Use o comando [create-role](#) para criar o perfil `ecsInstanceRole`. Especifique a localização do arquivo da política de confiança no parâmetro `assume-role-policy-document`.

```
$ aws iam create-role \
  --role-name ecsInstanceRole \
  --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json
```

3. Use o [create-instance-profile](#) comando para criar um perfil de instância chamado `ecsInstanceRole`.

 Note

Você precisa criar funções e perfis de instância como ações separadas na AWS API AWS CLI e.

```
$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

O seguinte é um exemplo de resposta.

```
{
  "InstanceProfile": {
    "Path": "/",
    "InstanceProfileName": "ecsInstanceRole",
    "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
    "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
    "CreateDate": "2022-06-30T23:53:34.093Z",
    "Roles": [],
  }
}
```

4. Use o comando [add-role-to-instance-profile](#) para adicionar a ecsInstanceRole função ao perfil da ecsInstanceRole instância.

```
aws iam add-role-to-instance-profile \
  --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Use o [attach-role-policy](#) comando para anexar a política AmazonEC2ContainerServiceforEC2Role AWS gerenciada à ecsInstanceRole função.

```
$ aws iam attach-role-policy \
  --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
  --role-name ecsInstanceRole
```

Perfil de frota spot Amazon EC2

Caso crie um ambiente de computação gerenciado que faça uso de instâncias de frota spot Amazon EC2, você deve criar a política de AmazonEC2SpotFleetTaggingRole. Esta política concede à

frota spot permissão para iniciar, marcar e encerrar instâncias em seu nome. Especifique a função em sua solicitação de frota Spot. Você também deve ter as funções `AWSServiceRoleForEC2spot` e `AWSServiceRoleForEC2SpotFleet` vinculadas ao serviço para Amazon EC2 Spot e Spot Fleet. Use as instruções a seguir para criar estes perfis. Para obter mais informações, consulte [Como usar funções vinculadas ao serviço](#) e [Criar uma função para delegar permissões a um AWS serviço](#) no Guia do usuário do IAM.

Tópicos

- [Crie perfis de frota spot Amazon EC2 no Console de gerenciamento da AWS](#)
- [Crie funções de frota spot do Amazon EC2 com o AWS CLI](#)

Crie perfis de frota spot Amazon EC2 no Console de gerenciamento da AWS

Para criar o **AmazonEC2SpotFleetTaggingRole** perfil do IAM vinculado ao serviço para frota spot Amazon EC2

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. Em Gerenciamento de Acesso, escolha Perfis,
3. Em Perfis, escolha Criar Perfil.
4. De Selecionar Entidade Confiável para Tipo de Entidade Confiável, escolha AWS service (Serviço da AWS).
5. Para casos de uso para outros Serviços da AWS, escolha EC2 e, em seguida, escolha EC2 - Spot Fleet Tagging.
6. Escolha Próximo.
7. De Políticas de permissões para Nome da Política, verifique `AmazonEC2SpotFleetTaggingRole`.
8. Escolha Próximo.
9. Em Nomear, Revisar e Criar:
 - a. Em Nome do Perfil, insira um nome para identificar o perfil.
 - b. Em Descrição, insira uma breve explicação para a política.
 - c. (Opcional) Para Passo 1: Selecionar Entidades Confiáveis, escolha Editar para modificar o código.
 - d. (Opcional) Para Passo 2: Adicionar Permissões, escolha Editar para modificar o código.

- e. (Opcional) Em Adicionar Marcações, escolha Adicionar Marcação para adicionar marcações ao recurso.
- f. Escolha Criar Perfil.

Note

No passado, havia duas políticas gerenciadas para o perfil de frota spot Amazon EC2.

- Amazon EC2 SpotFleetRole: essa é a política gerenciada original para a função Spot Fleet. No entanto, não recomendamos mais que você o use com AWS Batch. Essa política não oferece suporte à marcação de Instância Spot em ambientes de computação, necessário para utilização do perfil vinculado ao serviço `AWSServiceRoleForBatch`. Caso já tenha criado um perfil de frota spot com essa política, aplique a nova política recomendada ao perfil. Para obter mais informações, consulte [Instâncias spot sem tags na criação](#).
- Amazon EC2 SpotFleetTaggingRole: essa função fornece todas as permissões necessárias para marcar instâncias spot do Amazon EC2. Use essa função para permitir a marcação de Instâncias Spot em seus ambientes AWS Batch computacionais.

Crie funções de frota spot do Amazon EC2 com o AWS CLI

Para criar a função do Amazon EC2 SpotFleetTaggingRole IAM para seus ambientes computacionais Spot Fleet

1. Execute o seguinte comando com o AWS CLI.

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
  --assume-role-policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "spotfleet.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```



```
}'
```

2. Para anexar a política de IAM EC2 SpotFleetTaggingRole gerenciada pela Amazon à sua EC2 SpotFleetTaggingRole função na Amazon, execute o seguinte comando com AWS CLI o.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Para criar um perfil do IAM **AWSServiceRoleForEC2Spot** vinculado ao serviço Amazon EC2 Spot

 Note

Caso o perfil do IAM **AWSServiceRoleForEC2Spot** vinculado ao serviço já exista, você verá uma mensagem de erro semelhante à seguinte.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation:
Service role name AWSServiceRoleForEC2Spot has been taken in this account,
please try a different suffix.
```

- Execute o seguinte comando com o AWS CLI.

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

Para criar o perfil do IAM **AWSServiceRoleForEC2SpotFleet** vinculado ao serviço para frota spot Amazon EC2

 Note

Caso o perfil do IAM **AWSServiceRoleForEC2SpotFleet** vinculado ao serviço já exista, você verá uma mensagem de erro semelhante à seguinte.

```
An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
operation:
```

Service role name `AWSServiceRoleForEC2SpotFleet` has been taken in this account, please try a different suffix.

- Execute o seguinte comando com o AWS CLI.

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

EventBridge IAM role (Função do IAM)

A Amazon EventBridge fornece um fluxo quase em tempo real de eventos do sistema que descrevem mudanças nos AWS recursos. AWS Batch empregos estão disponíveis como EventBridge alvos. Por meio de regras simples e rapidamente configuráveis, você pode equivaler eventos e submeter trabalhos AWS Batch em resposta. Antes de enviar AWS Batch trabalhos com EventBridge regras e metas, você EventBridge deve ter permissões para executar AWS Batch trabalhos em seu nome.

Note

Ao criar uma regra no EventBridge console que especifica uma AWS Batch fila como destino, você pode criar essa função. Para ver uma demonstração de exemplo, consulte [Trabalhos AWS Batch como destinos do EventBridge](#). Você pode criar a EventBridge função manualmente usando o console do IAM. Para instruções, consulte [Criando um Perfil Usando Políticas de Confiança Customizadas \(Console\)](#) no Guia de Usuário do IAM.

A relação de confiança da sua função EventBridge do IAM deve fornecer ao diretor do `events.amazonaws.com` serviço a capacidade de assumir a função.

JSON

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

```
    "Principal": {
      "Service": "events.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

Certifique-se de que a política anexada à sua função EventBridge do IAM permita `batch:SubmitJob` permissões em seus recursos. No exemplo a seguir, o AWS Batch fornece a política gerenciada `AWSBatchServiceEventTargetRole` para essas permissões.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": "*"
    }
  ]
}
```

Criar uma nuvem privada virtual

Os recursos de computação dos seus ambientes de computação demandam acesso a rede externa para se comunicarem com o AWS Batch e com endpoints de serviço Amazon ECS. No entanto, você pode ter tarefas que deseje executar em sub redes privadas. A criação de uma VPC com sub redes públicas e privadas concede flexibilidade para executar tarefas em uma sub rede pública ou privada.

Você pode usar a Amazon Virtual Private Cloud (Amazon VPC) para lançar AWS recursos em uma rede virtual que você define. Este tópico fornece um link para o assistente Amazon VPC e uma lista de opções para seleção.

Crie uma VPC

Para informações sobre como criar uma Amazon VPC, consulte [Criar uma Única VPC](#) no Guia de Usuário Amazon VPC e use a tabela a seguir para determinar quais opções selecionar.

Opção	Valor	
Recursos a criar	Somente VPC	
Nome	Forneça um nome opcional para a sua VPC.	
IPv4 Bloco CIDR	IPv4 Entrada manual CIDR O tamanho do bloco CIDR deve ter um tamanho entre /16 e /28.	
IPv6 Bloco CIDR	Sem IPv6 bloco CIDR	
Localção	Padrão	

Para saber mais sobre a Amazon VPC, consulte [O Que é Amazon VPC?](#) no Guia de usuário Amazon VPC.

Próximas etapas

Depois de criar sua VPC, considere os seguintes próximos passos:

- Crie grupos de segurança para seus recursos públicos e privados, caso precisem de acesso à rede de entrada. Para mais informações, consulte [Trabalhe com Grupos de Segurança](#) no Guia de Usuário Amazon VPC.
- Crie um ambiente de computação AWS Batch gerenciado que lance recursos computacionais em sua nova VPC. Para obter mais informações, consulte [Crie um ambiente de computação](#). Se você usar o assistente de criação de ambiente computacional no AWS Batch console, poderá especificar a VPC que acabou de criar e as sub-redes públicas ou privadas nas quais deseja iniciar suas instâncias.

- Crie uma fila de AWS Batch trabalhos mapeada para seu novo ambiente computacional. Para obter mais informações, consulte [Crie uma fila de trabalhos](#).
- Crie uma definição de tarefa para executar seus trabalhos. Para obter mais informações, consulte [Criar uma definição de tarefa de nó único](#).
- Insira uma tarefa com sua definição de tarefa para a nova fila de tarefas. Este trabalho será posicionado no ambiente de computação criado com VPC e sub redes novas. Para obter mais informações, consulte [Tutorial: enviar um trabalho](#).

Use um endpoint de interface para acessar AWS Batch

Você pode usar AWS PrivateLink para criar uma conexão privada entre sua VPC e AWS Batch. Você pode acessar o AWS Batch como se estivesse em sua VPC, sem o uso de gateway da internet, dispositivo NAT, conexão VPN ou conexão do Direct Connect. As instâncias na sua VPC não precisam de endereços IP públicos para acessar o AWS Batch.

Estabeleça essa conectividade privada criando um endpoint de interface, habilitado pelo AWS PrivateLink. Criaremos um endpoint de interface de rede em cada sub-rede que você habilitar para o endpoint de interface. Estas são interfaces de rede gerenciadas pelo solicitante que servem como ponto de entrada para o tráfego destinado ao AWS Batch.

Para obter mais informações, consulte, consulte [VPC Endpoints de Interface](#) no Guia AWS PrivateLink.

Considerações para AWS Batch

Antes de configurar um endpoint de interface para AWS Batch, revise as [propriedades e limitações do endpoint de interface](#) no AWS PrivateLink Guia.

AWS Batch suporta fazer chamadas para todas as suas ações de API por meio do endpoint da interface.

Antes de configurar a interface para VPC endpoints AWS Batch, esteja ciente das seguintes considerações:

- Trabalhos usando o tipo de lançamento de recursos Fargate não exigem a interface VPC endpoints para Amazon ECS, mas você pode precisar de endpoints VPC de interface para Amazon ECR, Secrets Manager ou AWS Batch Amazon Logs descritos nos pontos a seguir. CloudWatch

- Para executar trabalhos, você deve criar VPC endpoints de interface para o Amazon ECS. Para obter mais informações, consulte [VPC Endpoints de Interface \(AWS PrivateLink\)](#) no Guia do desenvolvedor do Amazon Elastic Container Service.
- Para permitir que seus trabalhos extraiam imagens privadas do Amazon ECR, você deve criar VPC endpoints de interface para o Amazon ECR. Para obter mais informações, consulte [VPC Endpoints de interface \(AWS PrivateLink\)](#) no Manual do Usuário do Amazon Elastic Container Registry.
- Para permitir que seus trabalhos extraiam dados sigilosos do Secrets Manager, é necessário criar VPC endpoints de interface para o Secrets Manager. Para obter mais informações, consulte [Usando o Secrets Manager com Endpoints da VPC](#) no Manual do Usuário do AWS Secrets Manager .
- Se sua VPC não tiver um gateway de internet e seus trabalhos usarem o driver de `awslogs` log para enviar informações de log para CloudWatch Logs, você deverá criar uma interface VPC endpoint para Logs. CloudWatch Para obter mais informações, consulte Como [usar CloudWatch registros com endpoints VPC de interface](#) no Guia do usuário do Amazon CloudWatch Logs.
- Os trabalhos que usam os requisitos dos recursos do EC2 exigem que as instâncias de contêiner nas quais forem iniciadas executem a versão 1.25.1 ou superior do atendente de contêiner do Amazon ECS. Para obter mais informações, consulte [Versões do atendente de Contêiner do Amazon ECS](#) no Guia do Desenvolvedor do Amazon Elastic Container Service.
- Atualmente, os endpoints da VPC não oferecem suporte a solicitações entre Regiões. Garanta a criação do seu endpoint na mesma Região onde planeja emitir as chamadas de API para o AWS Batch.
- Os endpoints da VPC oferecem suporte somente a DNS fornecidos pela Amazon por meio do Amazon Route 53. Se quiser usar seu próprio DNS, você pode usar o encaminhamento de DNS condicional. Para obter mais informações, consulte [Conjuntos de Opções de DHCP](#) no Manual do Usuário da Amazon VPC.
- O grupo de segurança anexado ao endpoint da VPC deve permitir entrada conectada a porta 443 na sub-rede privada da VPC.
- AWS Batch não oferece suporte a endpoints de interface VPC no seguinte: Regiões da AWS
 - Ásia-Pacífico (Osaka) (`ap-northeast-3`)
 - Ásia-Pacífico (Jacarta) (`ap-southeast-3`)

Crie um endpoint de interface para AWS Batch

Você pode criar um endpoint de interface para AWS Batch usar o console Amazon VPC ou AWS Command Line Interface o AWS CLI(). Para obter mais informações, consulte [Criar um endpoint de interface](#) no Guia do usuário do AWS PrivateLink .

Crie um endpoint de interface para AWS Batch usar os seguintes nomes de serviço:

- com.amazonaws. *region*.lote
- com.amazonaws. *region*.batch-fips (para endpoints compatíveis com FIPS, consulte endpoints e cotas)[AWS Batch](#)

Por exemplo:

```
com.amazonaws.us-east-2.batch
```

```
com.amazonaws.us-east-2.batch-fips
```

Na partição aws-cn, o formato é diferente:

```
cn.com.amazonaws.region.batch
```

Por exemplo:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Nomes DNS privados para endpoints de AWS Batch interface

Se você habilitar o DNS privado para o endpoint da interface, poderá usar nomes DNS específicos para se conectar. Fornecemos AWS Batch estas opções:

- lote. *region*.amazonaws.com
- lote. *region*.api.aws

Para terminais compatíveis com FIPS:

- dicas em lote. *region*.api.aws

- fips.batch. *region*.amazonaws.com não é suportado

Para obter mais informações, consulte [Acessando um Serviço por meio de um Endpoint de Interface](#) no AWS PrivateLink Guia .

Criar uma política de endpoint para o endpoint de interface

Uma política de endpoint é um recurso do IAM que pode ser anexado ao endpoint de interface. A política de endpoint padrão permite acesso total AWS Batch por meio do endpoint da interface. Para controlar o acesso permitido ao AWS Batch pela VPC, anexe uma política de endpoint personalizada ao endpoint de interface.

Uma política de endpoint especifica as seguintes informações:

- As entidades principais que podem executar ações (Contas da AWS, usuários e perfis do IAM).
- As ações que podem ser realizadas.
- Os recursos nos quais as ações podem ser executadas.

Para obter mais informações, consulte [Controlar o acesso aos serviços usando políticas de endpoint](#) no Guia do AWS PrivateLink .

Exemplo: política de VPC endpoint para ações AWS Batch

Veja a seguir um exemplo de uma política de endpoint personalizado. Quando você anexa essa política ao seu endpoint de interface, ela concede acesso às AWS Batch ações listadas para todos os diretores em todos os recursos.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob",
        "batch:ListJobs",
        "batch:DescribeJobs"
      ],
      "Resource": "*"
    }
  ]
}
```



```
]
}
```

Validação de conformidade para AWS Batch

Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .

Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#) .

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a [Documentação AWS de segurança](#).

Segurança da infraestrutura em AWS Batch

Como serviço gerenciado, AWS Batch é protegido pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar AWS Batch pela rede. Os clientes devem oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

Você pode chamar essas operações de API de qualquer local de rede, mas AWS Batch oferece suporte a políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar AWS Batch políticas para controlar o acesso

de endpoints específicos da Amazon Virtual Private Cloud (Amazon VPC) ou específicos. VPCs Efetivamente, isso isola o acesso à rede a um determinado AWS Batch recurso somente da VPC específica dentro da AWS rede.

Prevenção do problema "confused deputy" entre serviços

O problema de adjunto confuso é um problema de segurança em que uma entidade que não tem permissão para executar uma ação pode coagir outra entidade mais privilegiada a executá-la. Em AWS, a falsificação de identidade entre serviços pode resultar em um problema confuso de delegado. A personificação entre serviços pode ocorrer quando um serviço (o serviço de chamada) chama outro serviço (o serviço chamado). O serviço de chamada pode ser manipulado de modo a usar suas permissões para atuar nos recursos de outro cliente de uma forma na qual ele não deveria ter permissão para acessar. Para evitar isso, a AWS fornece ferramentas que ajudam você a proteger seus dados para todos os serviços com entidades principais de serviço que receberam acesso aos recursos em sua conta.

Recomendamos usar as chaves de contexto de condição [aws:SourceAccount](#) global [aws:SourceArn](#) e as chaves de contexto nas políticas de recursos para limitar as permissões que AWS Batch concedem outro serviço ao recurso. Se o valor de `aws:SourceArn` não contém ID da conta, como um ARN do bucket do Amazon S3, você deve usar ambas as chaves de contexto de condição global para limitar as permissões. Se você usa ambas as chaves de contexto de condição global, e o valor `aws:SourceArn` contém o ID da conta, o valor `aws:SourceAccount` e a conta no valor `aws:SourceArn` deverão utilizar a mesma ID de conta quando na mesma declaração de política. Use `aws:SourceArn` se quiser apenas um recurso associado a acessibilidade de serviço. Use `aws:SourceAccount` se quiser permitir que qualquer recurso nessa conta seja associado ao uso entre serviços.

O valor de `aws:SourceArn` deve ser o recurso AWS Batch armazenado.

A maneira mais eficaz de se proteger contra o problema do substituto confuso é usar a chave de contexto de condição global `aws:SourceArn` com o ARN completo do recurso. Se você não souber o ARN completo do recurso ou especificar vários recursos, use a chave de condição de contexto global `aws:SourceArn` com caracteres curinga (*) para as partes desconhecidas do ARN. Por exemplo, `.arn:aws:service:*:123456789012:*`

Os exemplos a seguir mostram como você pode usar as chaves de contexto de condição `aws:SourceAccount` global `aws:SourceArn` e as chaves de contexto AWS Batch para evitar o confuso problema substituto.

Exemplo: perfil para acessar apenas um ambientes de computação

A função a seguir pode ser usada apenas para acessar um ambiente computacional. O nome do trabalho deve ser especificado como *****, porque a fila de trabalhos pode ser associada a vários ambientes de computação.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Exemplo: perfil para acessar vários ambientes de computação

A função a seguir pode ser usada apenas para acessar vários ambientes de computação. O nome do trabalho deve ser especificado como *****, porque a fila de trabalhos pode ser associada a vários ambientes de computação.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Registrando chamadas de AWS Batch API com AWS CloudTrail

AWS Batch é integrado com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço em AWS Batch. CloudTrail captura todas as chamadas de API AWS Batch como eventos. As chamadas capturadas incluem chamadas do AWS Batch console e chamadas de código para as operações AWS Batch da API. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para AWS Batch. Se você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita AWS Batch, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre isso CloudTrail, consulte o [Guia AWS CloudTrail do usuário](#).

Tópicos

- [AWS Batch informações em CloudTrail](#)
- [Referência: Entendendo as entradas do arquivo de AWS Batch log](#)

AWS Batch informações em CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre em AWS Batch, essa atividade é registrada em um CloudTrail evento junto com outros eventos AWS de serviço no histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte [Visualização de eventos com histórico de CloudTrail eventos](#).

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos para AWS Batch, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, a trilha se aplica a todas as AWS regiões. A trilha registra eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para saber mais, consulte:

- [Visão Geral para Criar uma Trilha](#)
- [CloudTrail Serviços e integrações compatíveis](#)
- [Configurando notificações do Amazon SNS para CloudTrail](#)
- [Recebendo arquivos de CloudTrail log de várias regiões](#) e [recebendo arquivos de CloudTrail log de várias contas](#)

Todas as AWS Batch ações são registradas CloudTrail e documentadas nas <https://docs.aws.amazon.com/batch/latest/APIReference/>. Por exemplo, as chamadas para as seções [SubmitJob](#), [ListJobs](#) e [DescribeJobs](#) 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-raiz ou usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.

- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte [Elemento userIdentity do CloudTrail](#).

Referência: Entendendo as entradas do arquivo de AWS Batch log

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e hora da ação, parâmetros de solicitação, e assim por diante. arquivos de log do CloudTrail não são um rastreamento de pilha ordenada das chamadas da API pública. Assim, elas não são exibidas em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a [CreateComputeEnvironment](#) ação.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2017-12-20T00:48:46Z",
  "eventSource": "batch.amazonaws.com",
  "eventName": "CreateComputeEnvironment",
```

```

"awsRegion": "us-east-1",
"sourceIPAddress": "203.0.113.1",
"userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
"requestParameters": {
  "computeResources": {
    "subnets": [
      "subnet-5eda8e04"
    ],
    "tags": {
      "testBatchTags": "CLI testing CE"
    },
    "desiredvCpus": 0,
    "minvCpus": 0,
    "instanceTypes": [
      "optimal"
    ],
    "securityGroupIds": [
      "sg-aba9e8db"
    ],
    "instanceRole": "ecsInstanceRole",
    "maxvCpus": 128,
    "type": "EC2"
  },
  "state": "ENABLED",
  "type": "MANAGED",
  "computeEnvironmentName": "Test"
},
"responseElements": {
  "computeEnvironmentName": "Test",
  "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/Test"
},
"requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
"eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
"readOnly": false,
"eventType": "AwsApiCall",
"recipientAccountId": "012345678910"
}

```

Solucionar problemas do IAM AWS Batch

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS Batch um IAM.

Tópicos

- [Não tenho autorização para executar uma ação no AWS Batch](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas fora da minha AWS conta acessem meus AWS Batch recursos](#)

Não tenho autorização para executar uma ação no AWS Batch

Se isso Console de gerenciamento da AWS indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. O administrador é a pessoa que forneceu o seu nome de usuário e senha.

O erro do exemplo a seguir ocorre quando o usuário `mateojackson` tenta usar o console para visualizar detalhes sobre um recurso do `my-example-widget` fictício, mas não tem as permissões fictícias do batch: `GetWidget`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
batch:GetWidget on resource: my-example-widget
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas e permitir o acesso ao recurso `my-example-widget` usando a ação batch: `GetWidget`. Para obter mais informações sobre como conceder permissões para passar uma função, consulte [Conceder permissões a um usuário para passar uma função para um AWS serviço](#).

Não estou autorizado a realizar iam: PassRole

Se você receber uma mensagem de erro informando que não está autorizado a executar a ação `iam:PassRole`, as suas políticas devem ser atualizadas para permitir que você passe uma função para o AWS Batch.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazê-lo, você deve ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando uma usuária do IAM chamada `marymajor` tenta utilizar o console para executar uma ação no AWS Batch. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.


```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

Quero permitir que pessoas fora da minha AWS conta acessem meus AWS Batch recursos

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. 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 aos seus recursos.

Para saber mais, consulte:

- Para saber se é AWS Batch compatível com esses recursos, consulte [Como AWS Batch funciona com o IAM](#).
- Para saber como fornecer acesso aos seus recursos em todas as Contas da AWS que você possui, consulte Como [fornecer acesso a um usuário do IAM em outra Conta da AWS que você possui](#) no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como [fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para conhecer a diferença entre perfis e políticas baseadas em recurso para acesso entre contas, consulte [Acesso a recursos entre contas no IAM](#) no Guia do usuário do IAM.

AWS Step Functions

Você pode usar o console do AWS Batch para visualizar detalhes sobre as máquinas de estado do Step Functions e as perfis usadas por elas.

Seções

- [Tutorial: visualizar detalhes da máquina de estado](#)
- [Tutorial: editar uma máquina de estado](#)
- [Tutorial: executar uma máquina de estado](#)

Tutorial: visualizar detalhes da máquina de estado

O console do AWS Batch exibe uma lista de máquinas de estado na região da Região da AWS atual que contêm pelo menos uma etapa de fluxo de trabalho que envia um trabalho do AWS Batch.

Escolha uma máquina de estado para visualizar uma representação gráfica do fluxo de trabalho. As etapas destacadas em azul representam trabalhos do AWS Batch. Use os controles do gráfico para ampliar e diminuir o zoom e para centralizar o gráfico.

Note

Quando um trabalho do AWS Batch é [referenciada dinamicamente com JsonPath](#) na definição da máquina de estado, os detalhes da função não podem ser exibidos no console do AWS Batch. Em vez disso, o nome do trabalho será listado como Dynamic reference (Referência dinâmica), e as etapas correspondentes no gráfico serão desabilitadas.

Como visualizar detalhes da máquina de estado

1. Abra a página de [orquestração de fluxo de trabalho com tecnologia Step Functions do console do AWS Batch](#).
2. Escolha uma máquina de estado.

<result>

O console do AWS Batch abrirá a página Details.

</result>

Para obter mais informações, consulte [Step Functions](#) no Developer GuideAWS Step Functions.

Tutorial: editar uma máquina de estado

Se desejar editar uma máquina de estado, o AWS Batch abre a página Edit definition (Editar definição) do console do Step Functions.

Como editar uma máquina de estado

1. Abra a página de [orquestração de fluxo de trabalho com tecnologia Step Functions do console do AWS Batch](#).
2. Escolha uma máquina de estado.
3. Escolha Edit.

O console do Step Functions abrirá a página Edit definition.

4. Edite a máquina de estado e escolha Save.

Para obter mais informações sobre a edição de máquinas de estado, consulte [Step Functions state machine language](#) no Developer GuideAWS Step Functions.

Tutorial: executar uma máquina de estado

Se desejar executar uma máquina de estado, o AWS Batch abre a página New execution (Nova execução) do console do Step Functions.

Como executar uma máquina de estado

1. Abra a página de [orquestração de fluxo de trabalho com tecnologia Step Functions do console do AWS Batch](#).
2. Escolha uma máquina de estado.
3. Clique em Executar.

O console do Step Functions abrirá a página New execution.

4. (Opcional) Edite a máquina de estado e escolha Start execution.

Para obter mais informações sobre a execução de máquinas de estado, consulte os [conceitos de execução da máquina de estado do Step Functions](#) no Guia do desenvolvedor do AWS Step Functions.

Stream de evento AWS Batch para o Amazon EventBridge

É possível usar o fluxo de eventos do AWS Batch para o Amazon EventBridge para receber notificações quase em tempo real sobre o estado atual de trabalhos em suas filas de trabalho.

Você pode usar o EventBridge para obter mais informações sobre seu serviço AWS Batch. Mais especificamente, você pode usá-lo para verificar o progresso dos trabalhos, criar fluxos de trabalho personalizados AWS Batch, gerar relatórios ou métricas de uso ou criar seus próprios painéis. Com o AWS Batch e o EventBridge, você não precisa da programação e do monitoramento do código que consulta continuamente o AWS Batch em busca de alterações no status do trabalho. Em vez disso, você pode lidar com mudanças no estado do trabalho AWS Batch de forma assíncrona usando uma variedade de destinos do Amazon EventBridge. Os destinos podem incluir AWS Lambda, Amazon Simple Queue Service, Amazon Simple Notification Service ou Amazon Kinesis Data Streams.

Os eventos do fluxo de eventos do AWS Batch serão entregues pelo menos uma vez. Caso eventos duplicados sejam enviados, o evento fornece informações suficientes para identificar cópias. Dessa forma, você pode comparar a data e hora do evento e o status do trabalho.

Os trabalhos AWS Batch estão disponíveis como destinos do EventBridge. Usando regras simples você pode corresponder a eventos e enviar trabalhos do AWS Batch em resposta a esses eventos. Para obter mais informações, consulte [What is EventBridge?](#) no Guia do usuário do Amazon EventBridge. Você também pode usar o EventBridge para programar ações automatizadas que são acionadas automaticamente em determinados momentos usando cron ou expressões de taxa. Para obter mais informações, consulte [Creating an Amazon EventBridge rule that runs on a schedule](#) no Amazon EventBridge User Guide. Para ver uma demonstração de exemplo, consulte [Trabalhos AWS Batch como destinos do EventBridge](#). Para obter informações sobre o uso do EventBridge Scheduler, consulte [Configurando o Agendador do Amazon EventBridge](#) no Guia do usuário do Amazon EventBridge.

Tópicos

- [AWS Batch Eventos do](#)
- [Tutorial: usar notificações de usuário da AWS com o AWS Batch](#)
- [Trabalhos AWS Batch como destinos do EventBridge](#)
- [Tutorial: recepcionar eventos de trabalho do AWS Batch usando o EventBridge](#)
- [Tutorial: Como enviar alertas do Amazon Simple Notification Service para eventos de trabalho que falharam](#)

AWS BatchEventos do

O AWS Batch envia eventos de alteração do status de trabalho para o EventBridge. O AWS Batch rastreia o estado de seus trabalhos. Se o status de um trabalho enviado anteriormente mudar, um evento será invocado. Por exemplo, se um trabalho no status RUNNING passar para o status FAILED. Esses eventos são classificados como eventos de alteração do estado de trabalho.

Note

O AWS Batch pode adicionar outros tipos de evento, origens e detalhes no futuro. Caso você esteja desserializando programaticamente dados JSON do evento, certifique-se de que a aplicação esteja preparada para lidar com propriedades desconhecidas. Isso é para evitar problemas se e quando essas propriedades adicionais forem adicionadas.

Eventos de alteração de estado do trabalho

Toda vez que um trabalho (enviado anteriormente) mudar de estado, um evento será criado. Para obter mais informações sobre os estados de trabalho do AWS Batch, consulte [Estados do trabalho](#).

Note

Os eventos não são criados para o envio de trabalho inicial.

Example Evento de alteração de estado do trabalho

Os eventos de alteração do estado da tarefa são entregues no formato a seguir. A seção detail se assemelha ao objeto da [JobDetail](#) retornado de uma operação de API [DescribeTasks](#) na AWS Batch API Reference. Para obter mais informações sobre os parâmetros do EventBridge, consulte [Events and Event Patterns](#) no Amazon EventBridge User Guide.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
```

```

    "resources": [
      "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
      "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobName": "event-test",
      "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
      "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue",
      "status": "RUNNABLE",
      "attempts": [],
      "createdAt": 1641944200058,
      "retryStrategy": {
        "attempts": 2,
        "evaluateOnExit": []
      },
      "dependsOn": [],
      "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-run-job-definition:1",
      "parameters": {},
      "container": {
        "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
        "command": [
          "sleep",
          "600"
        ],
        "volumes": [],
        "environment": [],
        "mountPoints": [],
        "ulimits": [],
        "networkInterfaces": [],
        "resourceRequirements": [
          {
            "value": "2",
            "type": "VCPU"
          }, {
            "value": "256",
            "type": "MEMORY"
          }
        ],
        "secrets": []
      },
      "propagateTags": false,

```

```

    "platformCapabilities": []
  }
}

```

Eventos bloqueados da fila de trabalhos

Sempre que o AWS Batch detecta um trabalho no estado `RUNNABLE` e, portanto, bloqueia uma fila, é criado um evento no Amazon CloudWatch Events. Para obter mais informações sobre as causas com suporte de filas bloqueadas, consulte [Trabalhos presos no status `RUNNABLE`](#). O mesmo motivo também está disponível no campo `statusReason` na ação da API [DescribeJobs](#).

Example Evento bloqueado da fila de trabalhos

Os eventos bloqueados da fila de trabalho são entregues no formato a seguir. A seção `detail` se assemelha ao objeto da [JobDetail](#) retornado de uma operação de API [DescribeTasks](#) na AWS Batch API Reference. Para obter mais informações sobre os parâmetros do EventBridge, consulte [Events and Event Patterns](#) no Amazon EventBridge User Guide.

```

{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
  ],
  "detail": {
    "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobName": "event-test",
    "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "attempts": [],
    "createdAt": 1641944200058,

```



```

    "retryStrategy": {
      "attempts": 2,
      "evaluateOnExit": []
    },
    "dependsOn": [],
    "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-run-job-definition:1",
    "parameters": {},
    "container": {
      "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
      "command": [
        "sleep",
        "600"
      ],
      "volumes": [],
      "environment": [],
      "mountPoints": [],
      "ulimits": [],
      "networkInterfaces": [],
      "resourceRequirements": [
        {
          "value": "2",
          "type": "VCPU"
        }, {
          "value": "256",
          "type": "MEMORY"
        }
      ],
      "secrets": []
    },
    "propagateTags": false,
    "platformCapabilities": []
  }
}

```

Eventos de alteração de estado do trabalho de serviço

Toda vez que um serviço existente mudar de estado, um evento será criado. Para obter mais informações sobre estados de trabalho de serviço, consulte [Mapeando AWS Batch o status do trabalho do serviço para o status de SageMaker IA](#).

Note

Os eventos não são criados para o envio de trabalho inicial.

Example Eventos de alteração de estado do trabalho de serviço

Os eventos de alteração do estado de trabalho de serviço são entregues no formato a seguir. A seção detail se assemelha à resposta que é retornada de uma operação de API [DescribeServiceJob](#) na Referência de API AWS Batch. Para obter mais informações sobre os parâmetros do EventBridge, consulte [Events and Event Patterns](#) no Amazon EventBridge User Guide.

Note

Os campos tags e serviceRequestPayload não estão incluídos no evento detail.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [
      {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-training-job88b610a69aa8380ca5b0a7aba3f81cb8"
        },
        "startedAt": 1641944300058,
        "stoppedAt": 1641944400058,
        "statusReason": "Received status from SageMaker: Training job completed"
      }
    ]
  }
}
```

```

    }
  ],
  "createdAt": 1641944200058,
  "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
  "jobId": "0bb17543-ece6-4480-b1a7-a556d344746b",
  "jobName": "event-test",
  "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-
training-job88b610a69aa8380ca5b0a7aba3f81cb8"
    }
  },
  "serviceJobType": "SAGEMAKER_TRAINING",
  "startedAt": 1641944300058,
  "status": "SUCCEEDED",
  "statusReason": "Received status from SageMaker: Training job completed",
  "stoppedAt": 1641944400058,
  "timeoutConfig": {
    "attemptDurationSeconds": 60
  }
}
}
}

```

Eventos bloqueados da fila de trabalhos de serviço

Sempre que o AWS Batch detecta uma fila bloqueada, é criado um evento no Amazon CloudWatch Events. O motivo da fila bloqueada está disponível no campo `statusReason` na ação da API [DescribeServiceJob](#).

Example Evento bloqueado da fila de trabalhos de serviço

Os eventos bloqueados da fila de trabalho de serviço são entregues no formato a seguir. A seção `detail` se assemelha à resposta que é retornada de uma operação de API [DescribeServiceJob](#) na Referência de API AWS Batch. Para obter mais informações sobre os parâmetros do EventBridge, consulte [Events and Event Patterns](#) no Amazon EventBridge User Guide.

 Note

Os campos `tags` e `serviceRequestPayload` não estão incluídos no evento `detail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [],
    "createdAt": 1641944200058,
    "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
    "jobId": "6271dfdf-d8a7-41b1-a4d2-55a2224f5375",
    "jobName": "event-test",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "timeoutConfig": {
      "attemptDurationSeconds": 60
    }
  }
}
```

Tutorial: usar notificações de usuário da AWS com o AWS Batch

Você pode usar as [notificações de usuários da AWS](#) para configurar canais de entrega para receber notificações sobre eventos do AWS Batch. Você recebe uma notificação quando um evento corresponde a uma regra especificada. É possível receber notificações para eventos por meio de diversos canais, incluindo o e-mail, o [Amazon Q Developer em aplicações de chat](#), notificações por chat ou notificações push do [AWS Console Mobile Application](#). Você também pode ver as

notificações na [Central de notificações do console](#). As notificações do usuário permitem agregação, o que pode reduzir o número de notificações recebidas durante eventos específicos.

Para configurar notificações no AWS Batch:

1. Abra o [console de AWS Batch](#).
2. Escolha Dashboard.
3. Escolha Configurar notificações.
4. Em AWS User Notifications, selecione Create notification configuration.

Para obter mais informações sobre como configurar e visualizar as notificações do usuário, consulte [Introdução às notificações AWS do usuário](#).

Trabalhos AWS Batch como destinos do EventBridge

O Amazon EventBridge oferece um fluxo quase em tempo quase real dos eventos do sistema que descrevem as alterações nos recursos da Amazon Web Services. Em geral, AWS Batch no Amazon Elastic Container Service, no Amazon Elastic Kubernetes Service e no AWS Fargate, os trabalhos estão disponíveis como destinos do EventBridge. Usando regras simples você pode corresponder a eventos e enviar trabalhos do AWS Batch em resposta a esses eventos. Para obter mais informações, consulte [What is EventBridge?](#) no Guia do usuário do Amazon EventBridge.

Também é possível usar o EventBridge para programar ações automatizadas que são acionadas automaticamente em determinados momentos usando expressões cron ou rate. Para obter mais informações, consulte [Creating an Amazon EventBridge rule that runs on a schedule](#) no Amazon EventBridge User Guide.

Para obter informações sobre como criar uma regra quando um evento corresponder com um padrão de evento, consulte [Creating Amazon EventBridge rules that react to events](#) no Guia do usuário do Amazon EventBridge.

Os casos de uso comuns para trabalhos AWS Batch como alvo do EventBridge incluem os seguintes casos de uso:

- Um trabalho agendado ocorre em intervalos de tempo regulares. Por exemplo, um trabalho cron ocorre somente em horários de baixa utilização, quando as Instâncias Spot do Amazon EC2 são mais baratas.

- Um trabalho AWS Batch é executado em resposta a uma operação de API registrada no CloudTrail. Por exemplo, um trabalho é enviado sempre que um objeto é carregado em um bucket específico do Amazon S3. Sempre que isso acontece, o transformador de entrada EventBridge passa o bucket e o nome da chave do objeto para os parâmetros AWS Batch.

 Note

Nesse cenário, todos os recursos AWS relacionados devem estar na mesma região. Isso inclui recursos como o bucket do Amazon S3, a regra do EventBridge e os registros do CloudTrail.

Antes que você possa enviar trabalhos AWS Batch com regras e destinos do EventBridge, o serviço do CloudWay Events exige várias permissões para executar trabalhos. AWS Batch Ao criar uma regra no console do EventBridge que especifica um trabalho do AWS Batch como destino, você tem a oportunidade de criar esse perfil. Para obter mais informações sobre a entidade principal do serviço e as permissões do IAM necessárias para esse perfil, consulte [EventBridge IAM role \(Função do IAM\)](#).

Tópicos

- [Tutorial: criar um trabalho programado do AWS Batch](#)
- [Tutorial: criar uma regra com um padrão de evento](#)
- [Tutorial: passar informações de eventos para um destino do AWS Batch em um cronograma usando o transformador de entrada do EventBridge](#)

Tutorial: criar um trabalho programado do AWS Batch

O procedimento a seguir aborda como criar um trabalho AWS Batch programado e o perfil necessário do IAM do EventBridge.

Para criar um trabalho programado do AWS Batch com o EventBridge

 Note

Esse procedimento funciona para todos os trabalhos AWS Batch do Amazon ECS, Amazon EKS e AWS Fargate.

1. Abra o console do Amazon EventBridge em <https://console.aws.amazon.com/events/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Regras.
4. Escolha Criar Regra.
5. Para Nome, especifique um nome exclusivo para seu ambiente de computação. O nome pode conter até 64 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).

 Note

Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.

6. (Opcional) Em Description, insira uma descrição para a regra.
7. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
8. (Opcional) Desative a regra no barramento selecionado se não quiser executá-la imediatamente.
9. Em Rule type, escolha Schedule.
10. Escolha Continuar para criar a regra ou Avançar.
11. Em Schedule pattern, siga um destes procedimentos:
 - Escolha um cronograma refinado que seja executado em um horário específico, tal como 8:00 a.m. PST na primeira segunda-feira de cada mês e insira a expressão cron. Para obter mais informações, consulte [Expressões Cron](#) no Guia do usuário do Amazon EventBridge.
 - Escolha uma programação que seja executada em uma taxa regular, como a cada 10 minutos. e, em seguida, insira uma expressão rate.
12. Escolha Next (Próximo).
13. Em Target types (Tipos de destino), escolha AWS service (Serviço da AWS).
14. Em Selecionar um destino, escolha Fila de trabalhos em lote. Em seguida, configure o seguinte:
 - Job queue: (Fila de trabalhos:), insira nome do recurso da Amazon (ARN) da fila de trabalhos na qual o trabalho será programado.

- Job definition: (Definição do trabalho:) insira o nome e a revisão ou ARN completo da definição do trabalho a ser usado.
 - Job name: (Nome do trabalho:) insira um nome para o trabalho.
 - Array size: (Tamanho da matriz:) (opcional) insira um tamanho de matriz para que o trabalho execute mais de uma cópia. Para obter mais informações, consulte [Trabalhos de matriz](#).
 - Job attempts: (opcional) insira o número de vezes para tentar novamente caso ocorra uma falha no trabalho. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).
15. Para os tipos de destino Batch job queue (Fila de trabalhos em lote), o Eventbridge precisa de permissão para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar o perfil do IAM necessário para a função ser executada. Execute um destes procedimentos:
- Para criar um perfil do IAM automaticamente, escolha Create a new role for this specific resource.
 - Para usar uma função do IAM que você criou anteriormente, escolha Usar função existente.
16. (Opcional) Expanda Additional settings.
- a. Em Configurar entrada de destino, escolha como o texto de um evento é processado antes de ser passado para o destino.
 - b. Em Idade máxima do evento, especifique o intervalo de tempo por quanto tempo os eventos não processados são mantidos.
 - c. Em Tentativas de repetição, insira o número de vezes que um evento é repetido.
 - d. Em Dead-letter queue, escolha uma opção de como os eventos não processados são tratados. Se necessário, especifique a fila do Amazon SQS para usar como fila de mensagens mortas.
17. (Opcional) Selecione Add another target para adicionar outro destino a essa regra.
18. Escolha Next (Próximo).
19. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar um rótulo de recurso à regra. Para obter mais informações, consulte [tags do Amazon EventBridge](#).
20. Escolha Next (Próximo).
21. Para Revisar e criar, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Create rule.

Para obter mais informações sobre como criar regras de eventos, consulte [Creating an Amazon EventBridge rule that runs on a schedule](#) no Guia do usuário do Amazon EventBridge.

Tutorial: criar uma regra com um padrão de evento

O procedimento a seguir aborda como criar uma regra com um padrão de evento.

Para criar uma regra que envie o evento para um destino quando o evento corresponder a um padrão definido

Note

Esse procedimento funciona para todos os trabalhos AWS Batch do Amazon ECS, Amazon EKS e AWS Fargate.

1. Abra o console do Amazon EventBridge em <https://console.aws.amazon.com/events/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Regras.
4. Escolha Criar Regra.
5. Para Nome, especifique um nome exclusivo para seu ambiente de computação. O nome pode conter até 64 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).

Note

Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.

6. (Opcional) Em Description, insira uma descrição para a regra.
7. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
8. (Opcional) Desative a regra no barramento selecionado se não quiser executá-la imediatamente.
9. Em Rule type, escolha Rule with an event pattern.
10. Escolha Próximo.

11. Em Event source, escolha Eventos da AWS ou eventos de parceiro do EventBridge.
12. (Opcional) Para evento de amostra:
 - a. Em Tipo de evento de amostra, escolha AWSEventos.
 - b. Em Eventos de amostra, escolha Batch Job State Change.
13. Em Creation method, escolha Use pattern form.
14. Para padrão de eventos:
 - a. Em Event source, escolha Serviços da AWS.
 - b. Em AWS service (Serviço da AWS), escolha Batch.
 - c. Em Event type, escolha Batch Job State Change.
15. Escolha Next (Próximo).
16. Em Target types (Tipos de destino), escolha AWS service (Serviço da AWS).
17. Em Choose a target type, selecione o tipo de destino. Por exemplo, escolha Batch job queue. Em seguida, especifique o seguinte:
 - Job queue: (Fila de trabalhos:), insira nome do recurso da Amazon (ARN) da fila de trabalhos na qual o trabalho será programado.
 - Job definition: (Definição do trabalho:) insira o nome e a revisão ou ARN completo da definição do trabalho a ser usado.
 - Job name: (Nome do trabalho:) insira um nome para o trabalho.
 - Array size: (Tamanho da matriz:) (opcional) insira um tamanho de matriz para que o trabalho execute mais de uma cópia. Para obter mais informações, consulte [Trabalhos de matriz](#).
 - Job attempts: (opcional) insira o número de vezes para tentar novamente caso ocorra uma falha no trabalho. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).
18. Para os tipos de destino Batch job queue (Fila de trabalhos em lote), o Eventbridge precisa de permissão para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar o perfil do IAM necessário para a função ser executada. Execute um destes procedimentos:
 - Para criar um perfil do IAM automaticamente, escolha Create a new role for this specific resource.
 - Para usar um perfil do IAM que você criou antes, escolha Use existing role.
19. (Opcional) Expanda Additional settings.

- a. Em Configure target input, escolha como o texto de um evento é processado.
 - b. Em Idade máxima do evento, especifique o intervalo de tempo por quanto tempo os eventos não processados são mantidos.
 - c. Em Tentativas de repetição, insira o número de vezes que um evento é repetido.
 - d. Em Dead-letter queue, escolha uma opção de como os eventos não processados são tratados. Se necessário, especifique a fila do Amazon SQS para usar como fila de mensagens mortas.
20. (Opcional) Selecione Add another target (Adicionar outro destino) para adicionar outro destino a essa regra.
 21. Escolha Next (Próximo).
 22. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar um rótulo de recurso. Para mais informações, consulte [Tags Amazon EventBridge](#) em Guia de Usuário Amazon EventBridge.
 23. Escolha Next (Próximo).
 24. Para Revisar e criar, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Create rule.

Para obter informações sobre como criar regras, consulte [Criar regras do Amazon EventBridge que reagem a eventos](#) no Guia do usuário do Amazon EventBridge.

Tutorial: passar informações de eventos para um destino do AWS Batch em um cronograma usando o transformador de entrada do EventBridge

É possível usar o transformador de entrada do EventBridge para transmitir informações do evento para o AWS Batch em um envio de trabalho. Isso pode ser especialmente valioso se você invocar trabalhos como resultado de outras informações do evento AWS. Um exemplo é upload de objeto para um bucket do Amazon S3. Você também pode usar uma definição de tarefa com valores de substituição de parâmetros no comando do contêiner. O transformador de entrada EventBridge pode fornecer os valores dos parâmetros com base nos dados do evento.

Em seguida, basta criar um destino de evento do AWS Batch que analise as informações do evento que o inicia e o transforma em um objeto `parameters`. Quando o trabalho é executado, os parâmetros do evento do acionador são repassados para o comando do contêiner do trabalho.

 Note

Nesse cenário, todos os recursos AWS (como buckets do Amazon S3, regras do EventBridge e registros do CloudTrail) devem estar na mesma região.

Para criar um destino do AWS Batch que use o transformador de entrada

1. Abra o console do Amazon EventBridge em <https://console.aws.amazon.com/events/>.
2. Na barra de navegação, selecione Região da AWS a ser usada.
3. No painel de navegação, escolha Regras.
4. Escolha Criar Regra.
5. Para Nome, especifique um nome exclusivo para seu ambiente de computação. O nome pode conter até 64 caracteres. Pode conter letras minúsculas, maiúsculas, números, hifens e (-) e sublinhados (_).

 Note

Uma regra não pode ter o mesmo nome que outra regra na mesma Região da AWS e no mesmo barramento de eventos.

6. (Opcional) Em Description, insira uma descrição para a regra.
7. Em Barramento de Eventos, escolha o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, selecione default. Quando um AWS service (Serviço da AWS) na sua conta emite um evento, ele sempre vai para o barramento de eventos padrão da sua conta.
8. (Opcional) Desative a regra no barramento selecionado se não quiser executá-la imediatamente.
9. Em Rule type, escolha Schedule.
10. Escolha Continuar para criar a regra ou Avançar.
11. Em Schedule pattern, siga um destes procedimentos:
 - Escolha um cronograma refinado que seja executado em um horário específico, tal como 8:00 a.m. PST na primeira segunda-feira de cada mês e insira a expressão cron. Para obter mais informações, consulte [Expressões Cron](#) no Guia do usuário do Amazon EventBridge.
 - Escolha uma programação que seja executada em uma taxa regular, como a cada 10 minutos. e, em seguida, insira uma expressão rate.

12. Escolha Next (Próximo).
13. Em Target types (Tipos de destino), escolha AWS service (Serviço da AWS).
14. Em Selecionar um destino, escolha Fila de trabalhos em lote. Em seguida, configure o seguinte:
 - Job queue: (Fila de trabalhos:), insira nome do recurso da Amazon (ARN) da fila de trabalhos na qual o trabalho será programado.
 - Job definition: (Definição do trabalho:) insira o nome e a revisão ou ARN completo da definição do trabalho a ser usado.
 - Job name: (Nome do trabalho:) insira um nome para o trabalho.
 - Array size: (Tamanho da matriz:) (opcional) insira um tamanho de matriz para que o trabalho execute mais de uma cópia. Para obter mais informações, consulte [Trabalhos de matriz](#).
 - Job attempts: (opcional) insira o número de vezes para tentar novamente caso ocorra uma falha no trabalho. Para obter mais informações, consulte [Repetições de trabalho automatizadas](#).
15. Para os tipos de destino Batch job queue (Fila de trabalhos em lote), o Eventbridge precisa de permissão para enviar eventos ao destino. Nesses casos, o Eventbridge pode criar o perfil do IAM necessário para a função ser executada. Execute um destes procedimentos:
 - Para criar um perfil do IAM automaticamente, escolha Create a new role for this specific resource.
 - Para usar uma função do IAM que você criou anteriormente, escolha Usar função existente.
16. (Opcional) Expanda Additional settings.
17. Na seção Additional settings, para Configure target input, escolha Input Transformer.
18. Escolha Configure input transformer.
19. (Opcional) Para evento de amostra:
 - a. Em Tipo de evento de amostra, escolha AWSeventos.
 - b. Em Eventos de amostra, escolha Batch Job State Change.
20. Na seção Target input transformer, para Input path, especifique os valores a serem analisados no evento de acionamento. Por exemplo, para analisar o evento Batch Job State Change, use o seguinte formato JSON.

```
{
  "instance": "$.detail.jobId",
  "state": "$.detail.status"
```

```
}
```

21. Em Template, insira o seguinte:

```
{  
  "instance": <jobId> ,  
  "status": <status>  
}
```

22. Escolha Confirmar.

23. Em Idade máxima do evento, especifique o intervalo de tempo por quanto tempo os eventos não processados são mantidos.

24. Em Tentativas de repetição, insira o número de vezes que um evento é repetido.

25. Em Dead-letter queue, escolha uma opção de como os eventos não processados são tratados. Se necessário, especifique a fila do Amazon SQS para usar como fila de mensagens mortas.

26. (Opcional) Selecione Add another target (Adicionar outro destino) para adicionar outro destino a essa regra.

27. Escolha Next (Próximo).

28. (Opcional) Em Tags, escolha Adicionar nova tag para adicionar um rótulo de recurso. Para mais informações, consulte [Tags Amazon EventBridge](#) em Guia de Usuário Amazon EventBridge.

29. Escolha Next (Próximo).

30. Para Revisar e criar, revise as etapas de configuração. Se precisar fazer alterações, escolha Edit (Editar). Quando terminar, escolha Create rule.

Tutorial: recepcionar eventos de trabalho do AWS Batch usando o EventBridge

Neste tutorial, você irá configurar uma AWS Lambda função simples para receptar AWS Batch eventos de trabalho e registrá-los em um fluxo de logs CloudWatch Logs.

Pré-requisitos

Este tutorial pressupõe um ambiente de computação e uma fila de tarefas prontas para aceitar tarefas. Caso você não tenha um ambiente de computação e fila de trabalho em execução do qual capturar eventos, siga as etapas em [Introdução aos AWS Batch tutoriais](#) para criar um. Ao final deste

tutorial, você poderá enviar opcionalmente uma tarefa para essa fila e testar se configurou a função do Lambda corretamente.

Tópicos

- [Tutorial: criar a função do Lambda](#)
- [Tutorial: registrar a regra do evento](#)
- [Tutorial: testar sua configuração](#)

Tutorial: criar a função do Lambda

Neste procedimento, você criará uma função do Lambda simples para funcionar como destino AWS Batch de mensagens do fluxo de eventos.

Para criar uma função do Lambda de destino

1. Abra o console AWS Lambda em <https://console.aws.amazon.com/lambda/>.
2. Escolha Criar Função e, em seguida, Criar do Início.
3. Para Nome da Função, insira batch-event-stream-handler.
4. Para Runtime, escolha Python 3.8.
5. Escolha Criar Função.
6. Na seção Fonte do Código, edite o código de amostra de acordo com o exemplo a seguir:

```
import json

def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
type of: aws.batch")

    print(json.dumps(event))
```

Essa é uma função Python 3.8 simples, que imprime eventos enviados por AWS Batch. Se tudo estiver configurado corretamente, no final deste tutorial, os detalhes do evento aparecerão no fluxo de log do CloudWatch Logs associado a essa função do Lambda.

7. Escolha Implantar.

Tutorial: registrar a regra do evento

Nesta seção, você criará uma regra de evento EventBridge que captura eventos de trabalho vindos dos seus recursos AWS Batch. Esta regra captura todos os eventos vindos de AWS Batch dentro da conta na qual estiver definida. As próprias mensagens de trabalho contêm informações sobre a origem do evento, inclusive fila de trabalhos onde foi inserido. Você pode usar essa informação para filtrar e classificar eventos de forma programática.

Note

Caso use o Console de gerenciamento da AWS para criar uma regra de evento, o console adicionará automaticamente as permissões do IAM ao EventBridge para chamar sua função do Lambda. No entanto, se criar uma regra de evento usando AWS CLI, você deverá conceder permissões explicitamente. Para mais informações, consulte [Eventos e Padrões de Evento](#) no Guia de Usuário Amazon EventBridge.

Para criar sua regra EventBridge

1. Abra o console Amazon EventBridge em <https://console.aws.amazon.com/events/>.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.

5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS. Quando um serviço AWS em sua conta emite um evento, ele sempre irá para o barramento de eventos padrão da conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.
8. Em Origem do Evento, escolha Outro.
9. Em Padrão de eventos, selecione Padrões personalizados (editor JSON).

10. Cole o padrão de evento a seguir na área de texto.

```
{
  "source": [
    "aws.batch"
  ]
}
```

Essa regra se aplica a todos os grupos AWS Batch e a cada evento AWS Batch. Como alternativa, você pode criar uma regra mais específica para filtrar resultados.

11. Escolha Próximo.

12. Em Tipos de Destino, escolha AWS Serviço.

13. Em Selecionar um Destino, escolha Função do Lambda e selecione sua função.

14. (Opcional) Para Configurações Adicionais, proceda da seguinte forma:

- a. Em Tempo Máximo do Evento, insira um valor entre um minuto (00:01) e 24 horas (24:00).
- b. Em Tentativas de Repetição, insira um número entre 0 e 185.
- c. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como fila de mensagens não entregues. O EventBridge enviará eventos que correspondam a essa regra para a fila de mensagens não entregues caso não sejam entregues com êxito ao destino. Faça um dos procedimentos a seguir:
 - Escolha Nenhum para não usar uma fila de mensagens não entregues.
 - Escolha Selecionar uma Fila Amazon SQS na Conta AWS Atual para usá-la como fila de mensagens não entregues e então, na lista suspensa, selecione a fila a ser usada.
 - Escolha Selecionar uma Fila Amazon SQS em qualquer outra AWS conta como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma política baseada em recurso à fila responsável por conceder permissão ao EventBridge para enviar mensagens. Para mais informações, consulte [Concedendo Permissões à Fila de Mensagens Não Entregues](#) do Guia de usuário Amazon EventBridge.

15. Escolha Próximo.

16. (Opcional) Insira uma ou mais tags para a regra. Para mais informações, consulte [Tags Amazon EventBridge](#) em Guia de Usuário Amazon EventBridge.

17. Escolha Próximo.

18. Analise os detalhes da regra e escolha Criar Regra.

Tutorial: testar sua configuração

Agora, você pode testar sua configuração EventBridge enviando uma tarefa para a fila de tarefas. Se tudo for configurado corretamente, sua função do Lambda será acionada e registrará os dados de eventos em um fluxo de logs CloudWatch Logs para a função.

Para testar sua configuração

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Insira uma nova AWS Batch tarefa. Para mais informações, consulte [Tutorial: enviar um trabalho](#).
3. Abra o console CloudWatch em <https://console.aws.amazon.com/cloudwatch/>.
4. No painel de navegação, escolha Logs e selecione o grupo de logs para sua função do Lambda (por exemplo, `/aws/lambda/my-function`).
5. Selecione um fluxo de log para visualizar os dados do evento.

Tutorial: Como enviar alertas do Amazon Simple Notification Service para eventos de trabalho que falharam

Neste tutorial, você configura uma regra de evento do Amazon EventBridge que apenas captura eventos de trabalho em que o trabalho foi alterado para um status de FAILED. No final deste tutorial, você pode optar por enviar um trabalho para essa fila de trabalhos. Isso serve para testar se você configurou seus alertas do Amazon SNS corretamente.

Pré-requisitos

Este tutorial pressupõe um ambiente de computação e uma fila de tarefas prontas para aceitar tarefas. Caso você não tenha um ambiente de computação e fila de trabalho em execução do qual capturar eventos, siga as etapas em [Introdução aos AWS Batch tutoriais](#) para criar um.

Tópicos

- [Tutorial: criar e assinar um tópico do Amazon SNS](#)
- [Tutorial: registrar uma regra de evento](#)
- [Tutorial: testar sua regra](#)
- [Regra alternativa: fila de trabalhos em lote bloqueada](#)

Tutorial: criar e assinar 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 (Tópicos), Create topic (Criar tópico).
3. Em Tipo, escolha Padrão.
4. Em Nome, insira **JobFailedAlert** e selecione Criar tópico.
5. Na tela JobFailedAlert, escolha Criar assinatura.
6. Em Protocolo, escolha Email.
7. Em Endpoint, insira um endereço de e-mail ao qual tenha acesso e escolha Criar assinatura.
8. 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.

Tutorial: registrar uma regra de evento

Em seguida, registre uma regra de evento que captura apenas eventos com falha de trabalho.

Para registrar sua regra do EventBridge

1. Abra o console do Amazon EventBridge em <https://console.aws.amazon.com/events/>.
2. No painel de navegação, escolha Regras.
3. Escolha Create rule.
4. Insira um nome e uma descrição para a regra.

Uma regra não pode ter o mesmo nome que outra na mesma Região e barramento de eventos.

5. Em Barramento de eventos, selecione o barramento de eventos que você deseja associar a essa regra. Se quiser que essa regra faça a correspondência com eventos provenientes da sua conta, escolha Barramento de eventos padrão da AWS. Quando um serviço AWS em sua conta emite um evento, ele sempre irá para o barramento de eventos padrão da conta.
6. Em Tipo de Regra, escolha Regra com Padrão de Evento.
7. Escolha Próximo.

8. Em Origem do Evento, escolha Outro.
9. Em Padrão de eventos, selecione Padrões personalizados (editor JSON).
10. Cole o padrão de evento a seguir na área de texto.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Esse código define uma regra de evento do EventBridge que corresponde a qualquer evento em que o status do trabalho é FAILED. Para obter mais informações sobre padrões de eventos, consulte [Eventos e padrões de eventos](#) no Guia do usuário do Amazon EventBridge.

11. Escolha Próximo.
12. Em Tipos de destino, escolha Serviço da AWS.
13. Em Selecionar destino, escolha Tópico SNS e, em Tópico, escolha JobFailedAlert.
14. (Opcional) Para Configurações Adicionais, proceda da seguinte forma:
 - a. Em Tempo Máximo do Evento, insira um valor entre um minuto (00:01) e 24 horas (24:00).
 - b. Em Tentativas de Repetição, insira um número entre 0 e 185.
 - c. Em Fila de mensagens não entregues, escolha se será usada uma fila padrão do Amazon SQS como fila de mensagens não entregues. O EventBridge enviará eventos que correspondam a essa regra para a fila de mensagens não entregues caso não sejam entregues com êxito ao destino. Faça um dos procedimentos a seguir:
 - Escolha Nenhum para não usar uma fila de mensagens não entregues.
 - Escolha Selecionar uma Fila Amazon SQS na Conta AWS Atual para usá-la como fila de mensagens não entregues e então, na lista suspensa, selecione a fila a ser usada.
 - Escolha Selecionar uma Fila Amazon SQS em qualquer outra AWS conta como fila de mensagens não entregues e insira o ARN da fila a ser usada. Você deve anexar uma

política baseada em recurso à fila responsável por conceder permissão ao EventBridge para enviar mensagens. Para mais informações, consulte [Concedendo Permissões à Fila de Mensagens Não Entregues](#) do Guia de usuário Amazon EventBridge.

15. Escolha Próximo.
16. (Opcional) Insira uma ou mais tags para a regra. Para mais informações, consulte [Tags Amazon EventBridge](#) em Guia de Usuário Amazon EventBridge.
17. Escolha Próximo.
18. Analise os detalhes da regra e selecione Criar regra.

Tutorial: testar sua regra

Para testar sua regra, envie um trabalho que saia logo após começar com um código de saída diferente de zero. Caso a regra de evento esteja configurada corretamente, você deverá receber uma mensagem de e-mail em alguns minutos com o texto do evento.

Para testar uma regra

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Insira uma nova AWS Batch tarefa. Para obter mais informações, consulte [Tutorial: enviar um trabalho](#). Para o comando do trabalho, substitua este comando para sair do contêiner com um código de saída 1.

```
/bin/sh, -c, 'exit 1'
```

3. Verifique seu e-mail para confirmar que você recebeu um e-mail de alerta para a notificação de trabalho com falha.

Regra alternativa: fila de trabalhos em lote bloqueada

Para criar uma regra de evento que monitore a fila de trabalhos em lote bloqueados, repita esses tutoriais com as alterações a seguir:

1. Em [Tutorial: criar e assinar um tópico do Amazon SNS](#), use *BlockedJobQueue* como nome do tópico.
2. Em [Tutorial: registrar uma regra de evento](#), use o seguinte padrão no editor JSON:

```
{
```

```
{
  "detail-type": [
    "Batch Job Queue Blocked"
  ],
  "source": [
    "aws.batch"
  ]
}
```

Elastic Fabric Adapter

Um Elastic Fabric Adapter (EFA) é um dispositivo de rede para acelerar aplicativos de Computação de Alto Desempenho (HPC). O AWS Batch oferece suporte a aplicativos que usam o EFA se as condições a seguir são atendidas.

- Para obter uma lista de tipos de instância que oferecem suporte a EFAs, consulte [Tipos de instância compatíveis](#) no Guia do usuário do Amazon EC2.

Tip

Para ver uma lista de tipos de instância que oferecem suporte a EFAs em um Região da AWS, execute o comando a seguir. Em seguida, faça referência cruzada à lista que é retornada com a lista de tipos de instância disponíveis no console AWS Batch.

```
$ aws ec2 describe-instance-types --region us-east-1 --filters Name=network-info.efa-supported,Values=true --query "InstanceTypes[*].[InstanceType]" --output text | sort
```

- Para obter uma lista de sistemas operacionais compatíveis, consulte [Supported operating systems](#).
- A AMI tem o driver EFA carregado.
- O grupo de segurança do EFA deve permitir todo o tráfego de entrada e saída de e para o próprio grupo de segurança.
- Todas as instâncias que usam um EFA devem estar no mesmo grupo com posicionamento em cluster.
- A definição do trabalho deve incluir um membro `devices` com `hostPath` definido como `/dev/infiniband/uverbs0` para permitir que o dispositivo EFA seja transmitido ao contêiner. Se `containerPath` for especificado, ele também deverá ser definido como `/dev/infiniband/uverbs0`. Se `permissions` estiver definido, ele deverá ser definido como `READ | WRITE | MKNOD`.

A localização do membro [LinuxParameters](#) será diferente para trabalhos paralelos de vários nós e trabalhos de contêiner de nó único. Os exemplos a seguir demonstram as diferenças, mas estão faltando valores necessários.

Example Exemplo de trabalho paralelo de vários nós

```
{
```

```

"jobDefinitionName": "EFA-MNP-JobDef",
"type": "multinode",
"nodeProperties": {
  ...
  "nodeRangeProperties": [
    {
      ...
      "container": {
        ...
        "linuxParameters": {
          "devices": [
            {
              "hostPath": "/dev/infiniband/uverbs0",
              "containerPath": "/dev/infiniband/uverbs0",
              "permissions": [
                "READ", "WRITE", "MKNOD"
              ]
            }
          ],
        },
      },
    },
  ],
},
},
},
},
},
},
},
},
},
},
}

```

Example Exemplo de trabalho de contêiner de nó único.

```

{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
      ],
    },
  },
},
},
}

```


Para obter mais informações, consulte [Elastic Fabric Adapter](#) no Guia do usuário do Amazon EC2.

Monitorar AWS Batch

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho de AWS Batch sua AWS solução.

Recomendamos fortemente que você colete dados de monitoramento de todas as partes da sua AWS solução para facilitar a depuração de uma falha multiponto, caso ocorra. Comece criando um plano de monitoramento que responda às seguintes perguntas. Se você não tiver certeza de como responder a essas perguntas, ainda poderá usar o Amazon CloudWatch Logs para estabelecer suas linhas de base de desempenho.

- 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?

Sua próxima etapa é estabelecer uma linha de base do AWS Batch desempenho normal em seu ambiente medindo o desempenho em vários momentos e sob diferentes condições de carga. Ao monitorar AWS Batch, mantenha os dados históricos de monitoramento para poder compará-los com os dados de desempenho atuais. Isso ajuda a identificar padrões normais e anomalias de desempenho e a criar métodos para tratar os problemas.

Os tópicos desta seção podem ajudar você a começar o registro em log e o monitoramento da AWS Batch.

Tópicos

- [Usando CloudWatch registros com AWS Batch](#)
- [AWS Batch CloudWatch Container Insights](#)
- [Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS](#)

Usando CloudWatch registros com AWS Batch

Você pode configurar seus AWS Batch trabalhos em EC2 recursos para enviar informações e métricas detalhadas de CloudWatch registros para o Logs. Ao fazer isso, você pode exibir logs diferentes das tarefas em um local conveniente. Para obter mais informações sobre CloudWatch registros, consulte [O que é Amazon CloudWatch Logs?](#) no Guia do CloudWatch usuário da Amazon.

Note

Por padrão, CloudWatch os registros estão ativados para contêineres AWS Fargate.

Para ativar e personalizar o registro de CloudWatch registros, revise as seguintes tarefas de configuração únicas:

- Para ambientes AWS Batch computacionais baseados em EC2 recursos, adicione uma política do IAM à `ecsInstanceRole` função. Para obter mais informações, consulte [the section called “Tutorial: Adicionar uma política do CloudWatch Logs IAM”](#).
- Crie um modelo de EC2 lançamento da Amazon que inclua CloudWatch monitoramento detalhado e, em seguida, especifique o modelo ao criar seu ambiente AWS Batch computacional. Você também pode instalar o CloudWatch agente em uma imagem existente e depois especificar a imagem no assistente de AWS Batch primeira execução.
- (Opcional) Configure o driver `awslogs`. Você pode adicionar parâmetros que alteram o comportamento padrão em ambos os recursos EC2 e do Fargate. Para obter mais informações, consulte [the section called “Usar o driver de log `awslogs`”](#).

Tópicos

- [Tutorial: Adicionar uma política do CloudWatch Logs IAM](#)
- [Instalar e configurar o CloudWatch agente](#)
- [Tutorial: Exibir CloudWatch registros](#)

Tutorial: Adicionar uma política do CloudWatch Logs IAM

Antes que seus trabalhos possam enviar dados de registro e métricas detalhadas para o CloudWatch Logs, você precisa criar uma política do IAM que use os CloudWatch Logs APIs. Após criar a política do IAM, anexe a política a um perfil `ecsInstanceRole`.

Note

Se a ECS-CloudWatchLogs política não estiver vinculada à `ecsInstanceRole` função, as métricas básicas ainda poderão ser enviadas ao CloudWatch Logs. No entanto, as métricas básicas não incluem dados de registro ou métricas detalhadas, como espaço livre em disco.

AWS Batch ambientes computacionais usam EC2 recursos da Amazon. Quando você cria um ambiente computacional usando o assistente de AWS Batch primeira execução, AWS Batch cria a `ecsInstanceRole` função e configura o ambiente com ela.

Se você não estiver usando o assistente de primeira execução, poderá especificar a `ecsInstanceRole` função ao criar um ambiente computacional na API AWS Command Line Interface ou AWS Batch . Para obter mais informações, consulte [AWS CLI Command Reference](#) ou [AWS Batch API Reference](#).

Para criar a política de IAM **ECS-CloudWatchLogs**

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Políticas.
3. Escolha Create policy.
4. Escolha JSON e insira a seguinte política:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams"
      ],
      "Resource": [
        "arn:aws:logs:*:*:*"
      ]
    }
  ]
}
```

```
}  
]  
}
```

5. Escolha Next: Tags.
6. (Opcional) Em Adicionar tags, escolha Adicionar tag para adicionar uma tag à política.
7. Escolha Next: Review.
8. Na página Review policy, em Name, digite **ECS-CloudWatchLogs**, e, em seguida, digite uma Description opcional.
9. Escolha Create policy.

Para anexar a política **ECS-CloudWatchLogs** a **ecsInstanceRole**

1. Abra o console do IAM em <https://console.aws.amazon.com/iam/>.
2. No painel de navegação, selecione Roles.
3. Selecione ecsInstanceRole. Se o perfil não existir, siga os procedimentos em [Perfil de instância do Amazon ECS](#) para criar o perfil.
4. Escolha Adicionar Permissões e depois escolha Anexar Políticas.
5. Escolha a CloudWatchLogs política ECS e, em seguida, escolha Anexar política.

Instalar e configurar o CloudWatch agente

Você pode criar um modelo de EC2 lançamento da Amazon que inclua CloudWatch monitoramento. Para obter mais informações, consulte [Iniciar uma instância a partir de um modelo de lançamento](#) e [detalhes avançados](#) no Guia EC2 do usuário da Amazon.

Você também pode instalar o CloudWatch agente em uma Amazon EC2 AMI existente e depois especificar a imagem no assistente de AWS Batch primeira execução. Para obter mais informações, consulte [Instalação do CloudWatch agente](#) [Introdução aos AWS Batch tutoriais](#) e.

Note

Os modelos de lançamento não são compatíveis com os AWS Fargate recursos.

Tutorial: Exibir CloudWatch registros

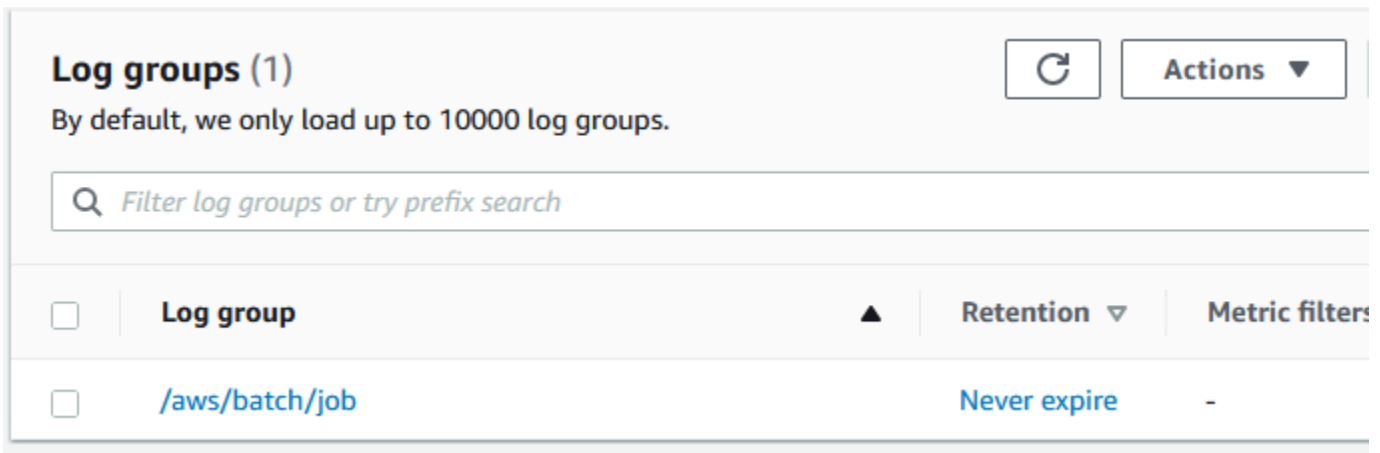
Você pode visualizar e pesquisar CloudWatch registros de registros no Console de gerenciamento da AWS.

Note

Pode levar alguns minutos para que os dados sejam exibidos nos CloudWatch registros.

Para ver seus dados CloudWatch de registros

1. Abra o CloudWatch console em <https://console.aws.amazon.com/cloudwatch/>.
2. No painel de navegação à esquerda, escolha Logs e, em seguida, escolha Grupos de logs.



3. Escolha um grupo de logs para visualizar.

Log streams (9)			Delete	Create log stream	Search all
Filter log streams or try prefix search		< 1 >			
☐	Log stream			Last event time	
☐	Test-jd/default/6622fe43-b2a3-4805-a0a6-3828329cc32b			2020-08-18T19:50:19.311Z	
☐	first-run-job-definition/default/86ed75ac-4f3f-4044-8fb0-dfd9c85ae6b2			2020-08-18T02:07:42.738Z	
☐	Test-jd/default/48f4a9dd-be07-4b43-8696-f0995eefe28b			2020-08-14T00:18:19.395Z	
☐	first-run-job-definition/default/d7d5ccf4-a0a0-44f1-bf36-35f2b3632912			2020-08-13T22:39:06.936Z	
☐	gpuJD/default/6ecf8ffb-ee03-4041-aa18-ab5e7a6dff0d			2019-03-26T08:48:39.637Z	

4. Escolha um stream de logs para visualizar. Por padrão, os fluxos são identificados pelos primeiros 200 caracteres do nome do trabalho e pelo ID da tarefa do Amazon ECS.

Tip

Para baixar os dados do fluxo de logs, escolha Ações.

Log events

Actions ▼

Create Metric Filter

Filter events

Clear

1m

30m

1h

12h

Custom

▶

Timestamp

Message

There are older events to load. [Load more.](#)

▶

2020-08-17T19:07:42.738-07:00...

'hello world'

No newer events at this moment. *Auto retry paused.* [Resume](#)

AWS Batch CloudWatch Container Insights

O CloudWatch Container Insights coleta, agrega e resume métricas e logs dos seus AWS Batch ambientes de computação e tarefas. As métricas incluem uso de CPU, memória, disco e rede. Você pode adicionar essas métricas aos painéis do CloudWatch.

Os dados operacionais são coletados como eventos de log de desempenho. Essas entradas usam um esquema JSON estruturado que permite que dados de alta cardinalidade sejam ingeridos e armazenados em escala. Com base nesses dados, o CloudWatch cria métricas agregadas de alto nível no ambiente de computação e nível de tarefa como métricas do CloudWatch. Para obter mais informações, consulte [Logs Estruturados do Container Insights para Amazon ECS](#) no Guia de Usuário do Amazon CloudWatch.

Important

O CloudWatch Container Insights é cobrado como “métricas personalizadas” pelo CloudWatch. Para mais informações, consulte [Precificação do Amazon CloudWatch Events](#)

Tópicos

- [Ativar o Container Insights](#)

Ativar o Container Insights

Conclua as etapas a seguir para ativar o Container Insights para ambientes de computação do AWS Batch.

1. Abra o [AWS Batch Console](#).
2. Escolha Environments (Ambientes).
3. Escolha o ambiente de computação que deseja.
4. Na aba Insights de contêiner, ative Insights de contêiner para o ambiente computacional.

Tip

Você pode selecionar um intervalo padrão para agregar as métricas ou criar um intervalo personalizado.

Por padrão, as seguintes métricas serão exibidas. Para obter uma lista completa de métricas Amazon ECS Container Insights, consulte [Métricas Amazon ECS Container Insights](#) no Guia de Usuário do Amazon CloudWatch.

- **JobCount** – número de tarefas executadas no ambiente de computação.
- **ContainerInstanceCount** – número de instâncias do Amazon Elastic Compute Cloud que executam o agente do Amazon ECS e estão registradas no ambiente de computação.
- **MemoryReserved** – Memória reservada pelas tarefas do ambiente de computação. Essa métrica é coletada apenas para tarefas com uma reserva de memória definida na definição da tarefa.
- **MemoryUtilized** – Memória sendo usada por tarefas do ambiente de computação. Essa métrica é coletada apenas para tarefas com reserva de memória definida na definição da tarefa.
- **CpuReserved** – Unidades de CPU reservadas pelas tarefas do ambiente de computação. Essa métrica é coletada apenas para tarefas com reserva de CPU definida na definição de tarefa.
- **CpuUtilized** – Unidades de CPU usadas por tarefas no ambiente de computação. Essa métrica é coletada apenas para tarefas com reserva de CPU definida na definição de tarefa.
- **NetworkRxBytes** - Número de bytes recebidos. Essa métrica está disponível apenas para contêineres em tarefas que usem modos de awsvpc ou de rede de ponte.
- **NetworkTxBytes** – Número de bytes transmitidos. Essa métrica está disponível apenas para contêineres em tarefas que usem modos de awsvpc ou de rede de ponte.
- **StorageReadBytes** – Número de bytes lidos do armazenamento.
- **StorageWriteBytes** – Número de bytes gravados no armazenamento.

Use o CloudWatch Logs para monitorar AWS Batch trabalhos do Amazon EKS

Você pode usar o Amazon CloudWatch Logs para monitorar, armazenar e visualizar todos os seus arquivos de log em um único local. Usando o CloudWatch Logs, você pode pesquisar, filtrar e analisar dados de log de várias fontes.

Você pode baixar uma AWS para imagem Fluent Bit que inclui um plug-in para monitorar AWS Batch em trabalhos do Amazon EKS no CloudWatch Logs. Fluent Bit é um processador e encaminhador de registros de código aberto que é compatível com Docker e Kubernetes. Recomendamos que você use Fluent Bit como roteador de log porque ele consome menos recursos do que Fluentd. Consulte mais informações em [Instalação do atendente CloudWatch com o complemento de observabilidade do EKS do Amazon CloudWatch ou com o chart do Helm](#).

Pré-requisitos

- Anexe uma política `CloudWatchAgentServerPolicy` à política AWS Identity and Access Management de seus nós de processamento. Para obter mais informações, consulte [Verify prerequisites](#).

Instalar o complemento

Para obter instruções sobre como instalar o AWS para Fluent Bit e criar grupos do CloudWatch, consulte [Instalar atendente CloudWatch com o complemento de observabilidade EKS do Amazon CloudWatch ou o chart do Helm](#).

Você deve fornecer os seguintes [dados de configuração adicionais](#) ao instalar o complemento:

- Se você instalar o complemento com o Console de gerenciamento da AWS, precisará fornecer as seguintes tolerâncias nos Valores de configuração:

```
{
  "tolerations": [
    {
      "key": "batch.amazonaws.com/batch-node",
      "operator": "Exists"
    }
  ]
}
```

- Se você instalar o complemento com o AWS CLI, adicione os seguintes argumentos:

```
--configuration-values '{"tolerations":[{"key":"batch.amazonaws.com/batch-node","operator":"Exists"}]}'
```

Tip

Lembre-se de que Fluent Bit usa 0.5 de CPU e 100 MB de memória nos nós AWS Batch. Isso reduz a capacidade total disponível para trabalhos AWS Batch. Considere isso ao dimensionar seus trabalhos.

Marcar com tag os recursos do AWS Batch

Para ajudar no gerenciamento de recursos AWS Batch, você pode atribuir seus próprios metadados a cada recurso em forma de tags. Este tópico descreve as tags e como criá-las.

Tópicos

- [Conceitos Básicos de Tags](#)
- [Marcar com tag os recursos do](#)
- [Restrições de tags](#)
- [Tutorial: gerenciar etiquetas usando o console](#)
- [Gerenciar etiquetas usando a CLI ou a API](#)

Conceitos Básicos de Tags

Uma tag um rótulo atribuído a um recurso AWS. Cada tag consiste em uma chave e um valor opcional, ambos definidos por você.

As tags permitem categorizar seus recursos AWS por finalidade, proprietário ou ambiente, por exemplo. Caso possua muitos recursos do mesmo tipo, você pode identificar rapidamente um recurso específico com base nas tags atribuídas a ele. Por exemplo, é possível definir um conjunto de tags para seus serviços AWS Batch para ajudá-lo a rastrear o proprietário e nível da pilha de cada serviço. Recomendamos planejar um conjunto consistente de chaves de tags para cada tipo de recurso.

Tags não são automaticamente atribuídas aos recursos. Após adicionar uma tag, você pode editar as chaves e os valores das tags ou removê-las de um recurso a qualquer momento. Caso exclua um recurso, todas as respectivas tags também serão excluídas.

As tags não têm significado semântico atrelado a AWS Batch e são interpretadas estritamente como string de caracteres. É possível definir o valor de uma tag em uma string vazia, mas não configurar o valor de um tag como nula. Caso adicione uma tag com a mesma chave de outra existente no recurso, o novo valor substituirá o antigo.

Você pode trabalhar com tags usando Console de gerenciamento da AWS, AWS CLI e API de AWS Batch.

Caso esteja usando AWS Identity and Access Management (IAM), você pode controlar quais usuários em sua conta AWS têm permissão para criar, editar ou excluir tags.

Marcar com tag os recursos do

Você pode marcar ambientes de computação AWS Batch, tarefas, definições de tarefas, filas de tarefas, políticas de agendamento novas ou existentes.

Caso esteja usando o console AWS Batch, você pode aplicar tags a novos recursos quando estes forem criados ou a recursos existentes a qualquer momento, por meio da guia Tags na página de recursos relevante.

Caso esteja usando a API AWS Batch, AWS CLI ou SDK AWS, é possível aplicar tags a novos recursos por meio do parâmetro `tags` na ação API relevante ou, para recursos existentes, da ação API `TagResource`. Para mais informações, consulte [TagResource](#).

Algumas ações de criação de recursos permitem especificar tags para um recurso quando o mesmo for criado. Caso as tags não possam ser aplicadas durante a criação dos recursos, haverá falha no processo de criação de recursos. Isso garante que recursos que você pretenda marcar na criação sejam criados com as tags especificadas ou não. Caso marque recursos no momento da criação, não precisará executar scripts de marcação personalizados após a criação do recurso.

A tabela a seguir descreve os recursos AWS Batch que podem ser marcados com tags e aqueles que podem ser marcados na criação.

Suporte para etiquetas de recursos do AWS Batch

Recurso	Compatível com tags	Compatível com a propagação de tags	Compatível com o uso de tags na criação (API AWS Batch, AWS CLI e SDK AWS)
Ambientes de computação AWS Batch	Sim	Não. As tags de ambiente de computação não são propagadas para nenhum outro recurso. Tags	Sim

Recurso	Compatível com tags	Compatível com a propagação de tags	Compatível com o uso de tags na criação (API AWS Batch, AWS CLI e SDK AWS)
		para recursos são especificadas nas tags membros do objeto ComputerResources transmitidas na operação API CreateComputeEnvironment .	
Tarefas AWS Batch	Sim	Sim	Sim
Definições de trabalho AWS Batch	Sim	Não	Sim
Filas de tarefa AWS Batch	Sim	Não	Sim
Políticas de agendamento AWS Batch	Sim	Não	Sim

Restrições de tags

As restrições básicas a seguir se aplicam a tags:

- Número máximo de tags por recurso — 50
- Em todos os recursos, cada chave de tag deve ser exclusiva e possuir apenas um valor.
- Comprimento máximo da chave — 128 caracteres Unicode em UTF-8
- Comprimento máximo do valor — 256 caracteres Unicode em UTF-8
- Caso seu esquema de marcação seja usado em vários serviços e recursos AWS, lembre-se de que outros serviços podem possuir restrições em caracteres permitidos. Em geral, caracteres

permitidos incluem letras, números, espaços representáveis em UTF-8 e os caracteres + - = . _ : / @.

- Chaves e valores de tags diferenciam maiúsculas de minúsculas.
- Não use `aws:`, `AWS:` ou qualquer combinação de letras maiúsculas e minúsculas como um prefixo para chaves ou valores, uma vez que as mesmas são reservadas para uso AWS. Você não pode editar nem excluir chaves ou valores de tags com esse prefixo. Tags com esse prefixo não contam em limites de tags por recurso.

Tutorial: gerenciar etiquetas usando o console

Ao utilizar o console do AWS Batch, você gerencia as tags associadas a ambientes de computação, tarefas, definições de tarefas e filas de tarefas novas ou existentes.

Adicionar etiquetas a um recurso individual na criação

Você pode adicionar tags a ambientes de computação AWS Batch, tarefas, definições de tarefas, filas de tarefas e políticas de agendamento ao criá-las.

Adicionar e excluir tags em um recurso individual

AWS Batch permite adicionar ou excluir tags associadas a clusters diretamente na página do recurso.

Para adicionar ou excluir uma etiqueta em um recurso individual

1. Abra o console AWS Batch em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, escolha a Região a ser usada.
3. No painel de navegação, escolha um tipo de recurso (por exemplo, Filas de Tarefas).
4. Escolha um recurso específico e, em seguida, Editar Tags.
5. Adicione ou exclua tags conforme necessário.
 - Para adicionar uma tag, especifique a chave e o valor nas caixas de texto vazias ao final da lista.
 - Para excluir uma tag, escolha o botão

Delete icon
ao lado.

- Repita esse processo para cada tag que desejar adicionar ou excluir e escolha Editar Tags para concluir.

Gerenciar etiquetas usando a CLI ou a API

Use os seguintes comandos AWS CLI ou operações de API da AWS Batch para adicionar, atualizar, listar e excluir as tags de seus recursos.

Suporte para etiquetas de recursos do AWS Batch

Tarefa	Ação API	AWS CLI	AWS Tools for Windows PowerShell
Adicione ou sobrescreva uma ou mais tags.	TagResource	tag-resource	Add-BATResourceTag
Exclua uma ou mais tags.	UntagResource	untag-resource	Remove-BATResourceTag
Listar as tags para um recurso	ListTagsForResource	list-tags-for-resource	Get-BATResourceTag

Os exemplos a seguir mostram como marcar ou desmarcar recursos usando AWS CLI.

Exemplo 1: marcar um recurso existente

O comando a seguir marca um recurso existente.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Exemplo 2: desmarcar um recurso existente

O comando a seguir exclui uma tag de um recurso existente.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Exemplo 3: listar etiquetas para um recurso

O comando a seguir lista as tags associadas a um recurso existente.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Algumas ações de criação de recursos permitem especificar tags ao criar o recurso. As ações a seguir são compatíveis com o uso de tags na criação.

Tarefa	Ação API	AWS CLI	AWS Tools for Windows PowerShell
Crie um ambiente de computação	CreateComputeEnvironment	create-compute-environment	New-BATComputeEnvironment
Crie uma fila de trabalhos	CreateJobQueue	create-job-queue	New-BATJobQueue
Criar uma política de agendamento	Criar política de agendamento	create-scheduling-policy	New-BATSchedulingPolicy
Registrar uma definição de tarefa	RegisterJobDefinition	register-job-definition	Register-BATJobDefinition
Enviar uma tarefa	SubmitJob	submit-job	Submit-BATJob
Criar um recurso consumível	CreateConsumableResource	create-consumable-resource	Create-BATConsumableResource

Práticas recomendadas do AWS Batch

Você pode usar o AWS Batch para executar uma variedade de workloads computacionais exigentes em grande escala sem gerenciar uma arquitetura complexa. Os trabalhos do AWS Batch podem ser usados em uma ampla variedade de casos de uso em áreas como epidemiologia, jogos e machine learning.

Este tópico aborda as práticas recomendadas a serem consideradas durante o uso do AWS Batch e instruções sobre como executar e otimizar suas workloads ao utilizar do AWS Batch.

Tópicos

- [Quando utilizar AWS Batch](#)
- [Lista de verificação para execução em escala](#)
- [Otimize contêineres e AMIs](#)
- [Escolha a opção direita de recurso de ambiente de computação](#)
- [Amazon EC2 Sob Demanda, ou Amazon EC2 Spot](#)
- [Use as práticas recomendadas do Amazon EC2 Spot para AWS Batch](#)
- [Erros Comuns e Solução de Problemas](#)

Quando utilizar AWS Batch

AWS Batch executa trabalhos em escala a baixo custo, além de prover serviços de filas em escala com otimização de custos. No entanto, nem toda workload é adequada para ser executada utilizando AWS Batch.

- **Trabalhos curtos:** se um trabalho for executado por apenas alguns segundos, a sobrecarga para agendar o trabalho em lotes poderá demorar mais do que o runtime do trabalho em si. Como solução alternativa, reúna suas tarefas binpack antes de inseri-las em AWS Batch. Em seguida, configure seus trabalhos do AWS Batch para repetir as tarefas. Por exemplo, prepare os argumentos de tarefas individuais em uma tabela do Amazon DynamoDB, ou como um arquivo em um bucket do Amazon S3. Considere agrupar tarefas para que os trabalhos sejam executados por 3-5 minutos cada. Depois de binpack os trabalhos, faça loop pelos seus grupos de tarefas em seu trabalho AWS Batch.
- **Trabalhos que devem ser executados imediatamente:** AWS Batch pode processar trabalhos rapidamente. No entanto, o AWS Batch é um programador, que otimiza para desempenho de

custo, prioridade de trabalho e throughput. AWS Batch pode exigir tempo para processar suas solicitações. Se você precisar de uma resposta em poucos segundos, uma abordagem baseada em serviço utilizando o Amazon ECS ou o Amazon EKS é mais adequada.

Lista de verificação para execução em escala

Antes de executar uma grande workload em 50 mil ou mais vCPUs, considere a lista de verificação a seguir.

Note

Se você planeja executar uma grande workload em um milhão ou mais vCPUs, ou precisa de orientação para execução em grande escala, entre em contato com sua equipe AWS.

- Verifique suas cotas do Amazon EC2: verifique suas cotas do Amazon EC2 (também conhecidas como limites) no painel Service Quotas do Console de gerenciamento da AWS. Se necessário, solicite um aumento de cota para seu número máximo de instâncias do Amazon EC2. Lembre-se de que o Amazon EC2 Spot e as instâncias sob demanda têm cotas separadas. Para obter mais informações, consulte [Conceitos Básicos de Service Quotas](#).
- Verifique sua cota do Amazon Elastic Block Store para cada Região: cada instância utiliza um volume GP2 ou GP3 para o sistema operacional. Por padrão, a cota para cada Região da AWS é de 300 TiB. No entanto, cada instância utiliza contagens de uso como parte dessa cota. Portanto, certifique-se de levar isso em consideração ao verificar sua cota do Amazon Elastic Block Store para cada Região. Se sua cota for alcançada, você não poderá criar mais instâncias. Para obter mais informações, consulte [Endpoints e Cotas do Amazon Elastic Block Store](#).
- Utilize o Amazon S3 para armazenamento: o Amazon S3 fornece alto throughput e ajuda a eliminar a suposição sobre quanto armazenamento provisionar baseado no número de trabalhos e instâncias em cada Zona de Disponibilidade. Para obter mais informações, consulte [Padrões de Design de Práticas Recomendadas: Otimizando a Performance do Amazon S3](#).
- Escale gradualmente para identificar gargalos com antecedência: Para um trabalho em execução em um milhão ou mais de vCPUs, comece menor e aumente gradualmente para que você possa identificar gargalos mais cedo. Por exemplo, comece executando em 50 mil vCPUs. Em seguida, aumente a contagem para 200 mil vCPUs, depois 500 mil vCPUs, e assim por diante. Em outras palavras, continue aumentando gradualmente a contagem de vCPUs até alcançar o número de vCPUs desejado.

- Monitore para identificar possíveis problemas com antecedência: Para evitar possíveis interrupções e problemas ao executar em escala, certifique-se de monitorar seu aplicativo e sua arquitetura. Podem ocorrer interrupções mesmo ao escalar de 1 mil para 5 mil vCPUs. Você pode utilizar o Amazon CloudWatch Logs para analisar dados de log ou utilizar o CloudWatch Embedded Metrics, com uma biblioteca de cliente. Para obter mais informações, consulte [Referência do Atendente do CloudWatch Logs](#) e [aws-embedded-metrics](#)

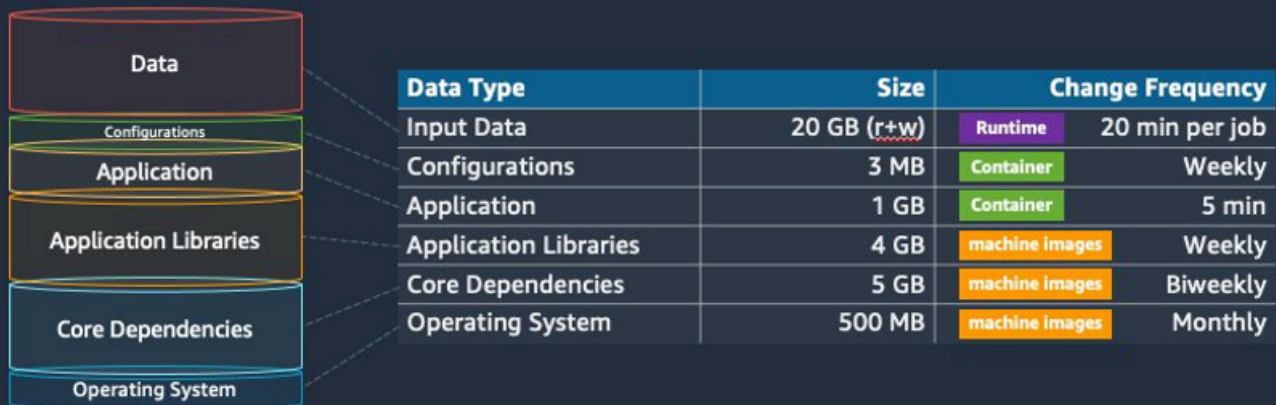
Otimize contêineres e AMIs

O tamanho e a estrutura do contêiner são importantes para o primeiro conjunto de trabalhos que você executar. Isso é especialmente verdadeiro quando o contêiner é maior que 4 GB. As imagens de contêiner são integradas em camadas. As camadas são restauradas em paralelo pelo Docker utilizando três threads simultâneas. Você pode aumentar o número de segmentos concomitantes utilizando o parâmetro `max-concurrent-downloads`. Para obter mais informações, consulte [Documentação do Docker](#).

Embora você possa utilizar contêineres maiores, recomendamos otimizar a estrutura e o tamanho do contêiner para tempos de startup mais rápidos.

- Contêineres menores são buscados mais rapidamente: Contêineres menores podem levar a tempos de início de aplicativos mais rápidos. Para diminuir o tamanho do contêiner, descarregue bibliotecas ou arquivos atualizados com pouca frequência para a imagem de máquina da Amazon (AMI). Você também pode associar montagens para dar acesso aos seus contêineres. Para obter mais informações, consulte [Associando Montagens](#).
- Crie camadas de tamanho uniformes em tamanho e quebre camadas grandes: Cada camada é restaurada por uma thread. Portanto, uma camada grande pode impactar significativamente seu tempo de startup de trabalho. É recomendado um tamanho máximo de camada de 2 GB como boa concessão entre um tamanho maior do contêiner e tempos de startup mais rápidos. Você pode executar o comando `docker history your_image_id` para verificar sua estrutura de imagens de contêiner e o tamanho da camada. Para obter mais informações, consulte [Documentação do Docker](#).
- Use o Amazon Elastic Container Registry como seu repositório de contêineres: Quando você executa milhares de trabalhos em paralelo, um repositório autogerenciado pode falhar ou controlar a utilização do throughput. O Amazon ECR funciona em escala e processa workloads de até um milhão de vCPUs.

Using machine images and containers



Escolha a opção direita de recurso de ambiente de computação

AWS Fargate exige menos definições e configurações iniciais do que o Amazon EC2 e provavelmente é mais fácil de utilizar, especialmente se for sua primeira vez. Com o Fargate, você não precisa gerenciar servidores, processar o planejamento de capacidade ou isolar workloads de contêiner para segurança.

Se você têm os seguintes requisitos, recomendamos que use instâncias do Fargate:

- seus trabalhos devem iniciar rapidamente, em menos de 30 segundos.
- Os requisitos dos seus trabalhos são 16 vCPUs ou menos, sem GPUs, e 120 GiB de memória ou menos.

Para obter mais informações, consulte [Quando usar o Fargate](#).

se você têm os seguintes requisitos, recomendamos que utilize instâncias Amazon EC2:

- Caso exija controle aumentado sobre a seleção da instância, ou exige a utilização de tipos de instância específicos.
- Seus trabalhos exigem recursos que o AWS Fargate não pode fornecer, como GPUs, mais memória, uma AMI personalizada, ou o Amazon Elastic Fabric Adapter.
- Caso exija um alto nível de throughput ou simultaneidade.

- Caso precise personalizar sua AML, acesse Modelos de Inicialização do Amazon EC2 ou acesse parâmetros especiais do Linux.

Com o Amazon EC2, você pode ajustar com maior precisão sua workload de acordo com seus requisitos específicos e executá-la em escala, se necessário.

Amazon EC2 Sob Demanda, ou Amazon EC2 Spot

A maioria dos clientes AWS Batch utiliza instâncias spot do Amazon EC2 devido a economia em relação às instâncias sob demanda. No entanto, se sua workload for executada por várias horas e não puder ser interrompida, as instâncias sob demanda podem ser mais adequadas para você. Você sempre pode experimentar as instâncias spot primeiro e mudar para sob demanda, se necessário.

Caso tenha os seguintes requisitos e expectativas, utilize as instâncias sob demanda do Amazon EC2:

- quando o runtime dos seus trabalhos for maior que uma hora e você não possa tolerar interrupções na workload.
- Você tem um SLO (objetivo de nível de serviço) rigoroso para sua workload geral e não pode aumentar o tempo de computação.
- As instâncias requeridas têm maior probabilidade de sofrer interrupções.

Caso tenha os seguintes requisitos e expectativas, utilize as instâncias spot do Amazon EC2:

- quando o runtime usual de seus trabalhos for de até 30 minutos.
- Você pode tolerar potenciais interrupções e reagendamento de tarefas como parte da workload. Para obter mais informações, consulte [Advisor de Instâncias Spot](#).
- Tarefas em execução longas podem ser reiniciadas a partir de um ponto de verificação, caso sejam interrompidas.

Você pode misturar os dois modelos de compra enviando na instância spot primeiro e depois utilizando a Instância Sob Demanda como opção fallback. Por exemplo, insira seus trabalhos em uma fila conectada a ambientes computacionais em execução em instâncias spot do Amazon EC2. Se um trabalho for interrompido, capture o evento de Amazon EventBridge e correlacione-o a uma recuperação de instância Spot. Em seguida, reenvie o trabalho para uma fila sob demanda utilizando uma função AWS Lambda ou AWS Step Functions. Para obter mais informações, consulte [Tutorial](#):

[Como enviar alertas do Amazon Simple Notification Service para eventos de trabalho que falharam](#), [Práticas Recomendadas para Processar Interrupções da Instância Spot do Amazon EC2](#) e [Gerenciar AWS Batch com Step Functions](#).

 Important

Utilize diferentes tipos, tamanhos e Zonas de Disponibilidade de instâncias para seu ambiente de computação sob demanda para manter a disponibilidade dos grupos de instâncias spot do Amazon EC2 e diminuir a taxa de interrupção.

Use as práticas recomendadas do Amazon EC2 Spot para AWS Batch

Quando você escolhe as instâncias spot do Amazon Elastic Compute Cloud (EC2), é provável que você possa otimizar seu fluxo de trabalho para economizar custos - às vezes, de forma significativa. Para obter mais informações, consulte [Práticas Recomendadas para o Amazon EC2 Spot](#).

Para otimizar seu fluxo de trabalho e economizar custos, considere as seguintes práticas recomendadas do Amazon EC2 Spot para AWS Batch:

- Escolher a estratégia de alocação **SPOT_CAPACITY_OPTIMIZED**: AWS Batch escolhe instâncias do Amazon EC2 a partir dos grupos de capacidade spot mais profundos do Amazon EC2. Se estiver preocupado com interrupções, essa é uma escolha adequada. Para obter mais informações, consulte [Estratégias de alocação de tipos de instâncias para o AWS Batch](#).
- Diversifique os tipos de instância: Para diversificar seus tipos de instância, considere tamanhos e famílias compatíveis e, em seguida, deixe AWS Batch escolher com base em preço ou disponibilidade. Por exemplo, considere c5.24xlarge como uma alternativa para c5.12xlarge ou famílias c5a, c5n, c5d, m5 e m5d. Para obter mais informações, consulte [Seja Flexível sobre Tipos de Instância e Zonas de Disponibilidade](#).
- Reduza o runtime do trabalho ou o ponto de verificação: Desaconselhamos executar tarefas que levem uma hora ou mais ao utilizar instâncias spot do Amazon EC2 para evitar interrupções. Caso divida ou verifique seus trabalhos em partes menores que consistam em 30 minutos ou menos, você pode reduzir significativamente a possibilidade de interrupções.
- Utilize repetições automatizadas: Para evitar interrupções em configurações de trabalhos AWS Batch, defina repetições automatizadas para trabalhos. Os trabalhos em lote podem ser

interrompidos por qualquer um dos seguintes motivos: um código de saída retornado diferente de zero; um erro de serviço ocorrido; ou uma recuperação de instância ocorrida. Você pode configurar até 10 repetições automatizadas. De início, é recomendado definir pelo menos de 1 a 3 repetições automatizadas. Para obter informações sobre o monitoramento de interrupções spot do Amazon EC2, consulte [Painel de Interrupções de Spot](#).

No AWS Batch, se você definir o parâmetro de repetição, o trabalho será colocado na frente da fila de trabalhos. Ou seja, o trabalho ganha prioridade. Ao criar a definição de trabalho ou inseri-lo no AWS CLI, você pode configurar uma estratégia de repetição. Para obter mais informações, consulte [submit-job](#).

```
$ aws batch submit-job --job-name MyJob \  
  --job-queue MyJQ \  
  --job-definition MyJD \  
  --retry-strategy attempts=2
```

- Utilize novas tentativas personalizadas: Você pode configurar uma estratégia de repetição de trabalho para um código de saída de aplicativo específico ou recuperação de instância. No exemplo a seguir, caso o host cause a falha, o trabalho poderá ser repetido até cinco vezes. No entanto, trabalhos com falha por um motivo diferente serão encerrados, e o status será definido como FAILED.

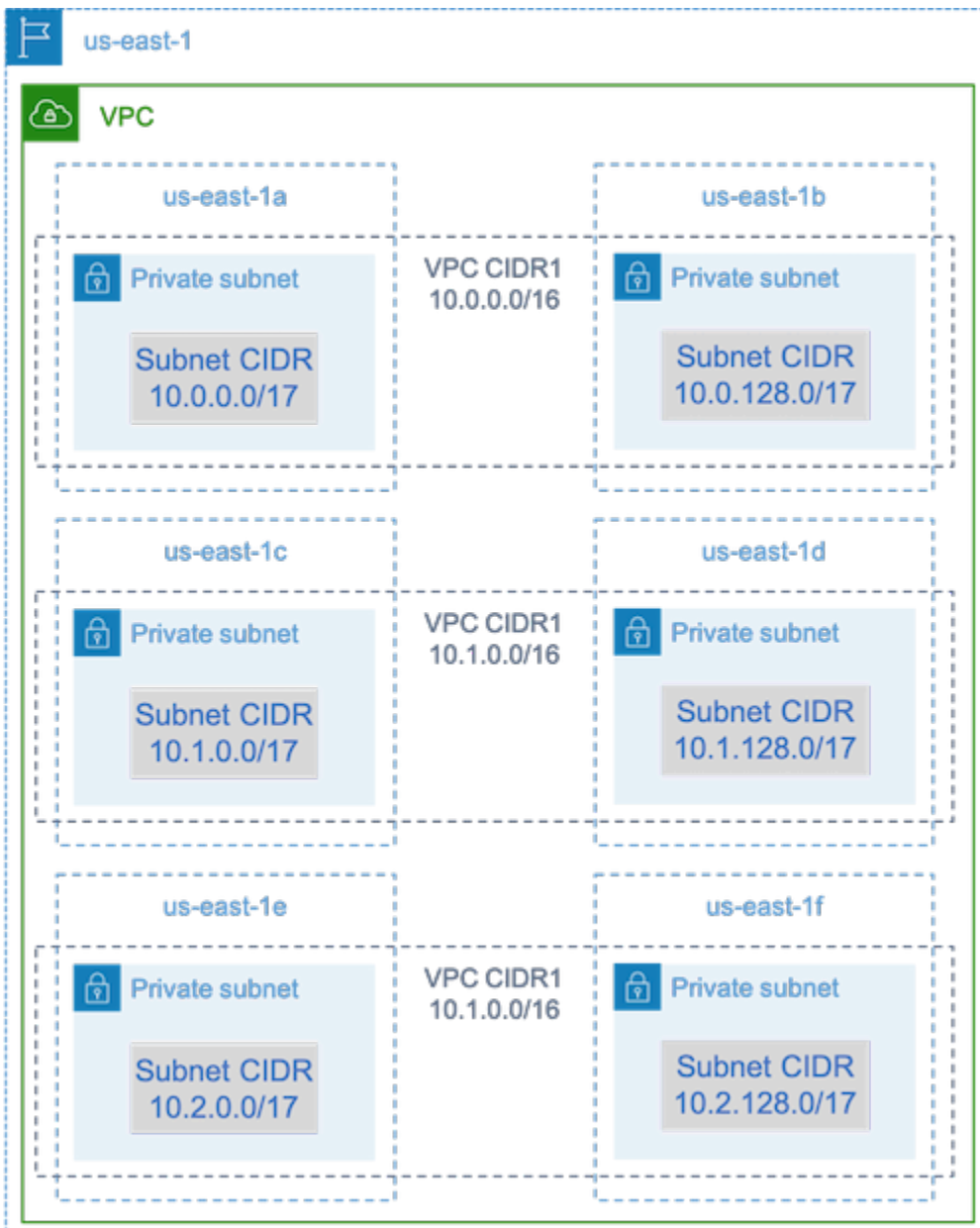
```
"retryStrategy": {  
  "attempts": 5,  
  "evaluateOnExit":  
    [{  
      "onStatusReason" : "Host EC2*",  
      "action": "RETRY"  
    }, {  
      "onReason" : "*",  
      "action": "EXIT"  
    }]  
}
```

- Use o Painel de Interrupções Spot: Você pode usar o Painel de Interrupções Spot para monitorar interrupções de Spot. O aplicativo fornece métricas sobre instâncias spot do Amazon EC2 assumidas de novo e em cujas Zonas de Disponibilidade as instâncias spot estejam. Para obter mais informações, consulte [Painel de Interrupções Spot](#)

Erros Comuns e Solução de Problemas

Erros em AWS Batch ocorrem frequentemente no nível do aplicativo, ou são ocasionados por configurações de instância que não estejam em conformidade com os requisitos específicos obrigatórios do trabalho. Outros problemas incluem trabalhos presos no status `RUNNABLE` ou ambientes computacionais presos em um estado `INVALID`. Para obter mais informações sobre soluções de problemas de trabalhos presos no status `RUNNABLE`, consulte [Trabalhos presos no status `RUNNABLE`](#). Para obter informações sobre solução de problemas em ambientes de computação em um estado `INVALID`, consulte [Ambiente de computação do `INVALID`](#).

- Verifique as Cotas vCPU do Amazon EC2 Spot: Verifique se suas cotas de serviço atuais atendem aos requisitos do trabalho. Por exemplo, ao supor que sua cota de serviço atual seja de 256 vCPUs e que o trabalho exija 10.000 vCPUs. Nesse caso, a cota de serviço não está em conformidade com os requisitos do trabalho. Para obter mais informações e instruções sobre solução de problemas, consulte [cotas de serviço do Amazon EC2](#) e [Como aumento a cota de serviços dos meus recursos do Amazon EC2?](#).
- Trabalhos com falha antes da execução do aplicativo: Alguns trabalhos podem falhar devido a um erro `DockerTimeoutError` ou a um erro `CannotPullContainerError`. Para obter informações sobre solução de problemas, consulte [Como resolver o erro “`DockerTimeoutError`” em AWS Batch](#).
- Endereços IP insuficientes: O número de endereçamento de IP na sua VPC e sub redes pode limitar o número de instâncias que você pode criar. Use Encaminhamento Entre Domínios Sem Classificação (CIDR) para fornecer mais endereços IP do que o obrigatório para a execução de suas workloads. Se necessário, você também pode compilar uma VPC dedicada com grande espaço de endereço. Por exemplo, você pode criar uma VPC com vários CIDRs em `10.x.0.0/16` e uma sub-rede em cada zona de disponibilidade com um CIDR de `10.x.y.0/17`. Neste exemplo, x está entre 1-4 e y é 0 ou 128. Essa configuração fornece 36.000 endereços IP em cada sub-rede.



- Verifique se as instâncias estão registradas no Amazon EC2: Se você vê suas instâncias no console do Amazon EC2 mas não vê nenhuma instância de contêiner do Amazon Elastic Container Service em seu cluster do Amazon ECS, o atendente do Amazon ECS pode não estar instalado em uma imagem de máquina da Amazon (AMI). O atendente do Amazon ECS, a entrada de dados do Amazon EC2 em sua AM, ou o modelo de inicialização também podem não estar configurados corretamente. Para isolar a causa raiz, crie uma instância separada do Amazon EC2 ou conecte-se a uma instância existente usando SSH. Para obter mais informações, consulte [Configuração do Agente de Contêiner do Amazon ECS](#), [Locais de Arquivo de Log do Amazon ECS](#) e [Recursos de computação de AMIs](#).

- Analise o painel AWS: Analise o Painel AWS para verificar se os estados esperados da tarefa e se o ambiente computacional estão em escala conforme o esperado. Você também pode analisar os logs de trabalho no CloudWatch.
- Verifique se sua instância foi criada: Se uma instância tiver sido criada, significa que seu ambiente de computação foi escalado conforme o esperado. Se suas instâncias não foram criadas, descubra as sub-redes associadas em seu ambiente de computação para alterá-las. Para obter mais informações, consulte [Verificar uma Ação em Escala para um Grupo do Auto Scaling](#).

Também recomendamos que você verifique se suas instâncias podem atender aos requisitos de trabalho relacionados. Por exemplo, um trabalho pode exigir 1 TiB de memória, mas o ambiente de computação utiliza um tipo de instância C5 limitado a 192 GB de memória.

- Verifique se suas instâncias estão sendo solicitadas por AWS Batch: Verifique o histórico do grupo do Auto Scaling para ter certeza de que se suas instâncias estão sendo solicitadas por AWS Batch. Esse é um indicador de como o Amazon EC2 tenta adquirir instâncias. Se você receber um estado de erro informando que o Amazon EC2 Spot não pode adquirir uma instância em uma Zona de Disponibilidade específica, talvez esta Zona de Disponibilidade não ofereça uma família de instâncias específica.
- Verifique se as instâncias estão registradas no Amazon ECS: Caso você veja suas instâncias no console do Amazon EC2, mas não veja nenhuma instância de contêiner do Amazon ECS em seu cluster do Amazon ECS, o atendente do Amazon ECS pode não estar instalado em uma imagem de máquina da Amazon (AMI). Além disso, o atendente do Amazon ECS, os Dados Amazon EC2 em sua AMI, ou o modelo de inicialização podem não estar configurados corretamente. Para isolar a causa raiz, crie uma instância separada do Amazon EC2 ou conecte-se a uma instância existente usando SSH. Para obter mais informações, consulte [Arquivo de Configuração do Atendente do CloudWatch: Seção Logs](#), [Locais de Arquivos de Log do Amazon ECS](#) e [Recursos de computação de AMIs](#).
- Abra um tíquete de suporte: Se ainda estiver enfrentando problemas após uma solução de problemas e tiver um Plano do Support, abra um tíquete de suporte. No tíquete de suporte, certifique-se de incluir informações sobre o problema, as especificações da workload, a configuração e os resultados do teste. Para obter mais informações, consulte [Comparar Planos Suporte](#).
- Analise os fóruns de AWS Batch e HPC: para obter mais informações, consulte os fóruns [AWS Batch](#) e [HPC](#).
- Analise o Painel de Monitoramento de Runtime AWS Batch: Esse painel usa uma arquitetura de tecnologia sem servidor para capturar eventos do Amazon ECS, de AWS Batch, e do Amazon

EC2, fornecer insights sobre trabalhos e instâncias. Para obter mais informações, consulte [Solução de Painéis de Monitoramento de Runtime AWS Batch](#).

Solução de problemas AWS Batch

Você talvez precise resolver problemas relacionados a ambientes de computação, filas de trabalho, definições de trabalho ou trabalhos. Este capítulo descreve como solucionar e resolver esses problemas em seu AWS Batch ambiente.

AWS Batch usa políticas, funções e permissões do IAM e é executado na infraestrutura Amazon EC2, Amazon ECS e Amazon Elastic Kubernetes Service. Para solucionar problemas relacionados a esses serviços, consulte o seguinte:

- [Solução de problemas do IAM](#) no Guia do Usuário do IAM
- [Solução de problema do Amazon ECS](#) no Guia do Desenvolvedor do Amazon Elastic Container Service
- [Solução de problemas do Amazon EKS](#) no Guia do Usuário do Amazon EKS
- [Solucione problemas de EC2 instâncias](#) no Guia do EC2 usuário da Amazon

Sumário

- [AWS Batch](#)
 - [Configuração ideal do tipo de instância para receber atualizações automáticas da família de instâncias](#)
 - [Ambiente de computação do INVALID](#)
 - [Nome do perfil ou ARN incorreto](#)
 - [Reparar um ambiente de computação INVALID](#)
 - [Trabalhos presos no status RUNNABLE](#)
 - [Instâncias spot sem tags na criação](#)
 - [As instâncias spot não estão diminuindo](#)
 - [Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil da Frota Spot no Console de gerenciamento da AWS](#)
 - [Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil de Frota Spot com o AWS CLI](#)
 - [Não é possível recuperar segredos do Secrets Manager](#)
 - [Não é possível substituir os requisitos de recursos de definição de trabalho](#)
 - [Mensagem de erro ao atualizar a configuração desiredvCpus](#)

- [AWS Batch no Amazon EKS](#)
 - [Ambiente de computação do INVALID](#)
 - [Versão Kubernetes sem suporte](#)
 - [O perfil da instância não existe](#)
 - [Namespace Kubernetes inválido](#)
 - [Ambiente de computação gerenciado](#)
 - [Os nós não se juntam ao cluster Amazon EKS](#)
 - [AWS Batch no Amazon EKS, o trabalho está preso no status RUNNABLE](#)
 - [AWS Batch no Amazon EKS, o trabalho está preso no status STARTING](#)
 - [Cenário: falha persistente na conexão ou na montagem da solicitação de volume](#)
 - [Verifique se o aws-auth ConfigMap está configurado corretamente](#)
 - [As permissões ou vinculações do RBAC não estão configuradas corretamente](#)

AWS Batch

Analise os tópicos a seguir para encontrar processos de revisão e possíveis soluções para problemas comuns que você pode encontrar ao usar o AWS Batch.

Tópicos

- [Configuração ideal do tipo de instância para receber atualizações automáticas da família de instâncias](#)
- [Ambiente de computação do INVALID](#)
- [Trabalhos presos no status RUNNABLE](#)
- [Instâncias spot sem tags na criação](#)
- [As instâncias spot não estão diminuindo](#)
- [Não é possível recuperar segredos do Secrets Manager](#)
- [Não é possível substituir os requisitos de recursos de definição de trabalho](#)
- [Mensagem de erro ao atualizar a configuração desiredvCpus](#)

Configuração ideal do tipo de instância para receber atualizações automáticas da família de instâncias

Note

A partir de 01/11/2025, o comportamento do `optimal` será alterado para corresponder com `default_x86_64`. Durante a mudança, suas famílias de instâncias podem ser atualizadas para uma nova geração. Não há necessidade de realizar nenhuma ação para que a atualização ocorra.

AWS Batch suportou uma única opção em `instanceTypes` `optimal` para atender à demanda de suas filas de trabalho. Introduzimos duas novas opções de tipo de instância: `default_x86_64` e `default_arm64`. Usaremos `default_x86_64` se você não fizer nenhuma seleção do tipo de instância. Essas novas opções selecionarão automaticamente tipos de instância econômicos em diferentes famílias e gerações com base nos requisitos da fila de trabalhos, permitindo que você execute seus workloads rapidamente.

À medida que a capacidade suficiente de novos tipos de instância for disponibilizada em um Região da AWS, o pool padrão correspondente será atualizado automaticamente com o novo tipo de instância. A opção `optimal` existente continuará sendo compatível e não está sendo descontinuada, pois será compatível com os pools padrão subjacentes para fornecer instâncias atualizadas daqui para frente. Se você estiver usando `optimal`, não há necessidade de realizar nenhuma ação.

No entanto, esteja ciente de que somente `ENABLED` e `VALID` Compute Environments (CEs) serão atualizados automaticamente com novos tipos de instância. Se você tiver algum `DISABLED` ou `INVALID` CEs, eles receberão atualizações assim que forem reativados e definidos para um `VALID` estado.

Ambiente de computação do **INVALID**

É possível que você tenha configurado incorretamente um ambiente de computação gerenciado. Se você fez isso, o ambiente de computação entrará em um estado `INVALID` e não poderá aceitar trabalhos para colocação. As seções a seguir descrevem as possíveis causas e como solucionar problemas com base na causa.

 Important

AWS Batch cria e gerencia vários AWS recursos em seu nome e dentro da sua conta, incluindo Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets e Amazon ECS Clusters. Esses recursos gerenciados são configurados especificamente para garantir a operação ideal do AWS Batch. A modificação manual desses recursos gerenciados em lote, a menos que seja explicitamente declarado na AWS Batch documentação, pode resultar em comportamento inesperado, resultando em ambiente de INVALID computação, comportamento de escalabilidade de instâncias abaixo do ideal, atraso no processamento do workload ou custos inesperados. Essas modificações manuais não podem ser sustentadas de forma determinística pelo AWS Batch serviço. Sempre use o Batch compatível APIs ou o console Batch para gerenciar seus ambientes de computação.

Nome do perfil ou ARN incorreto

A causa mais comum de um ambiente computacional entrar em um INVALID estado é que a função de AWS Batch serviço ou a função Amazon EC2 Spot Fleet tem um nome ou nome de recurso da Amazon (ARN) incorreto. Isso é mais comum em ambientes computacionais criados usando o AWS CLI ou o AWS SDKs. Quando você cria um ambiente computacional no Console de gerenciamento da AWS, AWS Batch ajuda você a escolher o serviço correto ou as funções do Spot Fleet. No entanto, suponha que você insira manualmente o nome ou o ARN e os insira incorretamente. Então, o ambiente de computação resultante também é INVALID.

No entanto, suponha que você insira manualmente o nome ou o ARN de um recurso do IAM em um AWS CLI comando ou no código do SDK. Nesse caso, não é AWS Batch possível validar a string. Em vez disso, AWS Batch deve aceitar o valor ruim e tentar criar o ambiente. Se AWS Batch não conseguir criar o ambiente, o ambiente será movido para um INVALID estado e você verá os seguintes erros.

Para um perfil de serviços inválida:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

Para um perfil de Spot Fleet inválida:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Uma causa comum para esse problema é o cenário a seguir. Você só especifica o nome de uma função do IAM ao usar o AWS CLI ou o AWS SDKs, em vez do Amazon Resource Name (ARN) completo. Dependendo de como você criou o perfil, o ARN poderá conter o prefixo de caminho `aws-service-role`. Por exemplo, se você criar manualmente o perfil de serviço do AWS Batch usando os procedimentos em [Usar perfis vinculados a serviço do AWS Batch](#), o ARN do perfil de serviço deverá ser semelhante ao seguinte.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

No entanto, se você criar o perfil de serviço como parte da primeira execução do assistente do console hoje, o ARN do perfil de serviço terá a seguinte aparência.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Esse problema também pode ocorrer se você anexar a política de AWS Batch nível de serviço (`AWSBatchServiceRole`) a uma função que não seja de serviço. Por exemplo, você pode receber uma mensagem de erro semelhante à seguinte neste cenário:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
authorized to perform: action on resource ...
```

Para resolver esse problema, execute um dos seguintes procedimentos:

- Use uma string vazia para a função de serviço ao criar o ambiente AWS Batch computacional.
- Especifique o perfil de serviço no seguinte formato: `arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Quando você especifica apenas o nome de uma função do IAM ao usar o AWS CLI ou o AWS SDKs, AWS Batch presume que seu ARN não usa `aws-service-role` o prefixo do caminho. Por isso, recomendamos que você especifique o ARN completo para seu perfil do IAM; ao criar ambientes de computação.

Para reparar um ambiente de computação que é configurado incorretamente dessa forma, consulte [Reparar um ambiente de computação INVALID](#).

Reparar um ambiente de computação **INVALID**

Quando você tem um ambiente de computação em um estado **INVALID**, deve atualizá-lo para reparar o parâmetro inválido. Para um [Nome do perfil ou ARN incorreto](#), você pode atualizar o ambiente de computação com o perfil de serviço correto.

Para reparar um ambiente de computação configurado incorretamente

1. Abra o AWS Batch console em <https://console.aws.amazon.com/batch/>.
2. Na barra de navegação, selecione o Região da AWS a ser usado.
3. No painel de navegação, escolha Ambientes de computação.
4. Na página Ambientes de computação, selecione o botão próximo ao ambiente de computação a ser editado e, em seguida, escolha Edit.
5. Na página Atualizar ambiente de computação, para Perfil de serviço, escolha perfil do IAM a ser usado com seu ambiente de computação. O console AWS Batch exibe somente perfis que têm a relação de confiança correta para ambientes de computação.

Tip

Para obter instruções sobre como criar um perfil vinculado a um serviço, consulte [Usando funções para AWS Batch](#).

6. Escolha Salvar para atualizar seu ambiente de computação.

Trabalhos presos no status **RUNNABLE**

Suponha que seu ambiente de computação contenha recursos computacionais, mas seus trabalhos não progridam além do status **RUNNABLE**. Então, é provável que algo esteja impedindo que os trabalhos sejam colocados em um recurso de computação e fazendo com que suas filas de trabalho sejam bloqueadas. Veja como saber se o seu trabalho está aguardando a vez dele ou se está preso e bloqueando a fila.

Se AWS Batch detectar que você tem um **RUNNABLE** emprego como chefe e está bloqueando a fila, você receberá um [Eventos bloqueados da fila de trabalhos](#) evento da Amazon CloudWatch Events

com o motivo. O mesmo motivo também é atualizado no campo `statusReason` como parte das chamadas de API [ListJobs](#) e [DescribeJobs](#).

Opcionalmente, você pode configurar o parâmetro `jobStateTimeLimitActions` por meio das ações de API [CreateJobQueue](#) e [UpdateJobQueue](#).

 Note

Atualmente, a única ação que você pode usar com `jobStateLimitActions.action` é cancelar um trabalho.

O `jobStateTimeLimitActions` parâmetro é usado para especificar um conjunto de ações que são AWS Batch executadas em trabalhos em um estado específico. É possível definir um limite de tempo em segundos por meio do campo `maxTimeSeconds`.

Quando um trabalho está em um `RUNNABLE` estado com o `definidoStatusReason`, AWS Batch executa a ação especificada após `maxTimeSeconds` ter decorrido.

Por exemplo, você pode definir o parâmetro `jobStateTimeLimitActions` para aguardar até 4 horas por qualquer trabalho no estado `RUNNABLE` que esteja aguardando a disponibilidade de capacidade suficiente. É possível fazer isso configurando `statusReason` a `maxTimeSeconds` e `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` como 144000 antes de cancelar o trabalho e permitir que o próximo trabalho avance para o início da fila de trabalhos.

A seguir estão os motivos que ocorrem AWS Batch quando ele detecta que uma fila de trabalhos está bloqueada. Esta lista fornece as mensagens retornadas das ações de API `ListJobs` e `DescribeJobs`. Também são os mesmos valores que podem ser definidos para o parâmetro `jobStateLimitActions.statusReason`.

1. Motivo: todos os ambientes de computação conectados têm erros de capacidade insuficiente. Quando solicitado, AWS Batch detecta EC2 instâncias da Amazon que apresentam erros de capacidade insuficientes. O cancelamento manual do trabalho permitirá que o trabalho subsequente seja transferido para o início da fila mas, sem resolver o(s) problema(s) do perfil de serviço, é provável que o próximo trabalho também seja bloqueado. É melhor investigar e resolver esse problema manualmente.
 - Mensagem **statusReason** enquanto o trabalho está travado:
`CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]`

- **reason** usado para **jobStateTimeLimitActions**:
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY
- Mensagem **statusReason** depois que o trabalho é cancelado:
Canceled by JobStateTimeLimit action due to reason:
CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY

Observações:

- a. A função AWS Batch de serviço requer `autoscaling:DescribeScalingActivities` permissão para que essa detecção funcione. Se você usar o perfil vinculado a serviço (SLR) [Usar perfis vinculados a serviço do AWS Batch](#) ou a política gerenciada [AWS política gerenciada: AWSBatchServiceRolepolítica](#), não precisará tomar nenhuma medida, pois as políticas de permissão serão atualizadas.
 - b. Se você usar a SLR ou a política gerenciada, deverá adicionar as permissões `autoscaling:DescribeScalingActivities` e `ec2:DescribeSpotFleetRequestHistory` para poder receber eventos de fila de trabalho bloqueada e status de trabalho atualizado quando estiver em `RUNNABLE`. Além disso, o AWS Batch precisa dessas permissões para executar ações do `cancellation` por meio do parâmetro `jobStateTimeLimitActions`, mesmo que elas estejam configuradas na fila de trabalho.
 - c. No caso de um trabalho paralelo de vários nós (MNP), se o ambiente EC2 computacional da Amazon anexado de alta prioridade apresentar `insufficient capacity` erros, ele bloqueará a fila mesmo que um ambiente computacional de prioridade mais baixa apresente esse erro.
2. Motivo: todos os ambientes de computação têm um parâmetro [maxvCpus](#) menor do que os requisitos do trabalho. O cancelamento do trabalho, seja manualmente ou definindo o parâmetro `jobStateTimeLimitActions` em `statusReason`, permite que o trabalho subsequente seja movido para o topo da fila. Como opção, é possível aumentar o parâmetro `maxvCpus` do ambiente de computação primário para atender às necessidades do trabalho bloqueado.
 - Mensagem **statusReason** enquanto o trabalho está travado:
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.
 - **reason** usado para **jobStateTimeLimitActions**:
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE

- Mensagem **statusReason** depois que o trabalho é cancelado:
Canceled by JobStateTimeLimit action due to reason:
MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE
3. Motivo: nenhum dos ambientes de computação tem instâncias que atendem aos requisitos do trabalho. Quando um trabalho solicita recursos, AWS Batch detecta que nenhum ambiente computacional conectado é capaz de acomodar o trabalho recebido. O cancelamento do trabalho, seja manualmente ou definindo o parâmetro `jobStateTimeLimitActions` em `statusReason`, permite que o trabalho subsequente seja movido para o topo da fila. Como opção, é possível redefinir os tipos de instância permitidos do ambiente de computação para adicionar os recursos de trabalho necessários.
- Mensagem **statusReason** enquanto o trabalho está travado:
MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.
 - **reason** usado para **jobStateTimeLimitActions**:
MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT
 - Mensagem **statusReason** depois que o trabalho é cancelado:
Canceled by JobStateTimeLimit action due to reason:
MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT
4. Motivo: todos os ambientes de computação têm problemas de perfil de serviço. Para resolver isso, compare suas permissões de perfil de serviço com o [AWS políticas gerenciadas para AWS Batch](#) e resolva as divergências. Observação: não é possível configurar uma ação programável por meio do parâmetro `jobStateTimeLimitActions` para solucionar esse erro.

É uma prática recomendada usar o [Usar perfis vinculados a serviço do AWS Batch](#) para evitar erros semelhantes.

O cancelamento do trabalho, seja manualmente ou definindo o parâmetro `jobStateTimeLimitActions` em `statusReason`, permite que o trabalho subsequente seja movido para o topo da fila. Sem resolver o(s) problema(s) do perfil de serviço, é provável que o próximo trabalho também seja bloqueado. É melhor investigar e resolver esse problema manualmente.

- Mensagem **statusReason** enquanto o trabalho está travado:
MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS - Batch service role has a permission issue.

5. Motivo: seu ambiente computacional tem uma configuração de tipo de instância incompatível. Isso pode ocorrer quando os tipos de instância não estão disponíveis nas zonas de disponibilidade selecionadas ou quando seu modelo de execução ou configuração de execução contém configurações incompatíveis com os tipos de instância especificados. Para resolver isso, verifique se seus tipos de instância são compatíveis com suas zonas especificadas Região da AWS e de disponibilidade, verifique se as configurações do modelo de execução são compatíveis com seus tipos de instância e considere atualizar para tipos de instância de geração mais recente. Para obter mais informações sobre como encontrar tipos de instância compatíveis, consulte [Como encontrar um tipo de EC2 instância](#) da Amazon no Guia EC2 do usuário da Amazon.
- Mensagem **statusReason** enquanto o trabalho está travado:
MISCONFIGURATION:EC2_INSTANCE_CONFIGURATION_UNSUPPORTED - Your compute environment associated with this job queue has an unsupported instance type configuration.
6. Motivo: todos os ambientes de computação são inválidos. Para obter mais informações, consulte [Ambiente de computação do INVALID](#). Observação: não é possível configurar uma ação programável por meio do parâmetro `jobStateTimeLimitActions` para solucionar esse erro.
- Mensagem **statusReason** enquanto o trabalho está travado: ACTION_REQUIRED - CE(s) associated with the job queue are invalid.
7. Motivo: AWS Batch detectou uma fila bloqueada, mas não consegue determinar o motivo. Observação: não é possível configurar uma ação programável por meio do parâmetro `jobStateTimeLimitActions` para solucionar esse erro. Para obter mais informações sobre a solução de problemas, consulte [Por que meu trabalho do AWS Batch está preso em RUNNABLE na AWS](#), no re:Post.
- Mensagem **statusReason** enquanto o trabalho está travado: UNDETERMINED - Batch job is blocked, root cause is undetermined.

Caso você não tenha recebido um evento da CloudWatch Events ou tenha recebido o evento de motivo desconhecido, aqui estão algumas causas comuns desse problema.

O driver de log **awslogs** não está configurado em seus recursos de computação

AWS Batch os jobs enviam suas informações de registro para o CloudWatch Logs. Para ativar, você deve configurar os recursos de computação para usar o driver de log **awslogs**. Suponha que você baseie sua AMI de recursos de computação na AMI otimizada do Amazon ECS (ou Amazon Linux). Em seguida, esse driver é registrado por padrão com o `ecs-init` pacote. Agora, suponha que você use uma AMI base diferente. Em seguida, você deve verificar que o driver

de log `awslogs` seja especificado como um driver de log disponível com a variável de ambiente `ECS_AVAILABLE_LOGGING_DRIVERS` quando o agente de contêiner do Amazon ECS for iniciado. Para obter mais informações, consulte [Especificação da AMI do recurso de computação](#) e [Tutorial: criar uma AMI de recurso de computação](#).

Recursos insuficientes

Se suas definições de trabalho especificarem mais recursos de CPU ou memória do que seus recursos de computação podem alocar, seus trabalhos nunca serão colocados. Por exemplo, suponha que seu trabalho especifique 4 GiB de memória e que seus recursos de computação tenham menos do que os disponíveis. Então, acontece que o trabalho não pode ser colocado nesses recursos de computação. Nesse caso, você deve reduzir a memória especificada em sua definição de trabalho ou adicionar mais recursos de computação ao seu ambiente. Alguma memória é reservada para o agente de contêiner do Amazon ECS e outros processos críticos do sistema. Para obter mais informações, consulte [Gerenciamento de memória de recursos de computação](#).

Sem acesso à Internet para os recursos de computação

Recursos de computação precisam de acesso para se comunicar com o endpoint de serviço do Amazon ECS. Isso pode ser feito por meio de uma interface do endpoint da VPC ou por meio dos das instâncias de contêiner que tenham endereços IP públicos.

Para obter mais informações sobre endpoints da VPC de interface, consulte [Endpoints da VPC de interface do Amazon ECS \(AWS PrivateLink\)](#) no Manual do Desenvolvedor do Amazon Elastic Container Service.

Se você não tiver um endpoint da VPC de interface configurado e seus das instâncias de contêiner não tiverem endereços IP públicos, eles deverão usar a conversão de endereço de rede (NAT) para fornecer esse acesso. Para obter mais informações, consulte [Gateways NAT](#) no Guia do usuário da Amazon VPC. Para obter mais informações, consulte [the section called “Crie uma VPC”](#).

Limite de EC2 instâncias da Amazon atingido

O número de EC2 instâncias da Amazon em que sua conta pode iniciar Região da AWS é determinado pela sua cota de EC2 instâncias. Alguns tipos de instância também têm uma per-instance-type cota. Para obter mais informações sobre a cota de EC2 instância da Amazon da sua conta, incluindo como solicitar um aumento de limite, consulte [Limites de EC2 serviços da Amazon](#) no Guia do EC2 usuário da Amazon.

O agente do contêiner do Amazon ECS não está instalado

O agente de contêiner do Amazon ECS deve ser instalado na Amazon Machine Image (AMI) para permitir a AWS Batch execução de trabalhos. O agente de contêiner do Amazon ECS é instalado por padrão no Amazon ECS otimizado. AMIs Para obter mais informações sobre o agente de contêineres do Amazon ECS, consulte [Agente de contêineres do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Para obter mais informações, consulte [Por que meu AWS Batch trabalho está preso no RUNNABLE status?](#) em re:POST.

Instâncias spot sem tags na criação

Haverá suporte para marcação em instâncias spot de recursos de computação do AWS Batch a partir de 25 de outubro de 2017. Antes, a política gerenciada de IAM recomendada (AmazonEC2SpotFleetRole) para o perfil Amazon EC2 Spot Fleet não continha permissões para marcar instâncias Spot no lançamento. A nova política gerenciada de IAM recomendada é chamada AmazonEC2SpotFleetTaggingRole. Ele suporta a marcação de instâncias spot no lançamento.

Para corrigir a marcação de Instância Spot na criação, siga o procedimento a seguir para aplicar a política gerenciada de IAM recomendada atualmente ao seu perfil do Amazon EC2 Frota Spot. Dessa forma, todas as futuras Instâncias Spot criadas com esse perfil têm permissões para aplicar tags de instância quando são criadas.

Para aplicar a política gerenciada de IAM atual ao seu perfil do Amazon EC2 Frota Spot

1. Abra o console do IAM, em <https://console.aws.amazon.com/iam/>.
2. Escolha Roles e escolha seu perfil de frota spot do seu Amazon EC2.
3. Escolha Attach policy.
4. Selecione AmazonEC2SpotFleetTaggingRole e Attach policy.
5. Escolha seu perfil Amazon EC2 Frota Spot novamente para remover a política anterior.
6. Selecione o x à direita da política AmazonEC2SpotFleetRole e escolha Detach.

As instâncias spot não estão diminuindo

AWS Batch introduziu o perfil vinculado ao serviço AWSServiceRoleForBatch em 10 de março de 2021. Se nenhum perfil for especificado no parâmetro serviceRole do ambiente de computação,

esse perfil vinculado ao serviço será usado como perfil de serviço. No entanto, suponha que o perfil vinculado ao serviço seja usado em um ambiente de computação EC2 Spot, mas o perfil spot usado não inclua a política gerenciada AmazonEC2SpotFleetTaggingRole. Então, a Instância Spot não é escala reduzida. Como resultado, você receberá uma mensagem de erro com a seguinte mensagem: “Você não está autorizado a executar esta operação”. Use as etapas a seguir para atualizar o perfil da frota spot que você usa no parâmetro spotIamFleetRole. Para obter mais informações, consulte [Using service-linked roles](#) e [Creating a role to delegate permissions to an AWS Service](#) no Guia do Usuário do IAM..

Tópicos

- [Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil da Frota Spot no Console de gerenciamento da AWS](#)
- [Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil de Frota Spot com o AWS CLI](#)

Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil da Frota Spot no Console de gerenciamento da AWS

Para aplicar a política gerenciada de IAM atual ao seu perfil do Amazon EC2 Frota Spot

1. Abra o console do IAM, em <https://console.aws.amazon.com/iam/>.
2. Escolha Roles e escolha seu perfil de frota spot do seu Amazon EC2.
3. Escolha Attach policy.
4. Selecione AmazonEC2SpotFleetTaggingRole e Attach policy.
5. Escolha seu perfil Amazon EC2 Frota Spot novamente para remover a política anterior.
6. Selecione o x à direita da política AmazonEC2SpotFleetRole e escolha Detach.

Anexe a política gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil de Frota Spot com o AWS CLI

Os comandos de exemplo pressupõem que seu perfil Amazon EC2 Frota Spot seja chamado AmazonEC2SpotFleetRole. Se seu perfil usar um nome diferente, ajuste os comandos para que correspondam.

Para anexar a política gerenciada AmazonEC2SpotFleetTaggingRole ao seu perfil Frota Spot

1. Para anexar a política do IAM gerenciada do AmazonEC2SpotFleetTaggingRole ao seu perfil *AmazonEC2SpotFleetRole*, execute o comando a seguir usando o AWS CLI.

```
$ aws iam attach-role-policy \  
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \  
    --role-name AmazonEC2SpotFleetRole
```

2. Para separar a política do IAM gerenciada do AmazonEC2SpotFleetRole do seu perfil *AmazonEC2SpotFleetRole*, execute o comando a seguir usando o AWS CLI.

```
$ aws iam detach-role-policy \  
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \  
    --role-name AmazonEC2SpotFleetRole
```

Não é possível recuperar segredos do Secrets Manager

Se você usar uma AML com um agente do Amazon ECS anterior à versão 1.16.0-1, deverá usar a variável `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` de configuração do agente do Amazon ECS para usar esse recurso. Você pode adicioná-lo ao arquivo `./etc/ecs/ecs.config` para uma nova instância de contêiner ao criar essa instância. Ou você pode adicioná-lo a uma instância existente. Se adicioná-lo a uma instância existente, reinicie o agente ECS após adicioná-lo. Para obter mais informações, consulte [Configuração do Agente de Contêineres do Amazon ECS](#) no Guia do desenvolvedor do Amazon Elastic Container Service.

Não é possível substituir os requisitos de recursos de definição de trabalho

As substituições de memória e vCPU especificadas nos membros `memory` e `vcpus` da estrutura [containerOverrides](#), que são passadas para [SubmitJob](#), não podem substituir os requisitos de memória e vCPU especificados na estrutura [resourceRequirements](#) na definição do trabalho.

Se tentar substituir esses requisitos de recursos, você poderá ver a mensagem de erro a seguir:

“Esse valor foi enviado em uma chave obsoleta e pode entrar em conflito com o valor fornecido pelos requisitos de recursos da definição de tarefa.”

Para corrigir isso, especifique os requisitos de memória e vCPU [no membro ResourceRequirements](#) do [ContainerOverrides](#). Por exemplo, se suas substituições de memória e vCPU forem especificadas nas linhas a seguir.

```
"containerOverrides": {  
  "memory": 8192,  
  "vcpus": 4  
}
```

Altere o código para o seguinte:

```
"containerOverrides": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",  
      "value": "8192"  
    },  
    {  
      "type": "VCPU",  
      "value": "4"  
    }  
  ],  
}
```

Faça a mesma alteração nos requisitos de memória e vCPU especificados no objeto [ContainerProperties](#) na definição do trabalho. Por exemplo, se seus requisitos de memória e vCPU estiverem especificados nas linhas a seguir.

```
{  
  "containerProperties": {  
    "memory": 4096,  
    "vcpus": 2,  
  }  
}
```

Altere o código para o seguinte:

```
"containerProperties": {  
  "resourceRequirements": [  
    {  
      "type": "MEMORY",
```

```
    "value": "4096"
  },
  {
    "type": "VCPU",
    "value": "2"
  }
],
}
```

Mensagem de erro ao atualizar a configuração **desiredvCpus**

Você vê a seguinte mensagem de erro ao usar a API AWS Batch para atualizar a configuração desejada de vCPUs (**desiredvCpus**).

Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to minvCpus.

Esse problema ocorre se o valor **desiredvCpus** atualizado for menor que o valor **desiredvCpus** atual. Ao atualizar o valor **desiredvCpus**, ambas as seguintes situações devem ser verdadeiras:

- O valor **desiredvCpus** deve estar entre os valores **minvCpus** e **maxvCpus**.
- O valor **desiredvCpus** atualizado deve ser maior ou igual ao valor **desiredvCpus** atual.

AWS Batch no Amazon EKS

Tópicos

- [Ambiente de computação do INVALID](#)
- [AWS Batch no Amazon EKS, o trabalho está preso no status RUNNABLE](#)
- [AWS Batch no Amazon EKS, o trabalho está preso no status STARTING](#)
- [Verifique se o aws-auth ConfigMap está configurado corretamente](#)
- [As permissões ou vinculações do RBAC não estão configuradas corretamente](#)

Analise os tópicos a seguir para encontrar processos de análise e possíveis soluções para problemas comuns que você pode encontrar ao usar o AWS Batch Amazon Elastic Kubernetes Service.

Ambiente de computação do **INVALID**

É possível que você tenha configurado incorretamente um ambiente de computação gerenciado. Se você fez isso, o ambiente de computação entrará em um estado **INVALID** e não poderá aceitar trabalhos para colocação. As seções a seguir descrevem as possíveis causas e como solucionar problemas com base na causa.

Versão Kubernetes sem suporte

Talvez você veja uma mensagem de erro semelhante à seguinte ao usar a operação de API `CreateComputeEnvironment` ou a operação de API `UpdateComputeEnvironment` para criar ou atualizar um ambiente de computação. Esse problema ocorre se você especificar uma versão Kubernetes não suportada em `EC2Configuration`.

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Para resolver esse problema, exclua o ambiente de computação e recrie-o com uma versão Kubernetes compatível.

Você pode realizar uma pequena atualização de versão no seu cluster Amazon EKS. Por exemplo, você pode atualizar o cluster de `1.xx` para `1.yy` mesmo que a versão secundária não seja compatível.

No entanto, o status do ambiente de computação pode mudar para **INVALID** depois de uma atualização de versão principal. Por exemplo, se você realizar uma atualização de versão principal de `1.xx` para `2.yy`. Se a versão principal não for compatível com AWS Batch, você verá uma mensagem de erro semelhante à seguinte.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Para resolver esse problema, especifique uma versão Kubernetes compatível ao usar uma operação de API para criar ou atualizar um ambiente computacional.

AWS Batch no Amazon EKS atualmente oferece suporte às seguintes Kubernetes versões:

- 1.34
- 1.33
- 1.32

- 1.31
- 1.30
- 1.29

O perfil da instância não existe

Se o perfil de instância especificado não existir, o status do ambiente computacional AWS Batch no Amazon EKS será alterado para `INVALID`. Você vê um erro definido no parâmetro `statusReason` semelhante ao seguinte.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Para resolver esse problema, especifique ou crie um perfil de instância em funcionamento. Para obter mais informações, consulte [função do IAM do nó do Amazon EKS](#) no Guia do usuário do Amazon EKS.

Namespace Kubernetes inválido

Se AWS Batch no Amazon EKS não puder validar o namespace para o ambiente computacional, o status do ambiente computacional será alterado para `INVALID`. Por exemplo, esse problema pode ocorrer se o namespace não existir.

Você vê uma mensagem de erro definida no parâmetro `statusReason` semelhante à seguinte.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Esse problema pode ocorrer se qualquer uma das seguintes situações for verdadeira:

- A string do namespace Kubernetes na chamada `CreateComputeEnvironment` não existe. Para obter mais informações, consulte [CreateComputeEnvironment](#).
- As permissões de controle de acesso baseado em perfil (RBAC) necessárias para gerenciar o namespace não estão configuradas corretamente.
- AWS Batch não tem acesso ao endpoint do servidor da Kubernetes API Amazon EKS.

Para resolver esse problema, consulte [Verifique se o aws-auth ConfigMap está configurado corretamente](#). Para obter mais informações, consulte [Conceitos básicos do AWS Batch no Amazon EKS](#).

Ambiente de computação gerenciado

Suponha que você exclua um cluster do Amazon EKS antes de excluir o anexo AWS Batch no ambiente computacional Amazon EKS. Em seguida, o status do ambiente computacional é alterado para `INVALID`. Nesse cenário, o ambiente computacional não funcionará adequadamente se você recriar o cluster Amazon EKS com o mesmo nome.

Para resolver esse problema, exclua e recrie o ambiente computacional AWS Batch no Amazon EKS.

Os nós não se juntam ao cluster Amazon EKS

AWS Batch no Amazon EKS, reduz a escala de um ambiente computacional se ele determinar que nem todos os nós se juntaram ao cluster do Amazon EKS. Quando o AWS Batch Amazon EKS reduz a escala do ambiente computacional, o status do ambiente computacional é alterado para `INVALID`.

Note

AWS Batch não altera o status do ambiente computacional imediatamente para que você possa depurar o problema.

Você vê uma mensagem de erro definida no parâmetro `statusReason` semelhante às seguintes:

```
Your compute environment has been INVALIDATED and scaled down because none of the instances joined the underlying ECS Cluster. Common issues preventing instances joining are the following: VPC/Subnet configuration preventing communication to ECS, incorrect Instance Profile policy preventing authorization to ECS, or customized AMI or LaunchTemplate configurations affecting ECS agent.
```

```
Your compute environment has been INVALIDATED and scaled down because none of the nodes joined the underlying Amazon EKS Cluster. Common issues preventing nodes joining are the following: networking configuration preventing communication to Amazon EKS Cluster, incorrect Amazon EKS Instance Profile or Kubernetes RBAC policy preventing authorization to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations affecting Amazon EKS/Kubernetes node bootstrap.
```

Ao usar uma AMI padrão do Amazon EKS, as causas mais comuns desse problema são as seguintes:

- O perfil da instância não está configurado corretamente. Para obter mais informações, consulte [função do IAM do nó do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- As sub-redes não estão configuradas corretamente. Para obter mais informações, consulte os [requisitos e considerações sobre VPC e sub-rede do Amazon EKS](#) no Guia do usuário do Amazon EKS.
- O grupo de segurança não está configurado corretamente. Para obter mais informações, consulte [Amazon EKS security group requirements and considerations](#) no Amazon EKS User Guide.

 Note

Você também pode ver uma notificação de erro no Personal Health Dashboard (PHD).

AWS Batch no Amazon EKS, o trabalho está preso no status **RUNNABLE**

O ConfigMap `aws-auth` é criado e aplicado automaticamente ao cluster ao criar um grupo de nós gerenciados ou ao criar um grupo de nós usando `eksctl`. Um `aws-auth` ConfigMap é criado inicialmente para permitir que os nós se juntem ao seu cluster. No entanto, você também pode usar o `aws-auth` ConfigMap para adicionar acesso com controle de acesso baseado em perfil (RBAC) para usuários e perfis.

Para verificar se o `aws-auth` ConfigMap está configurado corretamente:

1. Recupere as perfis mapeadas no: `aws-auth` ConfigMap

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Verifique se o `roleARN` está configurado da seguinte forma.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

Você também pode revisar os registros do plano de controle do Amazon EKS. Para obter mais informações, consulte [Logs do ambiente de gerenciamento do Amazon EKS](#) no Guia do usuário do Amazon EKS.

Para resolver um problema em que um trabalho está preso em um status **RUNNABLE**, recomendamos que você use `kubectl` para reaplicar o manifesto. Para obter mais informações, consulte [Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch](#). Ou, você pode usar `kubectl` para editar manualmente o `aws-auth ConfigMap`. Para obter mais informações, consulte [Habilitar o usuário do IAM e acesso ao seu cluster](#) no Amazon EKS User Guide.

AWS Batch no Amazon EKS, o trabalho está preso no status **STARTING**

Um trabalho pode permanecer no status **STARTING** quando o Pod fica preso em **PENDING** no **ContainerCreating** em qualquer solicitação de longa execução do kubelet (`pull`, `log`, `exec` e `attach`) até que o problema de inicialização do Pod seja resolvido ou o trabalho seja encerrado. Nos cenários de qualificação abaixo, o trabalho AWS Batch será encerrado em seu nome, caso contrário, o trabalho deverá ser encerrado manualmente usando a [API TerminateJob](#).

Para verificar o motivo pelo qual um trabalho pode estar preso no **STARTING**, use [Tutorial: mapear um trabalho em execução para um pod e um nó](#) para encontrar o `podName` e descreva o Pod:

```
% kubectl describe pod aws-batch.000c8190-87df-31e7-8819-176fe017a24a -n my-aws-batch-namespace
Name:          aws-batch.000c8190-87df-31e7-8819-176fe017a24a
Namespace:     my-aws-batch-namespace
...
Containers:
  default:
    ...
    State:      Waiting
    Reason:     ContainerCreating
    Ready:      False
    ...
Conditions:
  Type                               Status
  PodReadyToStartContainers         False
```



```

Initialized           True
Ready                 False
ContainersReady       False
PodScheduled          True
...
Events:
  Type      Reason      Age   From      Message
  ----      -
Warning    FailedMount  2m32s  kubelet   Unable to attach or mount volumes: ...

```

Considere configurar seu cluster EKS para [Enviar logs do ambiente de gerenciamento para o CloudWatch Logs](#) para obter visibilidade total.

Cenário: falha persistente na conexão ou na montagem da solicitação de volume

Trabalhos que usam declarações de volume persistentes em que o volume não é anexado ou montado são candidatos ao encerramento. Isso pode ser resultado de uma definição de trabalho configurada incorretamente. Consulte [Criar uma definição de trabalho de nó único nos recursos do Amazon EKS](#) para obter mais detalhes.

Verifique se o **aws-auth ConfigMap** está configurado corretamente

Para verificar se o aws-auth ConfigMap está configurado corretamente:

1. Recupere as perfis mapeadas no aws-auth ConfigMap.

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Verifique se o roleARN está configurado da seguinte forma.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

Note

O caminho `aws-service-role/batch.amazonaws.com/` foi removido do ARN so perfil vinculado ao serviço. Isso ocorre devido a um problema com o mapa de configuração aws-auth. Para obter mais informações, consulte [Perfis com caminhos não funcionam quando o caminho é incluído em seu ARN no aws-authconfigmap](#).

Note

Você também pode revisar os registros do plano de controle do Amazon EKS. Para obter mais informações, consulte [Logs do ambiente de gerenciamento do Amazon EKS](#) no Guia do usuário do Amazon EKS.

Para resolver um problema em que um trabalho está preso em um status `RUNNABLE`, recomendamos que você use `kubectl` para reaplicar o manifesto. Para obter mais informações, consulte [Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch](#). Ou, você pode usar `kubectl` para editar manualmente o `aws-auth` ConfigMap. Para obter mais informações, consulte [Habilitar o usuário e perfil do IAM e acesso ao seu cluster](#) no Amazon EKS User Guide.

As permissões ou vinculações do RBAC não estão configuradas corretamente

Se você tiver problemas de permissão ou vinculação de RBAC, verifique se o `aws-batch` Kubernetes perfil pode acessar o namespace Kubernetes:

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

Você também pode usar o comando `kubectl describe` para visualizar as autorizações para um perfil de cluster ou namespace Kubernetes.

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

O seguinte é um exemplo de saída.

```
Name:      aws-batch-cluster-role
Labels:    <none>
Annotations: <none>
PolicyRule:
  Resources                Non-Resource URLs  Resource Names
  Verbs
```

-----	-----	-----

configmaps	[]	[]
[get list watch]		
nodes	[]	[]
[get list watch]		
Pods	[]	[]
[get list watch]		
daemonsets.apps	[]	[]
[get list watch]		
deployments.apps	[]	[]
[get list watch]		
replicasets.apps	[]	[]
[get list watch]		
statefulsets.apps	[]	[]
[get list watch]		
clusterrolebindings.rbac.authorization.k8s.io	[]	[]
[get list]		
clusterroles.rbac.authorization.k8s.io	[]	[]
[get list]		
namespaces	[]	[]
[get]		
events	[]	[]
[list]		

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

O seguinte é um exemplo de saída.

Name:	aws-batch-compute-environment-role		
Labels:	<none>		
Annotations:	<none>		
PolicyRule:			
Resources	Non-Resource URLs	Resource Names	Verbs
-----	-----	-----	-----
Pods	[]	[]	[create
get list watch delete patch]			
serviceaccounts	[]	[]	[get list]
rolebindings.rbac.authorization.k8s.io	[]	[]	[get list]
roles.rbac.authorization.k8s.io	[]	[]	[get list]

Para resolver esse problema, reapply as permissões e comandos do RBAC. `rolebinding` Para obter mais informações, consulte [Etapa 2: preparar seu cluster do Amazon EKS para o AWS Batch](#).

Recursos: Service Quotas do AWS Batch

A tabela a seguir fornece as cotas de serviço para o AWS Batch que não podem ser alteradas. Cada cota é específica da região.

Recurso	Quota
Número máximo de filas de trabalho. Para obter mais informações, consulte Filas de tarefas .	50
Número máximo de ambientes de computação no Amazon ECS e no Amazon EKS. Para obter mais informações, consulte Ambientes computacionais para AWS Batch .	50
Número máximo de ambientes de computação por cluster do Amazon EKS.	5
Número máximo de ambientes de computação para cada fila de trabalho	3
Número máximo de dependências de trabalho para um trabalho	20
O tamanho máximo de definição da tarefa (para operações da API RegisterJobDefinition)	24 KiB
O tamanho máximo da carga da tarefa (para operações da API SubmitJob)	30 KiB
O tamanho máximo da matriz para trabalhos de matriz	10000
Número máximo de trabalhos no estado SUBMITTED	1000000
Número máximo de transações por segundo (TPS) para cada conta para operações SubmitJob	50
Número máximo de recursos consumíveis	50k
Número máximo de ambientes de serviço. Para obter mais informações, consulte Ambientes de serviço .	50
Número máximo de ambientes de serviço para cada fila de trabalho	1
Tamanho máximo da solicitação SubmitServiceJob	30 KiB

Recurso	Quota
Tamanho máximo da carga útil de solicitação de trabalho de serviço (para operações da API SubmitServiceJob)	10 KiB
Número máximo de transações por segundo (TPS) para cada conta para operações SubmitServiceJob	5
Número máximo de tentativas com estratégia de repetição para um trabalho de serviço	10

Dependendo de como você usa o AWS Batch, podem ser aplicadas cotas adicionais. Para saber mais sobre as cotas do Amazon EC2, consulte [Amazon EC2 Service Quotas](#) no Referência geral da AWS. Para obter mais informações sobre as cotas do Amazon ECS, consulte [Amazon ECS Service Quotas](#) no Referência geral da AWS. Para obter mais informações sobre as cotas do Amazon EKS, consulte [Amazon EKS Service Quotas](#) no Referência geral da AWS.

Histórico do documento

A tabela a seguir descreve as importantes alterações na documentação desde a versão inicial do AWS Batch. Também atualizamos a documentação com frequência para abordar os comentários enviados por você.

Alteração	Descrição	Data
Adicionados default_x86_64 e default_arm64	Foram adicionados novos default_x86_64 e default_arm64 para os Tipos de instância permitidos.	15 de agosto de 2025
Adicionados ambientes de serviço e trabalhos de serviço	Ambientes de serviço e trabalhos de serviço adicionados para usar o AWS Batch com o SageMaker AI.	30 de julho de 2025
Adicionado o AWSServiceRoleForAWSBatchWithSageMaker e o AWSBatchServiceRolePolicyForSageMaker	Foi adicionada uma nova função AWS vinculada ao serviço AWSServiceRoleForAWSBatchWithSageMaker e uma política gerenciada AWSBatchServiceRolePolicyForSageMaker que permitem ao AWS Batch gerenciar o SageMaker AI em seu nome.	30 de julho de 2025
Adiciona suporte para EKS AL2023 AMIs	Como atualizar do EKS AL2 para o EKS AL2023	24 de junho de 2025
Adiciona suporte para FireLens e comandos ECS Exec	Adiciona suporte para FireLens e comandos ECS Exec.	15 de abril de 2025

Adiciona suporte para programação com reconhecimento de recursos para o AWS Batch	Adiciona suporte para programação com consciência de recursos para o AWS Batch para o Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e o AWS Fargate.	27 de fevereiro de 2025
Atualização das versões do Amazon EKS compatíveis com o AWS Batch	Atualização das versões do Amazon EKS compatíveis pelo AWS Batch para remover a versão 1.22.	11 de março de 2024
Atualização das versões do Amazon EKS compatíveis com o AWS Batch	Atualização das versões do Amazon EKS compatíveis pelo AWS Batch para incluir a versão 1.29.	29 de fevereiro de 2024
Repetições de trabalho automatizadas	Corrigido o exemplo de código.	29 de fevereiro de 2024
Adiciona suporte a trabalhos com vários contêineres para o AWS Batch	Adiciona suporte a trabalhos com vários contêineres para o AWS Batch para o Amazon Elastic Container Service, o Amazon Elastic Kubernetes Service e o AWS Fargate.	28 de fevereiro de 2024
Atualização das versões do Amazon EKS compatíveis com o AWS Batch	Atualização das versões do Amazon EKS compatíveis pelo AWS Batch para incluir a versão 1.28	27 de janeiro de 2024

[Atualização do BatchServiceRolePolicy e do AWSBatchServiceRole](#)

BatchServiceRolePolicy

5 de dezembro de 2023

Atualização para adicionar suporte à descrição do histórico de solicitações do Spot Fleet e de atividades do Amazon EC2 Auto Scaling.

AWSBatchServiceRole

Atualização para adicionar IDs de extrato, conceder permissões do AWS Batch ao `ec2:DescribeSpotFleetRequestsHistory` e `autoscaling:DescribeScalingActivities`.

[AWS Batch no Amazon EKS](#)

AWS Batch adiciona suporte para execução de trabalhos em clusters do Amazon EKS.

25 de outubro de 2022

[Prevenção do problema do “confused deputy” entre serviços para AWS Batch](#)

AWS Batch agora fornece uma solução alternativa para o problema de segurança “confused deputy”, que surge quando uma entidade (um serviço ou uma conta) é coagida por outra entidade a realizar uma ação.

6 de junho de 2022

<u>VPC endpoint de interface (AWS PrivateLink)</u>	Adicionado suporte para a configuração de endpoints da VPC de interface com tecnologia AWS PrivateLink. Isso permite criar uma conexão privada entre sua VPC e o AWS Batch sem necessidade de acesso pela Internet por meio de uma instância NAT, de uma conexão VPN ou do Direct Connect.	15 de abril de 2022
<u>Atualizações aprimoradas do ambiente de computação</u>	Atualizações aprimoradas de suporte do AWS Batch para ambientes de computação.	14 de abril de 2022
<u>AWS Atualizações de política gerenciada da : atualização de uma política existente</u>	AWS Batch políticas gerenciadas existentes atualizadas.	6 de dezembro de 2021
<u>Programação de compartilhamento justo</u>	AWS Batch adiciona suporte para adicionar políticas de agendamento às filas de trabalhos.	9 de novembro de 2021
<u>Amazon EFS</u>	AWS Batch adiciona suporte para adicionar sistemas de arquivos Amazon EFS às suas definições de trabalho.	1.º de abril de 2021
<u>Perfil adicionado vinculado a serviço</u>	AWS Batch adiciona o perfil vinculado ao serviço AWSServiceRoleForBatch.	10 de março de 2021
<u>Ajuda do AWS Fargate</u>	AWS Batch adiciona suporte para trabalhos executados nos recursos do Fargate.	3 de dezembro de 2020

Marcação de recursos	AWS Batch adiciona suporte para adicionar tags de metadados aos seus ambientes de computação, definições de tarefas, filas de tarefas e tarefas.	7 de outubro de 2020
Segredos	AWS Batch adiciona suporte para transmitir segredos para trabalhos.	1.º de outubro de 2020
Registro em log de	AWS Batch adiciona suporte para especificar drivers de log adicionais para trabalhos.	1.º de outubro de 2020
Estratégias de alocação	AWS Batch adiciona suporte a várias estratégias para escolher os tipos de instância.	16 de outubro de 2019
Suporte do EFA	AWS Batch adiciona suporte para dispositivos Elastic Fabric Adapter (EFA).	2 de agosto de 2019
Programação de GPUs	AWS Batch adiciona agendamento de GPU. Com esse recurso, você pode especificar o número de GPUs que cada trabalho exige e AWS Batch escala as instâncias de acordo.	4 de abril de 2019
Trabalhos em paralelo de vários nós	AWS Batch adiciona suporte para trabalhos paralelos de vários nós. Você pode usar esse recurso para executar trabalhos únicos que abrangem várias instâncias do Amazon EC2.	19 de novembro de 2018

<u>Permissões em nível de recurso</u>	AWS Batch suporta permissões no nível do recurso em várias operações da API.	12 de novembro de 2018
<u>Suporte ao modelo de lançamento do Amazon EC</u>	AWS Batch adiciona suporte para o uso de modelos de execução com ambientes de computação.	12 de novembro de 2018
<u>AWS Batch Tempos limite de trabalhos do</u>	AWS Batch adiciona suporte para o tempo limite do trabalho. Com este suporte, você pode configurar o tempo limite dos seus trabalhos para que o AWS Batch encerre um trabalho caso ele esteja em execução por mais tempo.	5 de abril de 2018
<u>AWS Batch Trabalhos como destinos do EventBridge</u>	Trabalhos AWS Batch são disponibilizados como destinos do EventBridge. Ao criar regras simples, você pode combinar eventos e enviar trabalhos AWS Batch em resposta a eles.	1º de março de 2018
<u>Auditoria do CloudTrail para AWS Batch</u>	O CloudTrail pode auditar chamadas feitas para ações da API do AWS Batch.	10 de janeiro de 2018
<u>Trabalhos de matriz</u>	AWS Batch adiciona suporte para trabalhos de matriz. Você pode usar trabalhos de matriz para varredura de parâmetros e cargas de trabalho de Monte Carlo.	28 de novembro de 2017

<u>Marcação do AWS Batch expandida</u>	AWS Batch expande o suporte para a função de marcação. Você pode usar essa função para especificar tags para instâncias spot do Amazon EC2 lançadas em ambientes de computação gerenciados.	26 de outubro de 2017
<u>AWS Batch stream de eventos para EventBridge</u>	AWS Batch adiciona o stream de eventos para o EventBridge. É possível usar o stream de eventos do AWS Batch para receber notificações quase em tempo real sobre o estado atual de trabalhos que foram enviados a filas de trabalho.	24 de outubro de 2017
<u>Repetições de trabalho automatizadas</u>	AWS Batch adiciona suporte para novas tentativas de emprego. Com esta atualização, você pode aplicar uma estratégia de repetição a trabalhos e definições de tarefa que permite que seus trabalhos sejam automaticamente repetidos em caso de falha.	28 de março de 2017
<u>Disponibilidade geral do AWS Batch</u>	AWS Batch é introduzido, projetado como um meio de executar workloads de computação em lote na Nuvem AWS.	5 de janeiro de 2017

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